



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Evaluación de estabilidad de fallas con control cinemático en taludes rocosos mediante probabilidad conjunta

Jenny Astrid Barón Moreno

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Civil y Agrícola
Bogotá, Colombia
2023

Evaluación de estabilidad de fallas con control cinemático en taludes rocosos mediante probabilidad conjunta

Jenny Astrid Barón Moreno

Tesis de investigación presentada como requisito parcial para optar al título de:
Magister en Ingeniería – Geotecnia

Director:

Ph.D., Rodrigo Hernández Carrillo

Codirectora:

Ph.D., Gloria Inés Beltrán Calvo

Línea de Investigación:

Análisis de confiabilidad y riesgos asociados al entorno geotécnico

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería Civil y Agrícola

Bogotá, Colombia

2023

A Dios, por haberme dado la fortaleza y sabiduría para continuar.

A mis padres Félix e Inés, a mis hermanos, Fabián, Magda y Laura por su apoyo y comprensión incondicional.

Agradecimientos

En primer lugar, agradezco plenamente a mi familia, a mis padres: Félix e Inés y mis hermanos: Fabian, Magda y Laura, por su acompañamiento, animo e incondicionalidad en cada momento de mi vida y en este proceso universitario. Son mi motor para seguir adelante.

Agradezco de sobremanera al Dr. Rodrigo Hernández Carrillo, profesor de la Universidad Pontificia Bolivariana, por el apoyo y confianza que me brindo en la organización y enfoque de la investigación, su conocimiento sobre el tema e interés por la propuesta me permitieron lograr darle mayor valor al trabajo desarrollado.

De igual forma, expreso mi gratitud infinita a la Dra. Gloria Inés Beltrán, profesora de la Universidad Nacional de Colombia, por su confianza, respaldo y apoyo en la universidad, su gestión y entereza brindada fueron claves para continuar con la presentación de la investigación; sus concejos, sugerencias y conocimiento fueron oportunos en la buena orientación del trabajo.

Resumen

Evaluación de estabilidad de fallas con control cinemático en taludes rocosos mediante probabilidad conjunta

En el análisis de estabilidad cinemática en taludes en roca son varias las propuestas que se han perfeccionado, es el caso de los análisis con enfoque estadístico y probabilístico que permiten el estudio de la incertidumbre y variabilidad de los parámetros inherentes a la estabilidad como son la orientación de las discontinuidades y los parámetros de resistencia. La implementación de metodologías que presenten de forma gráfica la probabilidad de falla en taludes en roca no ha sido ampliamente desarrollada. En esta tesis se presenta una metodología para el cálculo de la probabilidad de falla con análisis de la variabilidad de orientación de discontinuidades y del ángulo de fricción de manera probabilística para diferentes mecanismos de falla en taludes en roca; con la aplicación de las funciones de densidad de probabilidad de variables se proyecta el cálculo de la probabilidad de falla mediante la probabilidad conjunta y la representación gráfica en 2D en herramientas normalmente usadas en el estudio de estabilidad cinemática como es la red estereográfica para los diferentes mecanismos de falla. Luego el análisis cinético de confiabilidad es presentado como complemento a la metodología con el cual se sugiere un tratamiento a través de probabilidad conjunta de variables específicas y la construcción de graficas 3D o volúmenes de probabilidad para cada mecanismo de falla a estudiar. Se desarrolla una herramienta de cálculo para este trabajo en lenguaje de programación Python en su aplicación PyCharm Community Edition y se implementa con datos de una mina a cielo abierto.

Palabras clave: Rocas, discontinuidades, probabilidad conjunta, análisis cinemático y cinético, red estereográfica.

Abstract

Evaluation of stability of faults with kinematic control in rock slopes by means of joint probability.

In the analysis of kinematic stability in rock slopes, there are several proposals that have been perfected, this is the case of analyzes with a statistical and probabilistic approach that allow the study of uncertainty and scarcity of parameters inherent to stability such as orientation of the discontinuities and resistance parameters. The implementation of methodologies that graphically present the probability of failure in rock slopes has not been widely developed. This thesis presents a methodology for calculating the probability of failure with analysis of the variability of orientation of discontinuities and of the friction angle in a probabilistic way for different failure mechanisms in rock slopes, which with the application of the functions of probability density of variables, the calculation of the probability of failure is projected through the joint probability and the 2D graphic representation in tools normally used in the study of kinematic stability, such as the stereographic network for the different failure mechanisms. Then the kinetic reliability analysis is presented as a complement to the methodology with which treatment is suggested through the joint probability of specific variables and the construction of 3D graphs or probability volumes for each failure mechanism to be studied. A calculation tool for this work is developed in the Python programming language in its PyCharm Community Edition application and implemented with data from an open pit mine.

Keywords: Rocks, discontinuities, joint probability, kinematic and kinetic analysis, stereographic network.

Contenido

	Pág.
Resumen	VII
Lista de figuras	XI
Lista de tablas	XV
Introducción	1
1. Antecedentes	5
2. Base conceptual	15
2.1 Tratamiento probabilístico de las variables.....	15
2.2 Análisis estadístico de las discontinuidades	18
▪ Cálculo de la orientación media	19
▪ Cálculo de la dispersión (k).....	20
▪ Cálculo del límite de confianza	20
▪ Cálculo de la desviación angular θ	21
2.3 Cálculo de la probabilidad de Fisher	21
2.4 Análisis estadístico del ángulo de fricción de los planos.....	24
2.5 Construcción de envolvente de afloramiento para taludes por falla planar	25
2.6 Análisis cinético de confiabilidad	26
2.6.1 Bloque deslizando sobre un plano inclinado	27
2.6.2 Volteo de bloques (método de Hoek y Bray).....	31
2.6.3 Cuña deslizando por dos planos que se intersecan	34
3. Metodología propuesta	43
3.1 Metodología para la evaluación probabilística	43
3.1.1 Probabilidad conjunta en red polar.....	45
3.2 Análisis cinético de confiabilidad	59
4. Caso de estudio y aplicación de la metodología	75
4.1 Área de estudio	75
4.2 Análisis de datos	76
4.2.1 Análisis estadístico de las discontinuidades.....	77
4.2.2 Análisis estadístico del ángulo de fricción	81
4.2.3 Probabilidad conjunta en red polar - Bloque Norte.....	83
4.2.4 Análisis por confiabilidad Bloque Norte.....	91
4.2.5 Probabilidad conjunta en red polar - Bloque sur.....	95
4.2.6 Análisis por confiabilidad - Bloque Sur.....	103
4.3 Análisis de resultados	107

5. Conclusiones y recomendaciones	113
5.1 Conclusiones	113
5.2 Recomendaciones	118
5.3 Limitaciones.....	118
A. Anexo: Trabajos desarrollados en análisis probabilístico en taludes en roca	121
B. Anexo: Funciones de densidad y distribución de Fisher.....	123
C. Anexo: Datos estructurales de caso de estudio.....	125
D. Anexo: Interfaz del programa.....	127
E. Anexo: Código fuente del programa	129
Bibliografía	130

Lista de figuras

	Pág.
Figura 1-1. $f(n)$ vs n para Distribución Fisher. Distribución Fisher. Fuente: Glynn, (1979).	6
Figura 1-2. Mapa de probabilidad de riesgo falla en cuña. Fuente: Gokceoglu et al., (2000).....	8
Figura 1-3. Captura de pantalla de la pestaña "Resultados" de ROCKTOPPLE que ilustra la salida típica del programa. Fuente: Tatone & Grasselli, (2010).....	9
Figura 1-4. Histogramas experimentales para los ángulos de orientación de un conjunto de discontinuidades dado. Fuente: Obregon & Mitri, (2019)	11
Figura 2-1. Cálculo del factor dividiendo ψ de la ecuación (2-18) .Fuente: propia	23
Figura 2-2. A). Definición de ángulos utilizados en la derivación de la fórmula para las coordenadas de puntos en el sobre de luz diurna. B) Ejemplos de zona critica natural para una serie de taludes de rocas con ángulos de inclinación que van desde 5° a 85° . Fuente: Lisle, (2004).....	25
Figura 2-3. Esquema de fuerzas actuantes para bloque deslizante.....	27
Figura 2-4. Función de densidad de probabilidad y función acumulada de ϕ°	28
Figura 2-5. Envolvente de factor de seguridad para C_r	29
Figura 2-6. Diagrama de funciones de probabilidad conjunta	30
Figura 2-7. Volumen resultante de probabilidad conjunta	30
Figura 2-8. Análisis de fuerzas para volteo de bloques.....	31
Figura 2-9. Esquema de variables probabilísticas para análisis de estabilidad por volteo.....	32
Figura 2-10. Probabilidad conjunta para análisis de estabilidad por volteo	33
Figura 2-11. Esquema de volumen de probabilidad.....	33
Figura 2-12. Acciones y reacciones sobre la cuña. Fuente: Granados, (2009).	36
Figura 2-13. Prisma de Presiones sobre la cuña. Fuente: Granados, (2009).....	36
Figura 2-14. Zona de falla para cohesión Fuente: Granados, (2009).....	39
Figura 2-15. Zona de falla para ángulo de fricción Fuente: Granados, (2009)	39
Figura 2-16. Volumen de Probabilidades sobre el área de falla Fuente: Granados, (2009)	40
Figura 3-1. Metodología propuesta para análisis probabilístico cinemático de falla. Fuente: propia	44
Figura 3-2. Red polar, localización planos de familias de discontinuidades Fuente: Resultados familias de discontinuidades ejercicio Goodman, (1989).....	48
Figura 3-3. Contornos de probabilidades con respecto a familia F1 Fuente: propia.....	49
Figura 3-4. Función de densidad de probabilidad para el ángulo de fricción Fuente: propia	49

Figura 3-5: Probabilidad de ángulo de fricción. Fuente: propia.....	50
Figura 3-6. Envoltentes y cono de probabilidad falla planar. Fuente: propia.....	52
Figura 3-7. Zona de falla critica falla planar. Fuente: propia.....	52
Figura 3-8. Probabilidad conjunta de falla planar Fuente: propia.....	54
Figura 3-9. Contornos de probabilidades con respecto a intersección F1-F2. Fuente: propia.....	57
Figura 3-10. Envoltentes y cono de probabilidad falla en cuña por F1-F2. Fuente: propia.....	58
Figura 3-11. Zona de falla critica falla en cuña por F1-F2. Fuente: propia.....	58
. Figura 3-12. Probabilidad conjunta falla en cuña por F1-F2. Fuente: propia.....	59
Figura 3-13. Histogramas y funciones de probabilidad Normal para ángulo de fricción y cohesión. Fuente: propia.....	60
Figura 3-14. Funciones de densidad y FS= 1. Fuente: propia.....	61
Figura 3-15. Volumen de probabilidad conjunta con funciones de densidad Beta. Fuente: propia.....	62
Figura 3-16. Probabilidad conjunta y límite de FS= 1, zona de falla. Fuente: propia.....	62
Figura 3-17. Función de densidad de probabilidad y función de distribución de probabilidad Fuente: propia.....	63
Figura 3-18. Función de densidad y distribución para superficie de deslizamiento o volteo. Fuente: propia.....	63
Figura 3-19. Funciones de densidad ángulo de fricción y de la inclinación de superficie de falla. Fuente: propia.....	65
Figura 3-20. Volumen probabilidad conjunta. Fuente: propia.....	66
Figura 3-21. Probabilidad conjunta, inclinación de superficie de falla 60°. Fuente: propia.....	67
Figura 3-22. Zonas de falla para cohesión y ángulo de fricción de los dos planos. Fuente: propia.....	68
Figura 3-23. Funciones de densidad sobre zona de falla para cálculo de volumen de probabilidad para cohesión Fuente: propia.....	68
Figura 3-24. Volumen de probabilidad final para cohesión. Fuente: propia.....	70
Figura 3-25. Visa en planta zona de falla sobre volumen de probabilidad final para cohesión. Fuente: propia.....	70
Figura 3-26. Funciones de densidad sobre zona de falla para cálculo de volumen de probabilidad para ángulo de fricción. Fuente: propia.....	71
Figura 3-27. Volumen de probabilidad final para ángulo de fricción. Fuente: propia.....	73
Figura 3-28. Visa en planta zona de falla sobre volumen de probabilidad final para ángulo de fricción. Fuente: propia.....	73
Figura 4-1. Esquema localización Mina el Pedregal. Fuente: Google Earth.....	76
Figura 4-2. Datos estructurales Bloque Norte. Fuente: propia+ Dips.....	77
Figura 4-3 Familias de discontinuidades Bloque Norte. Fuente: propia.....	78
Figura 4-4. Zonificación falla planar Bloque Norte. Fuente: propia.....	78
Figura 4-5. Datos estructurales Bloque Sur. Fuente: propia+ Dips.....	79
Figura 4-6 Familias de discontinuidades Bloque Sur. Fuente: propia.....	80
Figura 4-7. Zonificación falla planar Bloque Sur. Fuente: propia.....	80

Figura 4-8. Distribución normal truncada para el ángulo de fricción Fuente: propia 82

Figura 4-9. Probabilidad de ángulo de fricción. Fuente: propia 83

Figura 4-10. Contornos de probabilidades falla planar para F1 84

Figura 4-11. Zona de falla critica por falla planar para F1 84

Figura 4-12. Contornos de probabilidades falla planar para F2 85

Figura 4-13. Zona de falla critica por falla planar para F2..... 85

Figura 4-14. Contornos de probabilidades falla planar para F4 86

Figura 4-15. Zona de falla critica por falla planar para F4..... 86

Figura 4-16. Contornos de probabilidades, falla por volteo para F2..... 87

Figura 4-17. Zona de falla critica por falla por volteo para F2 88

Figura 4-18. Contornos de probabilidades, falla por volteo para F3..... 88

Figura 4-19. Zona de falla critica por falla por volteo para F3 89

Figura 4-20. Zona de falla critica, falla en cuña por F2-F3..... 90

Figura 4-21. Probabilidad conjunta, falla en cuña por F2-F3..... 90

Figura 4-22. Volumen de probabilidad conjunta con funciones, falla planar..... 91

Figura 4-23. Probabilidad conjunta y FS= 1, zona de falla..... 92

Figura 4-24. Volumen probabilidad conjunta, falla por volteo..... 92

Figura 4-25. Probabilidad conjunta falla por volteo 93

Figura 4-26. Volumen de probabilidad final para ángulo de fricción..... 93

Figura 4-27. Zona de falla sobre volumen de probabilidad final para ángulo de fricción. 94

Figura 4-28. Volumen de probabilidad final para cohesión 94

Figura 4-29. Zona de falla sobre volumen de probabilidad final para cohesión..... 95

Figura 4-30. Contornos de probabilidades, falla planar para F1 96

Figura 4-31. Zona de falla critica por falla planar para F1 96

Figura 4-32. Contornos de probabilidades, falla planar para F2 97

Figura 4-33. Zona de falla critica por falla planar para F2..... 97

Figura 4-34. Contornos de probabilidades falla planar para F3 98

Figura 4-35. Zona de falla critica por falla planar para F3..... 98

Figura 4-36. Contornos de probabilidades falla por volteo para F3..... 99

Figura 4-37. Zona de falla critica por falla por volteo para F3..... 100

Figura 4-38. Zona de falla critica falla en cuña por F1-F2..... 100

Figura 4-39. Probabilidad conjunta falla en cuña por F1-F2..... 101

Figura 4-40. Zona de falla critica falla en cuña por F1- F3..... 101

Figura 4-41. Probabilidad conjunta falla en cuña por F1-F3..... 102

Figura 4-42. Zona de falla critica falla en cuña por F2- F3..... 102

Figura 4-43. Probabilidad conjunta falla en cuña por F1-F3..... 103

Figura 4-44. Volumen de probabilidad conjunta con funciones falla planar..... 103

Figura 4-45. Probabilidad conjunta y FS= 1, zona de falla 104

Figura 4-46. Volumen probabilidad conjunta falla por volteo 104

Figura 4-47. Probabilidad conjunta falla por volteo 105

Figura 4-48. Volumen de probabilidad final para ángulo de fricción..... 105

Figura 4-49. Zona de falla sobre volumen de probabilidad final para ángulo de fricción 106

Figura 4-50. Volumen de probabilidad final para cohesión 106

Figura 4-51. Zona de falla sobre volumen de probabilidad final para cohesión..... 107

Figura 4-52. Probabilidad de falla mecanismo de falla planar Bloque Norte	108
Figura 4-53. Probabilidad de falla mecanismo de falla por volteo Bloque Norte	109
Figura 4-56. Probabilidad de falla mecanismo de falla por cuña Bloque Norte	109
Figura 4-54. Probabilidad de falla mecanismo de falla planar Bloque Sur	110
Figura 4-55. Probabilidad de falla mecanismo de falla por volteo Bloque Sur	110
Figura 4-55. Probabilidad de falla mecanismo de falla por Cuña Bloque Sur	111

Lista de tablas

	Pág.
Tabla 2-1. Funciones de distribución de probabilidad para los distintos parámetros.....	16
Tabla 4-1. Resumen análisis estadístico Bloque Norte.....	77
Tabla 4-2. Resumen análisis estadístico Bloque Sur	79
Tabla 4-3. Resumen de resultados de ensayos de corte por discontinuidades.....	81
Tabla 4-4. Datos y cálculo de función de densidad y distribución del ángulo de fricción.	82

Introducción

Los análisis geotécnicos de estabilidad van dirigidos al favorecimiento de un bienestar social, económico o disminución del impacto ambiental. La variabilidad en el comportamiento y propiedades de los materiales con los que se debe tratar hacen necesario el uso de métodos probabilísticos para considerar de forma explícita la incertidumbre a las sollicitaciones, geometría y parámetros geomecánicos de los modelos de estabilidad.

En el análisis de estabilidad de taludes en roca son numerosas las variables para tener en cuenta. Por una parte, están las fuerzas actuantes como, por ejemplo, el peso propio del material, las fuerzas hidrostáticas y de flujo, y las fuerzas sísmicas (Bolle, 1987). Por otra parte, están las propiedades de resistencia del material rocoso y la geometría de las discontinuidades del macizo, entre otras. La incertidumbre de estos parámetros se asocia a su variabilidad inherente por tratarse de materiales de origen geológico, pero también está ligada a la metodología dispersa, sistemática y quizás subjetiva de medición de datos (Ya-Fen Lee, 2012).

Por lo anterior, no resulta apropiado tratar los parámetros de un modelo de estabilidad, como un único valor determinístico. En su lugar, se recomienda darles un tratamiento probabilístico a las variables, con el fin de poder propagar esta incertidumbre a través del modelo estabilidad de taludes. En este sentido, existen diferentes niveles para el tratamiento de las variables, desde expresarlas con valores determinísticos que implícitamente consideren la variabilidad de las variables, hasta un enfoque completamente probabilístico, donde las variables de entrada se expresan como variables aleatorias.

El conocimiento de la distribución de probabilidad de cada uno de los parámetros involucrados en la estabilidad de un talud en roca permite evaluar la probabilidad de falla mediante la interacción de un conjunto de variables aleatorias, determinar un criterio de

falla permisible y emplearlo en la valoración y diseño de la estabilidad de un talud en particular.

Con este marco de referencia, en este trabajo se presentan diferentes alternativas para el análisis de las variables, que incluyen el tratamiento estadístico de los datos y metodologías con enfoque probabilístico. Posteriormente, se plantea una metodología para la evaluación de la probabilidad de fallas con control cinemático y el uso de probabilidad conjunta de variables aleatorias para el desarrollo dentro de una representación gráfica en red polar. Consecuentemente, la principal contribución de este trabajo se centra en involucrar una herramienta de uso cotidiano en mecánica de rocas, como lo es la red estereográfica, en la evaluación de los análisis probabilísticos.

El documento se desarrolla en cuatro capítulos: el primero muestra los antecedentes y los diferentes estudios e investigaciones desarrolladas a través de los años y que tratan la probabilidad de falla en taludes mediante alguna metodología gráfica; el segundo capítulo indica la base conceptual con la que se desarrollara parte de la metodología; para el capítulo 3 se indica paso a paso la metodología propuesta para la evaluación probabilística cinemática para cada uno de los mecanismos de falla representada como la probabilidad conjunta en red polar, luego se complementa la propuesta con la evaluación de la probabilidad conjunta en análisis cinético; finalmente en el capítulo 4 se presenta la aplicación del código desarrollado, mostrando los resultados del análisis probabilístico para datos estructurales y de resistencia tomados de una mina a cielo abierto del municipio de Uña Cundinamarca.

Objetivo general

Desarrollar una metodología para la evaluación probabilística de la estabilidad con control cinemático de taludes rocosos utilizando probabilidad conjunta de variables aleatorias.

Objetivos específicos

- Modelar la incertidumbre de los parámetros geométricos y geomecánicos como variables aleatorias.
- Articular la variabilidad de los parámetros geomecánicos y geométricos del problema de estabilidad en un modelo probabilístico mediante la red estereográfica.

- Implementar los algoritmos planteados para efectuar los análisis probabilísticos de estabilidad de taludes rocosos en un código de computador.

1. Antecedentes

La metodología convencional empleada para el análisis de estabilidad de taludes en roca es realizada mediante determinaciones estadísticas; sin embargo, la experiencia que ha dejado este tipo de análisis indica que hay incertidumbres no cuantificadas, por lo que diversos investigadores han propuesto un tratamiento diferente a dichos análisis, con los que se busca identificar y cuantificar variables influyentes, que de otra manera no podrían ser incorporadas en los análisis de estabilidad convencional.

En consecuencia, se hace necesario el empleo de metodologías para el manejo adecuado de la incertidumbre y poder evaluar la probabilidad de que se presenten diferentes escenarios de falla, reconociendo que el proceso de diseño debe considerar la interacción entre varios factores y su contribución a todo el sistema, señala Tang et al., (1976).

En el campo de la geotecnia, fueron pioneros los trabajos de Terzaghi, Casagrande, Skempton y Peck en los años de 1930 a 1940 quienes reconocieron la incertidumbre de los sistemas geológicos y la asociada a las características propias de los suelos. Para manejar el desconocimiento de las propiedades de los suelos, se implementaron técnicas que incluían el riesgo y la confiabilidad. Una de estas técnicas fue el factor de seguridad, pero su aplicación obedece más a conocimientos heurísticos y empíricos que a tratamientos estadísticos de la información.

En la década de 1970 se reporta la primera publicación de la teoría de la probabilidad en la geotecnia junto con la creación del Comité sobre Conceptos de Probabilidad y Confiabilidad en el Diseño de la Ingeniería Geotécnica de la ASCE, surgiendo entonces varias publicaciones por parte de los investigadores en geotecnia. Esto generó un nuevo enfoque para la valoración del riesgo de amenazas naturales en la ingeniería de rocas, en minería e ingeniería geotécnica que fue impulsado por el avance en los programas de

computador que facilitan la interpretación y generación resultados de los análisis probabilísticos.

Con respecto a los análisis probabilísticos con enfoque hacia el análisis cinemático, varios son los autores que han propuesto metodologías que analizan la interacción de los diferentes parámetros geométricos, de resistencia, la representación de los datos de orientación de discontinuidades y modelos probabilísticos en el cálculo de probabilidad de falla.

En 1971, McMahon introdujo la noción de usar la probabilidad de falla en lugar del factor de seguridad en los análisis de estabilidad de taludes rocosos. Propuso los procedimientos requeridos para la estabilidad cinemática y la estabilidad cinética para los métodos de análisis probabilísticos. Glynn, (1979) da a conocer algunas herramientas para usar en el análisis de la estabilidad de taludes en roca desde un punto de vista probabilístico. Presenta un procedimiento que permite utilizar las funciones de densidad de probabilidad (PDF, por sus siglas en inglés) de la orientación de las discontinuidades para derivar a las PDF de parámetros que caracterizan la forma y la orientación de las posibles cuñas de roca delimitadas por las discontinuidades. Igualmente, combina los modelos determinísticos con la simulación de Monte Carlo para que algunos de los parámetros de entrada se puedan tratar como variables aleatorias.

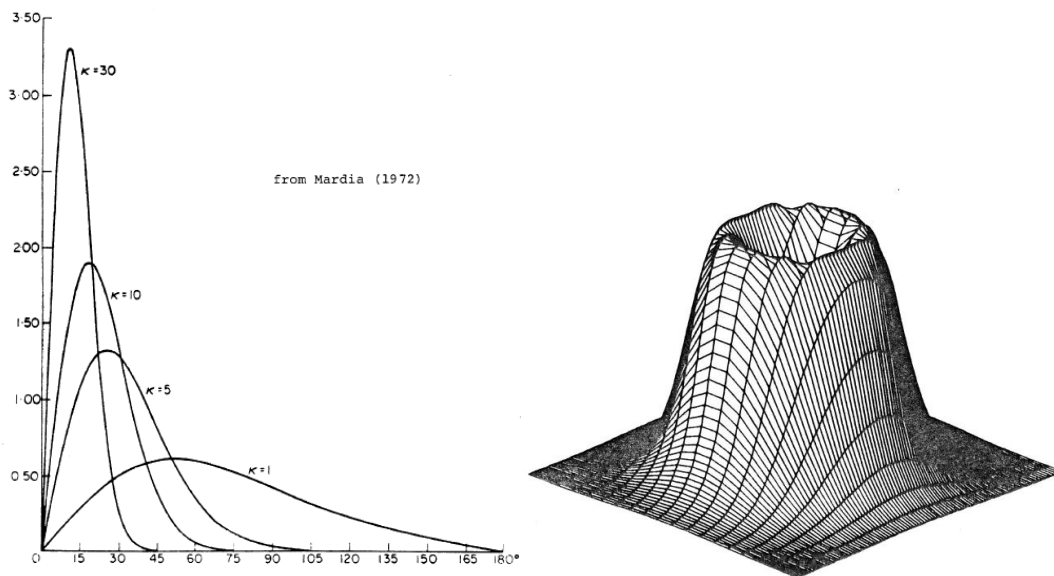


Figura 1-1. $f(n)$ vs n para Distribución Fisher. Distribución Fisher. Fuente: Glynn, (1979).

Por su parte, Miller (1983) recomienda un análisis probabilístico de la estabilidad de taludes para el uso en el diseño de minas a cielo abierto, en donde las inestabilidades del banco suelen influir en las geometrías generales de los taludes de la mina. Identifica los posibles modos de falla evaluando cómo interactúan las orientaciones estructurales mapeadas en un sector dado. Los datos cartografiados también se utilizan para estimar distribuciones estadísticas y correlaciones espaciales de los conjuntos de discontinuidades que comprometen los posibles modos de falla. La probabilidad de falla para varios mecanismos de falla de taludes se suele estimar mediante técnicas de simulación de Monte Carlo, el cual se basan en el muestreo repetido de las distribuciones de probabilidad de los parámetros de entrada. Carter & Lajtai, (1991) señalan que la probabilidad de falla se expresa como la proporción de cuñas cinemáticamente libres que tienen un factor de seguridad menor que la unidad con respecto al número total de cuñas. Introducen un código determinista (GEOSLIDE) y un código de microordenador probabilístico (PROSLIDE) para ayudar a realizar análisis de cuña de roca basados en el método de equilibrio límite. Posteriormente, Leung & Quek, (1995) señalan que la orientación de las discontinuidades es uno de los principales parámetros de entrada en el análisis de estabilidad determinista convencional de los bloques de roca, proponiendo un enfoque probabilístico para incorporar la constante de Fisher en el análisis de estabilidad de bloques de roca.

Park & West, (2000) desarrollan un enfoque probabilístico para la falla en cuña en roca, señalando que, para la ingeniería de taludes en roca, la incertidumbre y la variabilidad son inherentes a los datos recopilados sobre la orientación y la intensidad de las discontinuidades obteniendo variedad de resultados. Como parte de la solución, la probabilidad de inestabilidad cinemática y la probabilidad de inestabilidad cinética se evalúan por separado para proporcionar una evaluación adecuada y combinada para la probabilidad de falla. La probabilidad de falla se define como la probabilidad de tener $FS \leq 1$ dado como un porcentaje igual a la zona bajo la función de densidad de probabilidad (PDF) para $FS \leq 1$. Adicionalmente, usa el método de primer orden segundo momento (FOSM) para calcular la probabilidad de falla de manera aproximada.

Gokceoglu et al., (2000) examinaron la posibilidad de fallas de taludes controlados por las discontinuidades de andesitas fracturadas en la región de asentamientos de Altindag (Ankara), y como resultado, produjeron mapas de riesgo probabilístico para falla planar,

volteo y de cuña utilizando las reglas cinemáticas y el modelo de elevación digital del área de estudio. La comparación de la distribución de las fallas reales en el área y los mapas de riesgo probabilístico preparados para el área de estudio reveló que todas las fallas identificadas se encuentran ubicadas en las zonas de mayor riesgo en los mapas de riesgo probabilístico.

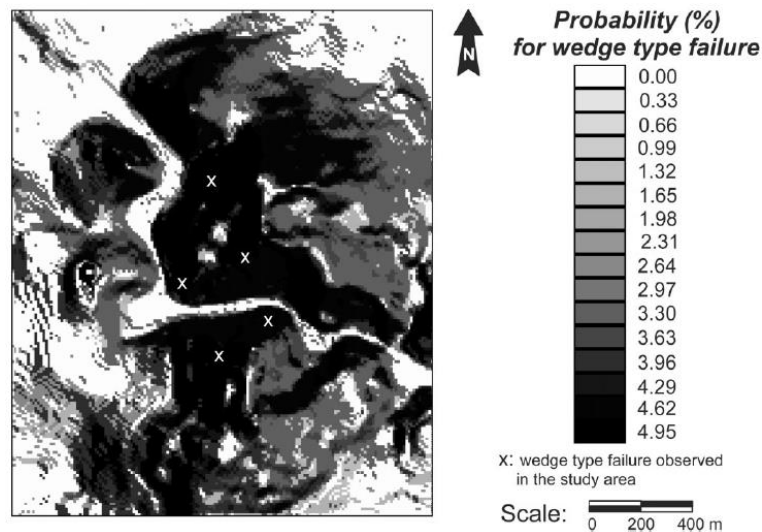


Figura 1-2. Mapa de probabilidad de riesgo falla en cuña. Fuente: Gokceoglu et al., (2000).

Mat Radhi et al., (2008) presentan un enfoque probabilístico de análisis de estabilidad en taludes en roca usando la simulación de Monte Carlo basado en los parámetros de las discontinuidades. El análisis probabilístico se realizó mediante el análisis cinemático y cinético. El análisis cinemático se estableció en el análisis de la proyección estereográfica y el análisis cinético se basó en el análisis determinista. Los datos de entrada para el análisis cinemático son buzamiento y orientación, ángulo de fricción, cohesión, y ángulo de inclinación del talud. Los datos de salida son el tipo de falla del talud: planar, cuña, volteo o combinación de dos tipos de falla. Los datos de entrada para el análisis cinético son propiedades del talud, la cohesión, ángulo de fricción y nivel freático.

Tatone & Grasselli, (2010) proponen el programa ROCKTOPPLE, consistente en hojas de cálculo para análisis probabilístico de volteo de bloques, empleando simulación de Monte Carlo para el análisis. El procedimiento de análisis considera ambas probabilidades de falla, cinemática y cinética. Estas probabilidades se evalúan por separado y se multiplican para dar la probabilidad total del volteo de bloques.

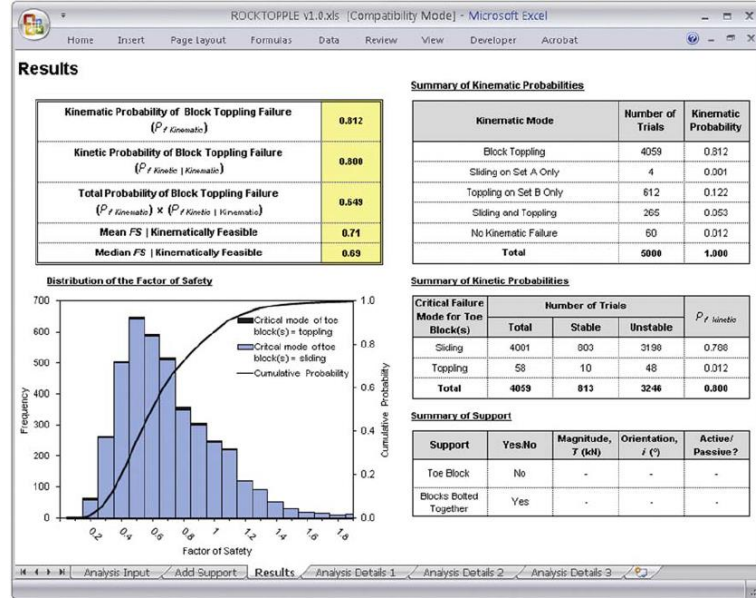


Figura 1-3. Captura de pantalla de la pestaña "Resultados" de ROCKTOPPLE que ilustra la salida típica del programa. Fuente: Tatone & Grasselli, (2010).

Admassu & Shakoor, (2013) plantean la herramienta DIPANALYST: Un programa informático para el análisis cinemático cuantitativo de fallas de taludes en roca, mediante el cual se analiza el potencial de los diversos modos de falla estructuralmente controlados (planar, cuña, y volteo) debido a la presencia de discontinuidades orientadas desfavorablemente. Usan la distribución de la orientación más común, la distribución circular de Fisher, que es la más cercana a la distribución normal (Borradaile, 2003). La probabilidad de inestabilidad cinemática es calculada como la relación entre el número de iteraciones que son cinemáticamente inestables con respecto al número total de iteraciones.

Zheng et al., (2014) presentan un análisis probabilístico de la teoría de bloques para un talud de roca en una mina a cielo abierto en EE.UU, desarrollando un código de computador (PBTA) Probabilistic Block Theory Analysis Code para realizar análisis de teoría de bloques tanto determinísticos como probabilísticos. La variabilidad de la orientación de las discontinuidades y la resistencia al corte se incorpora en el análisis probabilístico de la teoría de bloques. La orientación de discontinuidades se trata como una variable aleatoria bivariada que incluye la correlación que existe entre el ángulo de inclinación y su orientación. Las distribuciones Beta y Fisher se usaron para representar la

variabilidad del ángulo de inclinación y la orientación de las discontinuidades. El ángulo de inclinación y la orientación se tomaron como parámetros no correlacionados.

Jordá-Bordehore et al., (2016) presentan el análisis de estabilidad de taludes de múltiples caras, empleando el concepto de restricción de la envolvente “daylight envelope”, aplicando la metodología al análisis de un caso real de talud en curva convexa en pizarras metamórficas en Madrid, España. Emplearon el criterio de Barton & Bandis (1990) para la estimación de la resistencia al corte en discontinuidades rugosas y determinar el cono de fricción requerido para el análisis cinemático.

Hernandez- Carrillo (2017) presenta una Evaluación de la confiabilidad de la estabilidad de taludes en roca mediante la teoría de Dampster-Shafer, tiene en cuenta la limitada información, la incertidumbre de la veracidad de los datos que se recolectan a través del tiempo y que van alimentando la base de datos para el análisis de estabilidad de los taludes y representan cambios en los resultados con la nueva información que se va obteniendo. Con el uso de la teoría de Dampster y la combina con las propuestas de Yanger para no eliminar los datos contradictorios, sino que los asigna a un conjunto universal. Representa gráficamente cada una de las funciones de probabilidad acumulada de la orientación y ángulo de fricción de los diferentes conjuntos de datos recopilados en diferentes años, en dichas graficas se identifica los intervalos en donde la información presenta conflicto y la que es consistente usando el concepto DS y RST o teoría de conjuntos aleatorios.

Obregon & Mitri, (2019) muestran una evaluación de estabilidad de taludes para una mina a cielo abierto en Perú usando enfoque probabilístico, donde combinan un análisis cinemático basado en técnicas de proyección estereográfica y luego un análisis cinético mediante el método del equilibrio límite. Finalmente, estas dos probabilidades se combinan para proporcionar una medida general de la probabilidad de falla (PoF) del sistema de talud. Realizan un mapeo extenso de la superficie en 36 sitios siguiendo los procedimientos sugeridos por ISRM. Además, llevaron a cabo varios ensayos de corte directo y demostraron que, al combinar mediciones de campo, laboratorio y juicio de ingeniería, las funciones de densidad de probabilidad (PDF) de los parámetros de las discontinuidades pueden ser obtenidas. Luego utilizan simulaciones de Monte Carlo para calcular tanto la probabilidad cinemática como la probabilidad cinética de falla.

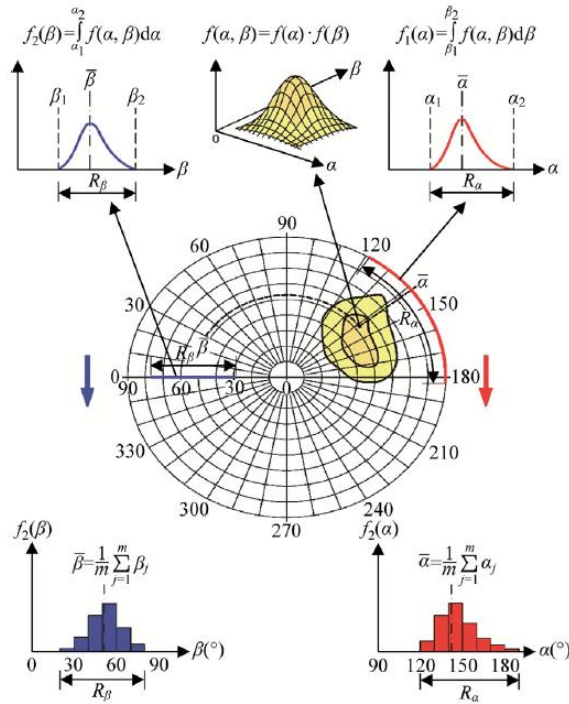


Figura 1-4. Histogramas experimentales para los ángulos de orientación de un conjunto de discontinuidades dado. Fuente: Obregon & Mitri, (2019)

Hernández- Carrillo y Beltrán, (2019) presentan una propuesta para el uso de la teoría de conjuntos aleatorios para combinar diferentes fuentes de información y efectuar análisis de confiabilidad cuando la información es escasa, y a medida que se obtenga nueva información poder actualizar la evaluación de probabilidad de falla para el análisis de confiabilidad de la estabilidad de cuñas de una mina en Une, Cundinamarca.

Hernández – Carillo, Peñuela y Beltrán, (2019) presentan una alternativa a la medición de características estructurales en macizos rocosos usando imágenes estereoscópicas de la cara del talud, usando programas computaciones de referenciación con el objetivo de generar imágenes 3D con la cual se pueda medir la orientación y geometría de las discontinuidades, procesar estos datos y representar los polos en la red estereográfica, permitiendo la agrupación y definición de familias de discontinuidades. Esta propuesta representa grandes aportes a la geotecnia de rocas en la recolección masiva de datos estructurales y aplicación de metodologías probabilísticas para la definición de las familias de discontinuidades representativas del macizo rocoso y demás parámetros geométricos.

Rosada (2020) presenta en su tesis de grado una Propuesta metodológica para el ajuste de una red de fracturas discretas (DFN) a partir de fotogrametría de corto alcance, en donde analiza la susceptibilidad de un modelo DFN ante la presencia de datos que direccionalmente se clasifican como discordantes, y mediante un código de programación estudia los datos mediante criterios estadísticos.

Los modelos empleados en geotecnia parten de la interacción de varios parámetros geométricos, resistencia y de orientación, los cuales, por la incertidumbre inherente a su naturaleza, recopilación y calculo hacen que sea necesaria la implementación de herramientas que faciliten la interpretación y evaluación de datos y resultados. El uso del análisis probabilístico en geotecnia en los últimos años ha posibilitado el empleo de diferentes funciones de densidad para la caracterización de la tendencia de cada una de las variables empleadas en los estudios geotécnicos y el estudio de la estabilidad cinemática y estabilidad cinética evaluadas por separado o combinada para la obtención de una probabilidad de falla.

La implementación de códigos computacionales, de modelos probabilísticos y simulaciones planteados por diferentes autores, han permitido el manejo y procesamiento de datos e interpretación de estos de forma práctica, lo cual posibilita la aplicación de modelos geotécnicos a los diversos mecanismos de falla con la interacción de diferentes variables de entrada.

Varias de las propuestas señaladas presentan análisis probabilísticos cuantitativos de fallas en taludes en roca, códigos deterministas y probabilísticos en análisis de teoría de bloques y diversos modos de falla estructuralmente controlados. Pocas de las propuestas representan los análisis probabilísticos y procesamiento de datos de forma gráfica, que es también una herramienta válida para la interpretación de datos y tendencias que pueden ser estudiados mediante estadística y funciones de probabilidad de datos.

Se hace necesaria la aplicación de probabilidad conjunta de variables en el análisis probabilístico de estabilidad de taludes en roca y la posterior representación gráfica de éstas en herramientas o aplicaciones habitualmente implementadas en ingeniería como es la red estereográfica con la cual se puede representar en dos y tres dimensiones diferentes variables, analizar los datos estructurales y de resistencia como es el ángulo de fricción,

como contornos o zonas con valor probabilístico en la solución de problemas de estabilidad; con la metodología aquí planteada se quiere dar la pauta para el uso de técnicas gráficas y de probabilidad en el desarrollo de nuevos procedimientos en el estudio de la estabilidad de taludes en geotecnia.

2. Base conceptual

En los estudios de la estabilidad de taludes en roca, varias son las herramientas, conceptos y premisas que se han desarrollado y que se han venido implementando dentro de las investigaciones en el área. Como parte fundamental en el desarrollo de la propuesta de investigación, a continuación, se presentan las bases teóricas que fueron estudiadas e implementadas dentro de la metodología propuesta en la investigación.

2.1 Tratamiento probabilístico de las variables

En el caso de los análisis probabilísticos, estos son considerados como una expansión de los análisis deterministas, pues en lugar de usar un único valor para representar un parámetro, en este caso se permite incluir las distribuciones de los parámetros, señala Nadim et al., (2005). La mayoría de estos parámetros son considerados como variables aleatorias y los tipos de funciones de distribución para cada variable aleatoria pueden ser seleccionadas cuidadosamente en un análisis probabilístico (H.-J. Park, 2000).

Un enfoque probabilístico ofrece una manera de incorporar todos los factores que están asociados con la falla de un talud. El método de abordar las incertidumbres en la estabilidad de taludes reconoce que el proceso de diseño no depende exclusivamente de un parámetro, sino que se basa en la interacción entre varios parámetros y sus contribuciones a todo el sistema (Tang et al., 1976).

Debido a la variabilidad de muchos de los parámetros y la dificultad para predecir o identificar un solo valor en los análisis de cada uno de los parámetros con carácter probabilístico es necesario describirlos como variables aleatorias y modelar la probabilidad relativa con el uso de funciones de densidad de probabilidad (PDF por su sigla en inglés).

En los casos en que se cree que un conjunto dado de datos medidos representa un conjunto de valores de muestra representativos de la variable y no hay otra información disponible, una PDF es representativa de la variable aleatoria. El valor medio de las PDF representa la mejor estimación de la variable aleatoria, y la desviación estándar o coeficiente de varianza (Cov) de la PDF que representa una evaluación de la incertidumbre como afirma Pathak et al., (2006).

Se debe tener en cuenta que la distribución más adecuada y que mejor representa las variables, por ejemplo, distribución normal podría producir números negativos que no tienen significado físico, como en la fricción o la cohesión. En estos casos es mejor usar una distribución lognormal indica Gibson, (2011) o una distribución truncada, como la normal truncada que permite analizar los datos limitando los extremos de la distribución.

En la Tabla 2-1 se relacionan las funciones de distribución de probabilidad que varios autores han propuesto para el tratamiento de los datos de cada uno de los parámetros.

Tabla 2-1. Funciones de distribución de probabilidad para los distintos parámetros

Descripción	Parámetro	Función de densidad de probabilidad (PDF)	Proponente
Geometría del talud	Altura	Normal truncada	(Johari & Mehrabani Iari, 2016)
		Uniforme	(Glynn, 1979)
	Inclinación	Normal truncada	(Johari & Mehrabani Iari, 2016)
Macizo	Localización de la grieta de tracción	Normal truncada	(Johari & Mehrabani Iari, 2016)
		Beta	(Hoek & Bray, 1981) (Low, 2008)
		Beta no simétrica	(Low, 2008)
	% Agua en la grieta de tracción	Exponencial truncada	(Johari & Mehrabani Iari, 2017a)
Propiedades del material	Cohesión	Normal	(Kerguelen, 2009)
		Normal truncada	(Johari & Mehrabani Iari, 2016)
		Lognormal	(Duzgun et al., 2002) (Jimenez-Rodriguez et al., 2006) (Griffiths & Fenton, 2007)
		Beta	(Jimenez-Rodriguez et al., 2006) (Low, 2008)
		Normal	(H.-J. Park, 2000) (Griffiths & Fenton, 2007)
	Ángulo de Fricción	Normal truncada	Mostry y Li (1993) (Hoek, 1998a) (Nilsen, 2000a) Pathak y Nilsen (2004) Zheng et al., (2014) (Johari & Mehrabani Iari, 2016)
			Beta
		Peso unitario	Normal truncada

Descripción	Parámetro	Función de densidad de probabilidad (PDF)	Proponente		
	Resistencia a la Compresión	Normal	Barton (1981)		
	Módulo de elasticidad	Normal	Barton (1981)		
	Resistencia al corte	Normal	Barton (1981)		
		Lognormal	(Miller, 1984)		
		Gamma	(Miller, 1983)		
Discontinuidades	Orientación	Fisher	(Fisher, 1953) , (Glynn, 1979)		
		Beta	Chen (2007)		
		Normal	(Miller, 1983) (Müller, 1982)		
		Arnold	(Glynn, 1979)		
		Bingham	(Glynn, 1979)		
		Dershowitz	(Glynn, 1979)		
		Normal	(Hammah et al., 2009)		
	Longitud	Lognormal	McMarton (1974) Bridges (1976) Beacher et al (1978)		
		Exponencial o exponencial negativa	(Robertson, 1970) Call, Savely y Nicholas (1976) Call y otros (1977) Cruden (1977) Beacher y Lanney (1978) (Marek & Savely, 1978) (Glynn, 1979) Wallis y Rey (1980) Herget (1981) Priest Huston (1981) (Einstein & Baecher, 1983) Kulatilake et al (1993)		
			Normal	(Hammah et al., 2009)	
	Lognormal		(H.-J. Park, 2000) Rouleau y Gale (1985) Sen y Kazi (1984) Kulatilake et al (2003)		
			Exponencial o exponencial negativa	Robertson (1970) Call, Savely y Nicholas (1976) Call y otros (1977) Cruden (1977) Marek (1978) (Glynn, 1979) Herget (1981) Herget (1981) Sea y Kazi (1984) Park (1999) Kulatilake (2003)	
				Poisson	(Glynn, 1979)
	Rugosidad			Exponencial o exponencial negativa	Robertson (1970) Call, Savely y Nicholas (1976) Call y otros (1977) Cruden (1977) Marek (1978) Herget (1981) Herget (1981)
		Agua			Exponencial truncada (Hoek, 1998a)
		Sismo			Exponencial truncada (Hoek, 1998a)
					Poisson (Bolle, 1987)

La presentación de diferentes alternativas de procesamiento de los datos referenciados en la tabla que varios autores han implementado en sus investigaciones permite que se tenga una visión amplia de las alternativas más propicias para el tratamiento de los datos de cada uno de los parámetros que usualmente son aplicados en los análisis de estabilidad en taludes en roca. Para la investigación en curso se tendrá en cuenta las propuestas de estudio para el ángulo de fricción, cohesión y sismo.

2.2 Análisis estadístico de las discontinuidades

La orientación de una discontinuidad en el espacio queda definida por su dirección de buzamiento (Azimut o dirección de la línea de máxima pendiente del plano de discontinuidad respecto al norte) y por su buzamiento (inclinación respecto a la horizontal de dicha línea). La definición más precisa de la orientación de cada familia de discontinuidades depende de la suficiente cantidad de datos recolectados en campo de cada una de las discontinuidades identificadas.

El procesamiento de dichos datos e identificación de familias de discontinuidades se puede realizar de manera gráfica, mediante la proyección estereográfica, representando los polos o planos con valores medios de las diferentes familias de discontinuidades; diagramas de rosetas que permiten representar un gran número de medidas de orientación de forma cuantitativa; bloque diagramas, permitiendo una visión general de las familias y sus orientaciones respectivas; símbolos en mapas geológicos, que indican los valores medios de dirección de buzamiento y valor del buzamiento para los diferentes tipos de discontinuidades (juntas, fallas, foliación, etc.) (González de Vallejo et al., 2004).

Otro método para el tratamiento de los datos de las discontinuidades y asignar un dato preferencial o valor medio para definir una familia de discontinuidades es mediante la suma vectorial de la normal de cada una de las discontinuidades. Cada normal puede ser considerada como una unidad vectorial y la orientación preferencial o media de la familia de discontinuidades es el resultado de todos los valores individuales de las discontinuidades. La sumatoria puede ser realizada mediante el cálculo de los cosenos directores propuesto en Goodman, (1989).

- Cálculo de la orientación media

La orientación media será aplicada a una población de orientaciones que se distribuyen de acuerdo con la función de distribución de Fisher. Para la implementación de la distribución se debe calcular como primera medida, una dirección media de las orientaciones de las discontinuidades; luego, parámetros estadísticos como la dispersión de las direcciones, que es análoga a la desviación estándar estimada en la estadística gaussiana y por último, un límite de confianza para la dirección media calculada, análogo al error estándar estimado de la media (Butler, 1992).

Se tiene una serie de datos de orientación de discontinuidades observados e identificados en un macizo rocoso del cual se registran los datos de cada discontinuidad, haciendo referencia a Azimut de buzamiento (α) y buzamiento (β). Para procesar dicha información e interpretar cada uno de los datos, Goodman, (1989) sugiere considerar cada orientación de la discontinuidad como un vector unitario (x,y,z), donde x está orientado horizontalmente y hacia el norte, y horizontalmente y hacia el Este, y z verticalmente hacia arriba.

El resultado de todos los valores individuales de un respectivo grupo representa la orientación promedio o media del grupo de datos. El resultado puede ser obtenido con la sumatoria de los cosenos directores, los cuales son:

$$\begin{aligned} l &= \cos \alpha \cos \beta \\ m &= \cos \beta \sin \alpha \\ n &= \sin \beta \end{aligned} \quad (2-1)$$

Luego de obtener cada uno de los cosenos directores para cada uno de los datos de las discontinuidades, y habiéndolas agrupado en familias, se procede al cálculo de la orientación media de cada una de las familias de discontinuidades, como sigue:

$$l_R = \frac{\sum l_i}{|\bar{R}|} \quad m_R = \frac{\sum m_i}{|\bar{R}|} \quad n_R = \frac{\sum n_i}{|\bar{R}|} \quad (2-2)$$

$$|\bar{R}| = \left[\left(\sum l_i \right)^2 + \left(\sum m_i \right)^2 + \left(\sum n_i \right)^2 \right]^{1/2} \quad (2-3)$$

El valor de la orientación media de la familia de discontinuidades es:

$$\begin{aligned}\beta_R &= \sin^{-1}(n_R) \quad 0 \leq \beta_R \leq 90^\circ \\ \alpha_R &= + \cos^{-1}\left(\frac{l_R}{\cos \beta_R}\right) \quad \text{si } m_R \geq 0 \\ \alpha_R &= - \cos^{-1}\left(\frac{l_R}{\cos \beta_R}\right) \quad \text{si } m_R < 0\end{aligned}\tag{2-4}$$

- Cálculo de la dispersión (k)

Calculada la orientación media de cada una de las familias de discontinuidades, se procede a calcular la media de la dispersión de los datos de las discontinuidades, representada por un parámetro k, designado como constante de Fisher.

$$k = \frac{N - 1}{N - |\bar{R}|}\tag{2-5}$$

Donde N es el número de datos y R es el vector resultante. Se puede estimar la dispersión comparando los resultados del parámetro k con el número N de los datos considerados. Si los valores de las discontinuidades son todos paralelos, los resultados serían iguales a N, mientras que, si fueran altamente dispersos en orientación, los resultados serían menores que N. Si k es muy grande la dispersión es muy pequeña (Goodman, 1989).

- Cálculo del límite de confianza

El límite de confianza para la dirección media calculada es análogo al error estándar estimado de la media de la orientación por las estadísticas gaussianas. Para las estadísticas de Fisher, el límite de confianza se expresa como un radio angular desde la dirección media calculada. Se debe indicar un nivel de probabilidad para que el límite de confianza esté completamente definido (Butler, 1992).

$$\cos \delta_{(1-p)} = 1 - \frac{N - R}{R} \left\{ \left(\frac{1}{p}\right)^{\frac{1}{N-1}} - 1 \right\}\tag{2-6}$$

Donde $\delta_{(1-p)}$ es el ángulo dentro del cual la media verdadera desconocida se encuentra en el nivel de confianza (1-p). La elección habitual del nivel de probabilidad (1-p) es 0.95 o $\delta = 95\%$ $p=5\%$.

- Cálculo de la desviación angular θ

El cálculo de la desviación angular θ el cual es el ángulo entre la orientación media de la discontinuidad y la orientación verdadera de otra discontinuidad, puede ser calculado mediante algebra vectorial, si ambos vectores tienen magnitud unitaria. Una fórmula para calcular el ángulo entre dos vectores según Ragan, (2009) es:

$$\begin{aligned}\hat{u} \cdot \hat{v} &= \cos \theta = l_1 l_2 + m_1 m_2 + n_1 n_2 \\ \theta &= \cos^{-1}(l_1 l_2 + m_1 m_2 + n_1 n_2)\end{aligned}\tag{2-7}$$

Donde (l_1, m_1, n_1) y (l_2, m_2, n_2) son los cosenos directores de cada uno de los vectores que se pueden calcular según lo presentado en las ecuaciones (2-1).

2.3 Cálculo de la probabilidad de Fisher

Para el análisis estadístico de los datos, varios autores han presentado el análisis de las discontinuidades mediante una distribución Fisher, la cual es considerada como la PDF (Función de densidad de probabilidad) más apropiada para representar la distribución del conjunto de orientaciones (Butler, 1992). Esta distribución supone que una población de polos se distribuye alrededor de una orientación verdadera (Fisher, 1953). Mediante el uso de dicha distribución se realizan los análisis de datos en el presente estudio.

La distribución Fisher fue desarrollada por el estadístico británico R.A Fisher, y plantea que a cada dirección se le asigna un peso unitario y se representa por un punto en una esfera de radio unitario. La función de distribución de Fisher $f(\theta)$ suministra la probabilidad, por unidad de área angular, de encontrar una dirección dentro de un área angular centrada en un ángulo θ , que es el ángulo entre la orientación medida y la orientación verdadera.

$$f(\theta) = \frac{k e^{k \cos \theta}}{4\pi \sinh k}\tag{2-8}$$

Donde, θ es la desviación angular de la orientación media en un rango entre 0 y $\pi/2$, k es la constante de Fisher la cual es la medida de la concentración de la distribución sobre la verdadera dirección media; k es 0 para una distribución de direcciones que es uniforme sobre la esfera, y se aproxima a ∞ para direcciones concentradas en un punto.

Luego la probabilidad de encontrar una dirección específica dentro de un área dA será:

$$p = \int_{\varepsilon_0}^{\varepsilon_1} \int_{\theta_0}^{\theta_1} \frac{k e^{k \cos \theta}}{4\pi \sinh k} \sin \theta d\theta d\varepsilon \quad (2-9)$$

Donde $0 \leq \varepsilon \leq 2\pi$ y $0 \leq \theta \leq \pi$

Para ε de 0 a 2π

$$p = \int_{\theta_0}^{\theta_1} \frac{k e^{k \cos \theta}}{2 \sinh k} \sin \theta d\theta \quad (2-10)$$

Y la probabilidad acumulada se calcularía de la siguiente manera:

$$P(\theta) = \int_0^{\theta} \frac{k e^{k \cos \theta}}{2 \sinh k} \sin \theta d\theta \quad (2-11)$$

$$P(\theta) = \frac{e^{k \cos \theta + k} - e^{2k}}{1 - e^{2k}} \quad \text{para } \theta \text{ de } 0 \text{ a } \pi \quad (2-12)$$

$$P(\pi) = 1 \quad (2-13)$$

Y la probabilidad truncada en $0 \leq \theta \leq \pi/2$

$$P_T(\theta) = P(\theta) * \lambda \quad (2-14)$$

$$\frac{1}{\lambda} = \frac{e^k (1 - e^k)}{1 - e^{2k}} \rightarrow \lambda = \frac{e^{-k} (1 - e^{2k})}{1 - e^k} \quad (2-15)$$

Remplazando en la ecuación (2-14) los resultados de las ecuaciones (2-12) y (2-15)

$$P_T(\theta) = \left(\frac{e^{k \cos \theta + k} - e^{2k}}{1 - e^{2k}} \right) \left(\frac{e^{-k} (1 - e^{2k})}{1 - e^k} \right) = \frac{e^k (1 - e^{k(\cos \theta - 1)})}{e^k - 1} \quad (2-16)$$

Si $k \geq 3 \rightarrow e^k - 1 \approx e^k$ se simplifica la ecuación anterior siguiendo una expresión aproximada para la distribución acumulada (H. Park & West, 2000).

$$P(\theta) \approx 1 - e^{k(\cos \theta - 1)} \quad (2-17)$$

El ángulo de la media verdadera dentro de la cual se encuentra un porcentaje elegido de direcciones, también se puede calcular de la distribución de Fisher. Adicional, un ángulo dentro del cual se encuentra un porcentaje de las direcciones es calculado como:

$$\theta_p = \frac{\psi}{\sqrt{k}} \tag{2-18}$$

Donde θ_p es el ángulo dentro de una probabilidad definida $P(\theta)$, ψ es el dividendo que depende de θ_p y k es la constante de Fisher, ecuación (2-5). La ecuación (2-18) se calcula a partir de la ecuación (2-12) ya mencionada, en donde se asume una probabilidad y se calcula el ángulo θ_p . En la gráfica de la Figura 2-1, se presenta la relación entre el termino ψ de la ecuación y la probabilidad de persistencia de un ángulo a calcular.

Por ejemplo, para una probabilidad del 50% la ecuación correspondiente será:

$$\theta_{50} = \frac{67.5^\circ}{\sqrt{k}} \tag{2-19}$$

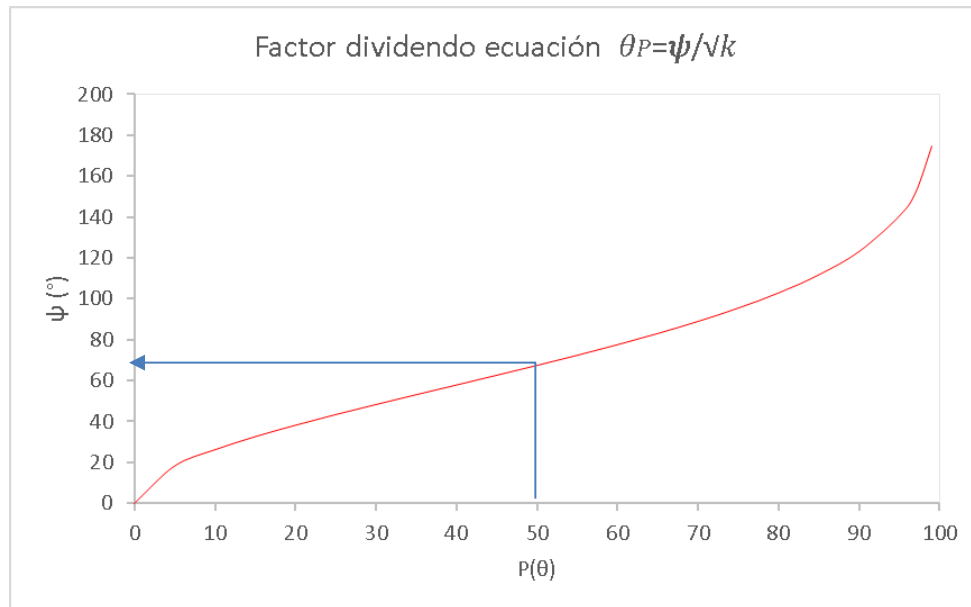


Figura 2-1. Cálculo del factor dividendo ψ de la ecuación (2-18). Fuente: propia

2.4 Análisis estadístico del ángulo de fricción de los planos

La estabilidad de los taludes en roca está afectada por parámetros de obtención aleatoria como el ángulo de fricción que no podrían ser representados por un solo valor para el análisis de la estabilidad. Desde un enfoque probabilístico, el ángulo de fricción se modela como una distribución Normal truncada, Normal o Beta como lo mencionan Hoek, (1998), Nilsen, (2000), Zheng et al., (2014), Johari & Mehrabani Lari, (2017) entre otros, la distribución normal se presenta truncada de tal manera que se pueda limitar el valor mínimo y máximo del ángulo de fricción con los cuales se permite representar con mayor precisión el comportamiento de la variable. La función normal truncada se definirá en dos pasos:

1. Seleccionar los parámetros de la función de distribución normal μ y σ^2 (media y varianza)
2. Luego los límites del truncamiento (a, b)

Dependiendo del rango de truncamiento, se tienen cuatro casos:

- No truncado: $-\infty = a; b = +\infty$
- Truncado por el extremo izquierdo: $-\infty < a; b = +\infty$
- Truncado por el extremo derecho: $-\infty = a; b < +\infty$
- Doblemente truncado: $-\infty < a; b < +\infty$

La función de densidad de probabilidad normal truncada puede ser evaluada como sigue:

$$f(x; \bar{\mu}, \bar{\sigma}, a, b) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{\phi(\bar{\mu}, \bar{\sigma}^2; x)}{\Phi(\bar{\mu}, \bar{\sigma}^2; b) - \Phi(\bar{\mu}, \bar{\sigma}^2; a)} & \text{si } a < x < b \\ 0 & \text{si } b \geq x \end{cases} \quad (2-20)$$

$$F(x; \bar{\mu}, \bar{\sigma}, a, b) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{\Phi(\bar{\mu}, \bar{\sigma}^2; x) - \Phi(\bar{\mu}, \bar{\sigma}^2; a)}{\Phi(\bar{\mu}, \bar{\sigma}^2; b) - \Phi(\bar{\mu}, \bar{\sigma}^2; a)} & \text{si } a < x < b \\ 1 & \text{si } b \geq x \end{cases} \quad (2-21)$$

Donde Φ es la función de distribución normal estándar CDF, y ϕ función de densidad normal estándar PDF.

2.5 Construcción de envolvente de afloramiento para taludes por falla planar

Para identificar una zona crítica de un talud dentro de la red polar en donde se vea la influencia de la probabilidad calculada, se hace necesario la obtención de la zona de dominio de la orientación de las discontinuidades en la cara del talud; para esto, una de las formas más prácticas es la planteada por Richard Lisle en su artículo “*Calculation of the daylight envelope for plane failure of rock slopes*” donde propone la relación que existe entre la evaluación del potencial de falla planar con relación a la orientación de las discontinuidades y la cara del talud. Las discontinuidades que se sumergen en la dirección de la cara del talud son cinemáticamente viables, y son las que pueden aflorar en la cara del talud o *daylight* como se conoce (Lisle, 2004).

Es una práctica común utilizar técnicas de proyección hemisférica para establecer el rango de orientaciones de las discontinuidades de *daylight* a partir de una consideración de la cara del talud en roca según Hoek y Bray, (1981); Bell, (1987); Lisle y Leyshon, (2004). En la proyección, las normales de los planos *daylight* ocupan un área delimitada por una curva conocida como la envolvente de afloramiento. La envolvente se define por los polos de los planos que son marginales en términos de la condición de daylight: estos son planos cuyas líneas descendentes se encuentran dentro del plano de la cara de la roca (Figura 2-2).

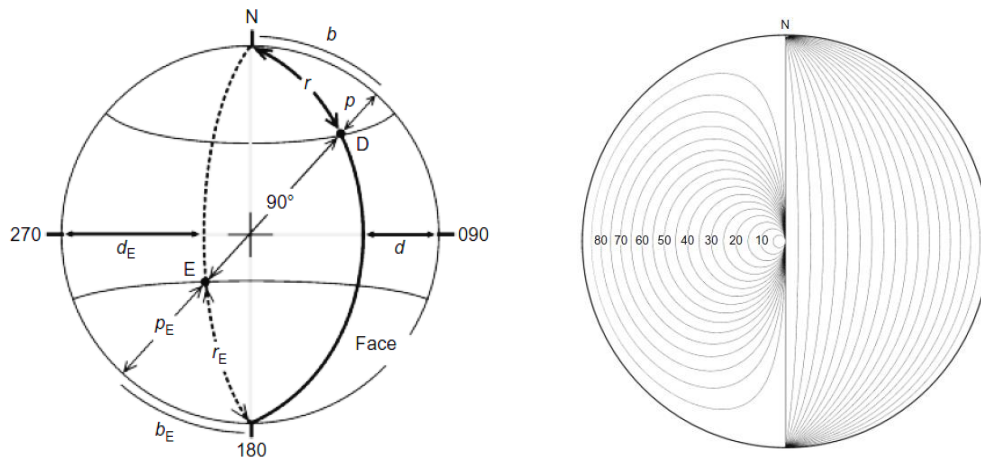


Figura 2-2. A). Definición de ángulos utilizados en la derivación de la fórmula para las coordenadas de puntos en el sobre de luz diurna. B) Ejemplos de zona crítica natural para una serie de taludes de rocas con ángulos de inclinación que van desde 5 ° a 85°. Fuente: Lisle, (2004).

Para la construcción de la envolvente, el autor recomienda seguir la metodología indicada a continuación y la Figura 2-2 (A).

1. Definir los ángulos b y d , calcular p y r con las siguientes ecuaciones:

$$\begin{aligned} \cos r &= \cos b \cos r \\ \tan d &= \frac{\tan p}{\sin b} \\ \sin r &= \frac{\sin p}{\sin d} \end{aligned} \quad (2-22)$$

2. Luego calcular k y las coordenadas x , y de puntos para trazar las curvas dentro de la red polar.

$$\begin{aligned} x &= k \sin r \cos d \\ y &= k \cos r \\ k &= \sqrt{2/(1 + \sin r \sin d)} \end{aligned} \quad (2-23)$$

3. Luego, si se quiere graficar la zona con alguna orientación específica (orientación de talud de estudio) se tiene que seguir los siguientes pasos:

$$\begin{aligned} x_E &= k_E \sin r_E \cos d_E \\ y_E &= k_E \cos r_E \end{aligned} \quad (2-24)$$

$$k_E = \sqrt{2/(1 + \sin r_E \sin d_E)} \quad (2-25)$$

$$\begin{aligned} p_E &= 90 - p \\ b_E &= b \end{aligned} \quad (2-26)$$

$$\begin{aligned} x' &= \cos \theta_{x_E} + \sin \theta_{y_E} \\ y' &= -\sin \theta_{x_E} + \cos \theta_{y_E} \end{aligned} \quad (2-27)$$

Con la ayuda de un programa de cómputo u hoja Excel se obtiene la respectivas “envolventes de afloramiento” mostradas en la Figura 2-2 (B).

2.6 Análisis cinético de confiabilidad

En análisis cinético involucra las fuerzas actuantes en el macizo rocoso para los diferentes mecanismos de falla y la implementación de la probabilidad conjunta de dos variables aleatorias continuas, ver Anexo 2.

Para el análisis cinético de estabilidad de los taludes en roca, se procede con una explicación de la metodología para cada uno de los mecanismos de falla. Comenzando para la condición planar con la similitud a un bloque deslizando sobre un plano inclinado, luego cuña deslizando por dos planos y volteo de bloques.

2.6.1 Bloque deslizando sobre un plano inclinado

La representación de la falla planar como el deslizamiento de un bloque sobre un plano inclinado es la forma básica de análisis de la estabilidad. Mediante el estudio de este caso se pretende dar una interpretación a las fuerzas actuantes, el equilibrio de fuerzas dentro del sistema y analizar las variables independientes de manera probabilística. A forma de esquema en la siguiente figura se representa el bloque deslizando y las respectivas fuerzas actuantes en el sistema.

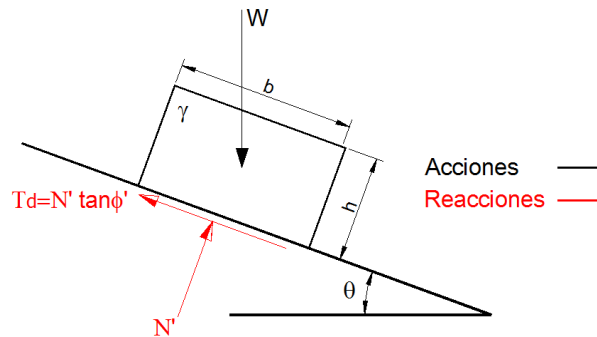


Figura 2-3. Esquema de fuerzas actuantes para bloque deslizando

Del esquema se puede plantear que la probabilidad de falla es la probabilidad de que las acciones sean mayores a las de diseño o que los parámetros de resistencia sean menores que los establecidos.

Para comenzar con la comprobación de dicha premisa se podría inferir primero que las acciones de diseño se definen de manera determinista, asignando valores medios, la mediana o la moda de los respectivos parámetros geotécnicos para ser usados en el análisis de equilibrio límite como se haría normalmente. Otra opción es realizar el análisis de la estabilidad a través de parámetros calculados probabilísticamente.

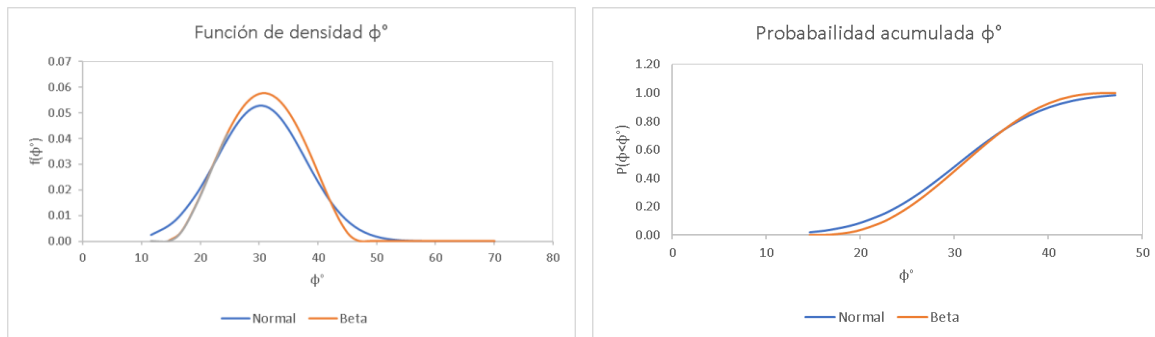


Figura 2-4. Función de densidad de probabilidad y función acumulada de ϕ°

Para la obtención de un valor de cohesión requerido para el equilibrio limite se sugiere tener presente las siguientes deducciones:

$$\text{Si } FS = 1 \rightarrow c' \leftrightarrow c'_r \text{ (Cohesion requerida para el equilibrio)}$$

$$\tan\phi' = \tan\phi'_r \text{ (Angulo de fricción requerido para el equilibrio)}$$

$$1 = \frac{c'_r}{\gamma h} * \frac{1}{\text{sen}\theta} + \frac{\tan\phi'_r}{\tan\theta} \quad NE = \frac{c'_r}{\gamma h} \quad (2-28)$$

$$c'_r = \gamma h \text{sen}\theta \left[1 - \frac{\tan\phi'_r}{\tan\theta} \right] \quad (2-29)$$

De esta forma el valor de la cohesión requerida está en función del ángulo de fricción interno necesario para un factor de seguridad 1.

$$c'_r = f(\phi'_r) \quad \text{para } FS = 1 \quad (2-30)$$

Una gráfica que represente el factor de seguridad en función de la cohesión y el ángulo de fricción requerido es la presentada en la Figura 2-5. Sabiendo que del ángulo de fricción requerido se obtiene el valor de la cohesión requerida para un factor de seguridad de 1, se identifica una zona de factor de seguridad mayor a 1 y otra menor a 1.

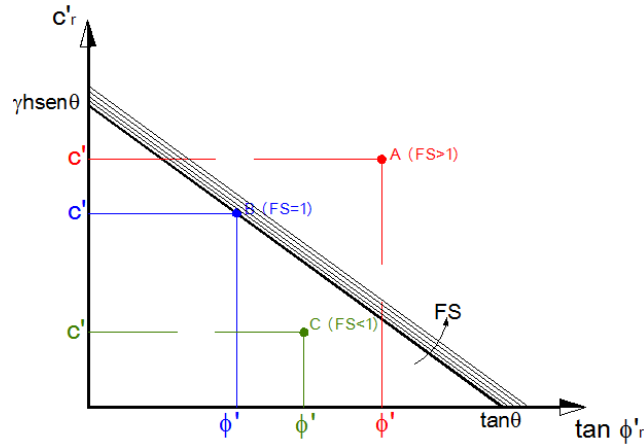


Figura 2-5. Envoltura de factor de seguridad para Cr

Para la obtención de la probabilidad de falla teniendo como variables probabilísticas el ángulo de fricción y la cohesión como factores requeridos para el equilibrio, se plantea mediante la probabilidad conjunta y como el resultado de la operación entre las dos probabilidades obtenidas del respectivo análisis probabilístico con las funciones de densidad de probabilidad para cada parámetro. La operación da como resultado una gráfica en tres dimensiones.

La Figura 2-6 representa, a la izquierda la función de densidad de probabilidad de cohesión, en la parte inferior la función de densidad de probabilidad del ángulo de fricción y a la derecha la envoltura de factor de seguridad y la operación entre la probabilidad de cada función y los contornos de probabilidad de falla para obtener lo que sería el volumen presentado en la Figura 2-7.

Es decir, si c y ϕ son variables independientes cada uno de los valores dentro de la región estará representado por el producto de las dos probabilidades de las variables.

$$P(c \cap \phi) = P(c) \times P(\phi) \tag{2-31}$$

$$P(\phi < \phi_{rj} \cap c < c_{rj}) = P(\phi < \phi_{rj}) * P(c < c_{rj}) \tag{2-32}$$

De la gráfica anterior se puede decir que la fracción del volumen por debajo del Factor de seguridad corresponderá a la probabilidad de falla.

2.6.2 Volteo de bloques (método de Hoek y Bray)

Otro de los métodos a estudiar es el método de volteo de bloques, el cual usa el principio de equilibrio límite al igual que en el bloque deslizante. En la siguiente figura se representa de forma gráfica un bloque sobre un plano inclinado, la representación de las fuerzas actuantes y las fuerzas resistentes dentro del sistema, un bloque con dimensiones relacionadas para que se produzca o no el volteo, o se presente el deslizamiento de dicho bloque.

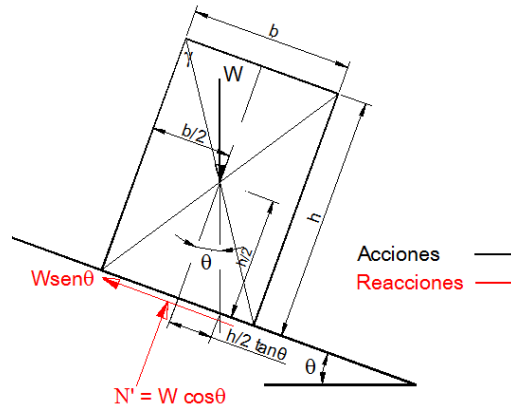


Figura 2-8. Análisis de fuerzas para volteo de bloques

Para que se presente volteo se tendrán que cumplir las siguientes condiciones:

$$\frac{h}{2} \tan \theta \geq \frac{b}{2} \rightarrow \tan \theta > \frac{b}{h} \quad \text{ó} \quad \frac{b}{h} < \tan \theta \quad (2-33)$$

O para que se presente deslizamiento (con solo fricción) se cumplirá la siguiente condición para el cálculo del factor de seguridad resultaría de la siguiente forma:

$$FS = \frac{W \cos \theta \tan \phi}{W \sin \theta} \rightarrow FS = \frac{\tan \phi}{\tan \theta} \quad (2-34)$$

Esto representado para un deslizamiento en donde la condición sea que el ángulo de fricción sea menor o igual a la inclinación del plano de falla $\phi \leq \theta$.

Para el análisis probabilístico de estabilidad por análisis cinético y como en el caso de falla planar, para este caso las variables probabilísticas para el desarrollo del análisis son: la inclinación del plano de falla, el ángulo de fricción entre las discontinuidades y la aceleración sísmica para la zona, para un factor de seguridad igual a 1.

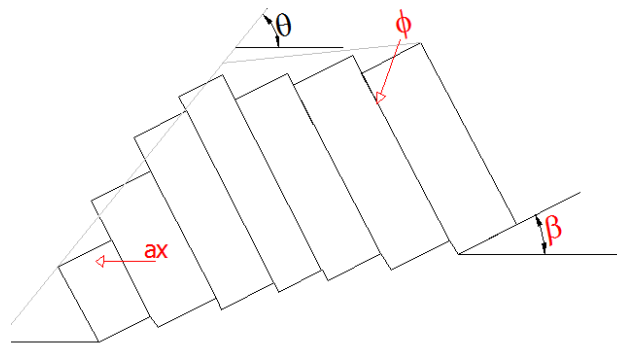


Figura 2-9. Esquema de variables probabilísticas para análisis de estabilidad por volteo

Al igual que con la metodología para la falla planar, la probabilidad se calculará como una probabilidad conjunta entre dos variables, que para este caso se tomará como V_1 la probabilidad de que el ángulo de fricción disponible sea mayor que el requerido, como V_2 será la probabilidad de que el ángulo de inclinación de la superficie de falla disponible sea menor que el requerido y una tercera variable será la probabilidad de la aceleración sísmica.

$$V_1 = P(\varphi > \varphi_d) \quad (2-35)$$

$$V_2 = P(\beta > \beta_d) \quad (2-36)$$

$$V_3 = P(a_x > a_a) \quad (2-37)$$

Para cada una de las variables se obtendrá la función de densidad de probabilidad correspondiente a la función que mejor se ajuste a la serie de datos y a las recomendaciones de funciones de densidad para cada variable. La función de probabilidad para la aceleración sísmica es obtenida a partir del registro del Servicio Geológico Nacional y el Modelo Nacional de Amenaza Sísmica para Colombia y las curvas de amenaza sísmica para varios sitios en Colombia.

El procedimiento para la obtención de la probabilidad conjunta será como se indica en la Figura 2-10, por un lado, se tendrá la función de densidad de la inclinación de la superficie de falla, en el otro eje la función de densidad del ángulo de fricción, y finalmente se obtiene la probabilidad conjunta como el producto de las probabilidades de las dos variables para

obtener la variable V de la ecuación (2-38) , que corresponderá al volumen de la fracción por debajo del ángulo de inclinación de talud seleccionado.

Luego en una tercera grafica se obtendrá la función de distribución de probabilidad para la aceleración sísmica como la variable V_3 , y finalmente para obtener la probabilidad de falla será la operación entre el valor de la variable V (volumen parcial) y la probabilidad de la aceleración sísmica, como se indica en las ecuaciones (2-38) y (2-39) y la Figura 2-10.

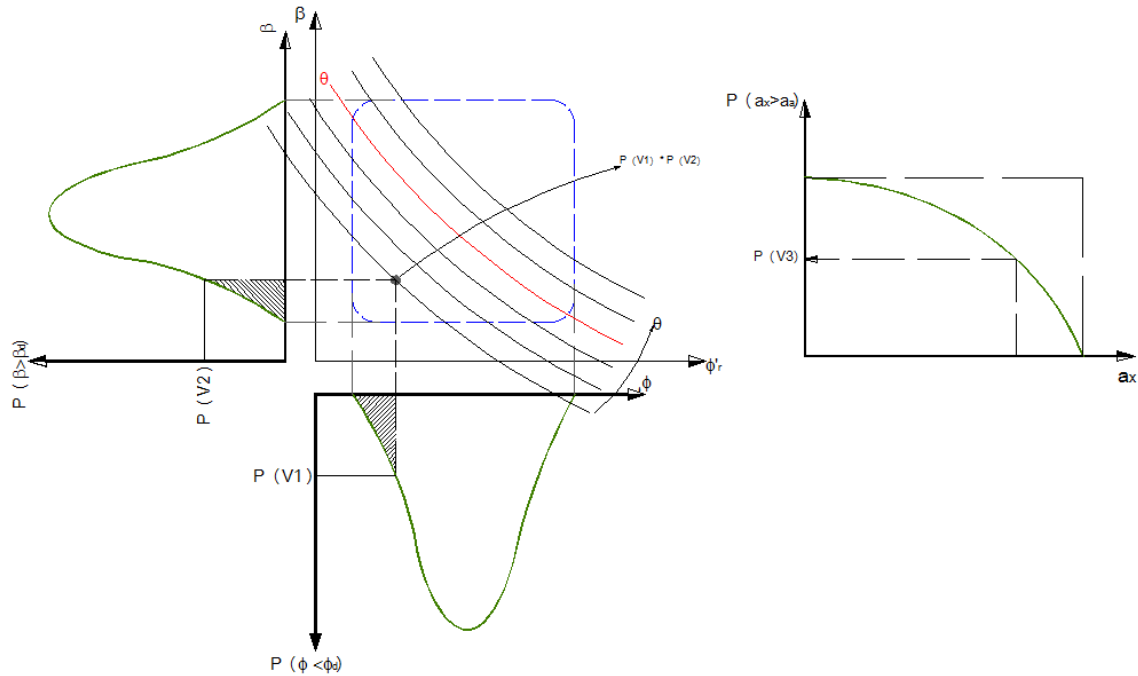


Figura 2-10. Probabilidad conjunta para análisis de estabilidad por volteo

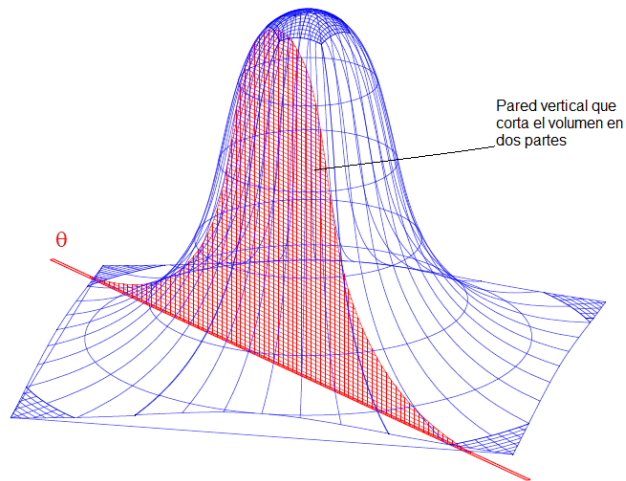


Figura 2-11. Esquema de volumen de probabilidad

$$V = P(V_1) * P(V_2) \quad (2-38)$$

$$P_{falla} = V * P(V_3) \quad (2-39)$$

2.6.3 Cuña deslizante por dos planos que se intersecan

Otro de los mecanismos de falla a tratar es la falla por cuña para un bloque formado por la intersección de dos planos de discontinuidades y que se desliza por ambos planos. Dentro de los análisis de estabilidad para cada mecanismo se presentan varias incertidumbres que van desde las propiedades físicas y mecánicas específicas del macizo rocoso, las discontinuidades, las acciones y las reacciones que intervienen en la estabilidad de un talud dado.

Como variables que intervienen en la estabilidad para este caso se tendrán en cuenta parámetros de resistencia tales como: el ángulo de fricción, cohesión y acciones detonantes al deslizamiento como la aceleración sísmica. Para el análisis probabilístico se tiene la premisa que la probabilidad de falla se define como la probabilidad conjunta de que los parámetros de resistencia disponibles sean menores que los requeridos y que las acciones sean mayores que las previstas en el diseño.

Para la definición de la presencia de cuñas se tendrá en cuenta los postulados mencionados por Hoek y Bray, (1981), y la posibilidad de que se presente falla por cuña, ya sea que se presente movimiento sobre la intersección de los planos o sobre alguno de los planos.

Para el desarrollo del análisis de estabilidad para cuña, se tendrá en cuenta el desarrollo de la propuesta de Granados, (2009), Probabilidad de falla de cuñas en roca, y que es presentado en resumen a continuación.

- Análisis de estabilidad.

Conociendo el buzamiento, β , y azimut de buzamiento, α , de cada uno de los planos que definen la cuña, es posible encontrar el vector normal unitario que define un plano:

$$\mathbf{n} = (\cos\alpha \operatorname{sen}\beta)\hat{i} + (\operatorname{sen}\alpha \operatorname{sen}\beta)\hat{j} - (\cos\beta)\hat{k}$$

De esta manera, y conociendo la altura del talud, es posible establecer las ecuaciones de cada uno de los planos, así mismo es posible encontrar los vértices de la cuña, calculando los puntos en donde se intersecan estos planos, mediante la resolución del sistema:

$$\begin{pmatrix} a_l & b_l & c_l \\ a_m & b_m & c_m \\ a_n & b_n & c_n \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} d_l \\ d_m \\ d_n \end{pmatrix}$$

Donde, a_l es la expresión que acompaña a x en la ecuación del Plano l ; b_l a y , c_l a z , y d_l es el término independiente de la ecuación. Obteniendo las coordenadas de los vértices de las cuñas, es posible encontrar la dirección de la línea de intersección sobre la cual se da el movimiento, así como el volumen y el peso de la cuña:

$$i_{12} = \frac{((x_O - x_R), (y_O - y_R), (z_O - z_R))}{\sqrt{((x_O - x_R)^2 + (y_O - y_R)^2 + (z_O - z_R)^2)}}$$

$$V = \frac{1}{6} \begin{vmatrix} 1 & x_O & y_O & z_O \\ 1 & x_P & y_P & z_P \\ 1 & x_Q & y_Q & z_Q \\ 1 & x_R & y_R & z_R \end{vmatrix}$$

$$W = V\gamma_t$$

- Determinación del factor de seguridad.

Se consideran las acciones del peso propio, del empuje del agua sobre la cuña y de una fuerza sísmica horizontal, que serán soportadas por las fuerzas normales y cortantes en los planos 1 y 2, debidas a la cohesión y a la fricción en los mismos. Definiendo las acciones y reacciones de manera vectorial:

$$\begin{aligned} \vec{F}_w &= W(0, 0, 1) & \vec{T}_1 &= T_1(-i_{12x}, -i_{12y}, -i_{12z}) \\ \vec{F}_a &= aW(i_{ax}, i_{ay}, 0) & \vec{T}_2 &= T_2(-i_{12x}, -i_{12y}, -i_{12z}) \\ \vec{N}_1 &= N_1(n_{1x}, n_{1y}, n_{1z}) & \vec{F}_{\omega 1} &= F_{\omega 1}(n_{1x}, n_{1y}, n_{1z}) \\ \vec{N}_2 &= N_2(n_{2x}, n_{2y}, n_{2z}) & \vec{F}_{\omega 2} &= F_{\omega 2}(n_{2x}, n_{2y}, n_{2z}) \end{aligned}$$

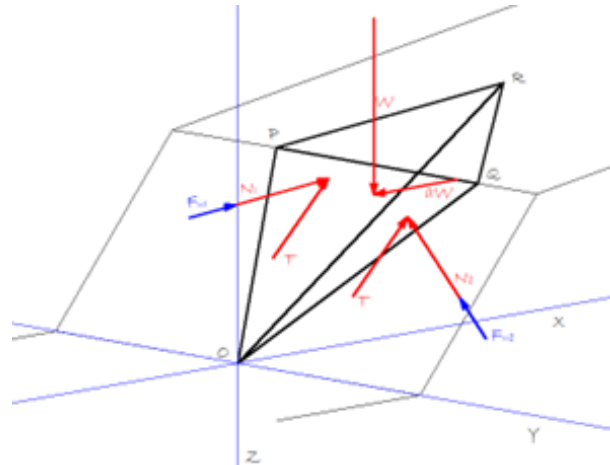


Figura 2-12. Acciones y reacciones sobre la cuña. Fuente: Granados, (2009).

Para considerar la acción del agua sobre los planos 1 y 2; se supone que el agua llega a cierta altura ηH , como se observa en la Figura 2-12 de este modo se forma un prisma de presiones como los representados en la figura, para determinar la magnitud de la acción ocasionada por esta sobre la cuña, basta con multiplicar el volumen de esta por el peso específico del agua.

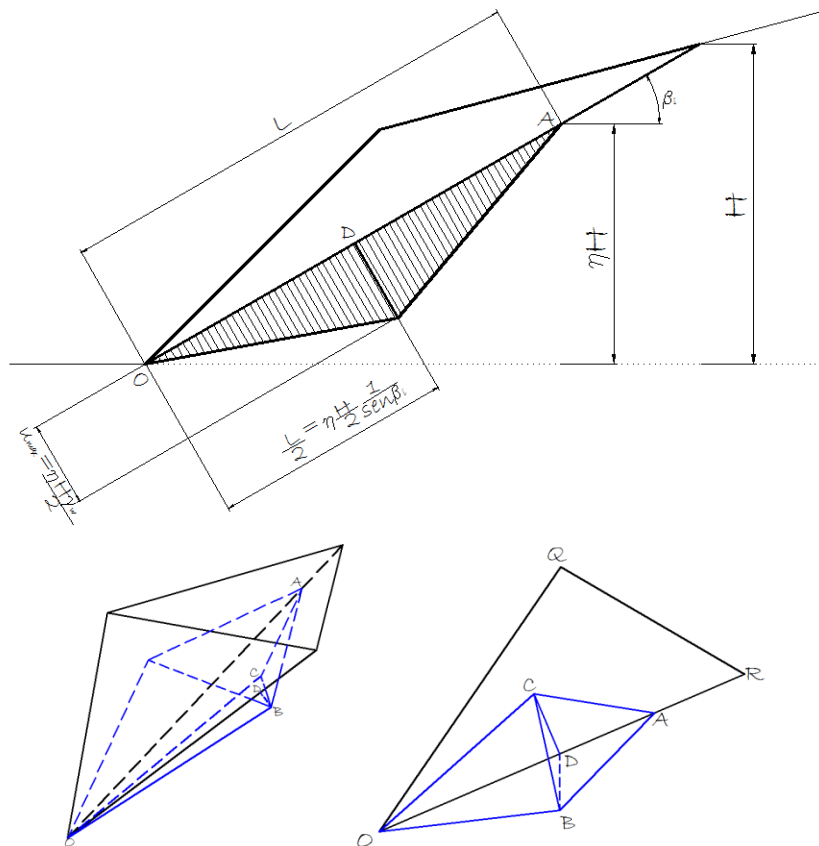


Figura 2-13. Prisma de Presiones sobre la cuña. Fuente: Granados, (2009).

La fuerza cortante requerida se encuentra entonces resolviendo el sistema:

$$\begin{bmatrix} n_{1x} & n_{2x} & i_{12x} \\ n_{1y} & n_{2y} & i_{12y} \\ n_{1z} & n_{2z} & i_{12z} \end{bmatrix} \begin{pmatrix} N_1 \\ N_2 \\ T_{req} \end{pmatrix} = \begin{pmatrix} aWi_{ax} + F_{\omega 1}n_{1x} + F_{\omega 2}n_{2x} \\ aWi_{ay} + F_{\omega 1}n_{1y} + F_{\omega 2}n_{2y} \\ W + F_{\omega 1}n_{1z} + F_{\omega 2}n_{2z} \end{pmatrix}$$

La fuerza cortante disponible, será entonces la adición de las fuerzas cohesivas y friccionales debidas a las propiedades propias de las discontinuidades y el macizo rocoso en que se encuentra el talud. Esta fuerza está dada por la ecuación:

$$T_{disp} = c_1'A_1 + c_2'A_2 + N_1\tan\phi_1' + N_2\tan\phi_2'$$

Y el factor de seguridad, de acuerdo con el método de las fuerzas, será:

$$FS = \frac{T_{disp}}{T_{req}}$$

- Análisis probabilístico.

Para el análisis probabilístico por cuña Granados, (2009) tiene en cuenta los parámetros de resistencia de cohesión y ángulo de fricción como parámetros probabilísticos, y la aceleración sísmica de cierta magnitud, y como en el caso anterior se procede mediante el principio de equilibrio limite y el método de fuerzas para obtener la probabilidad de falla como la probabilidad de que los parámetros de resistencia sean menores que los requeridos y que las acciones (aceleración, agua), sean mayores a las empleadas en el diseño.

$$P(T_{dis} < T_{req}) = (P(C < C_r) \vee P(R_{\phi} < R_{\phi r})) \wedge (P(a > a_d)) \quad (2-40)$$

Si se asume que el FS = 1, y se expresan los valores de las fuerzas cohesiva y friccional requeridas de la siguiente manera:

$$T_{req} = T_{disp} = c_1'A_1 + c_2'A_2 + N_1\tan\phi_1' + N_2\tan\phi_2' = C + R_{\phi} \quad (2-41)$$

Donde $C = c_1'A_1 + c_2'A_2$ y $R_{\phi} = N_1\tan\phi_1' + N_2\tan\phi_2'$

Luego se aproxima un valor de C_r y $R_{\phi r}$ y se calculan los valores de c_{1r}' y c_{2r}' ,

$$C_r = T_{req} * \eta C = c_{1r}'A_1 + c_{2r}'A_2 \quad (2-42)$$

$$R_{\phi r} = T_{req} * \eta R_{\phi} = N_1 \tan \phi_{1r'} + N_2 \tan \phi_{2r'} \quad (2-43)$$

$$c_{1r'} = -\frac{A_2}{A_1} c_{2r'} + \frac{T_{req} * \eta C}{A_1} \quad (2-44)$$

$$\tan \phi_{1r'} = -\frac{N_2}{N_1} \tan \phi_{2r'} + \frac{T_{req} * \eta R_{\phi}}{N_1} \quad (2-45)$$

Donde $\eta R_{\phi} = R_{\phi} / T_{disp}$ y $\eta C = C / T_{disp}$

Si se conocen los valores estadísticos como la media y la desviación estándar de los parámetros $c_{1'}$, $c_{2'}$, $\phi_{1'}$ y $\phi_{2'}$, y siguiendo las distribuciones de probabilidad recomendadas en la Tabla 2-1 para estos parámetros, se encontrara el valor máximo y mínimo posible para cada uno de los parámetros, de acuerdo a las siguientes expresiones y gráficamente, estos valores se representaran como se muestran en la y Figura 2-15.

$$c_{1' min} = c_{1'} - \sigma_{c_1} \frac{\sqrt{28}}{2} \quad (2-46)$$

$$c_{1' max} = c_{1'} + \sigma_{c_1} \frac{\sqrt{28}}{2} \quad (2-47)$$

$$\phi_{1' min} = \phi_{1'} - \sigma_{\phi_1} \frac{\sqrt{28}}{2} \quad (2-48)$$

$$\phi_{1' max} = \phi_{1'} + \sigma_{\phi_1} \frac{\sqrt{28}}{2} \quad (2-49)$$

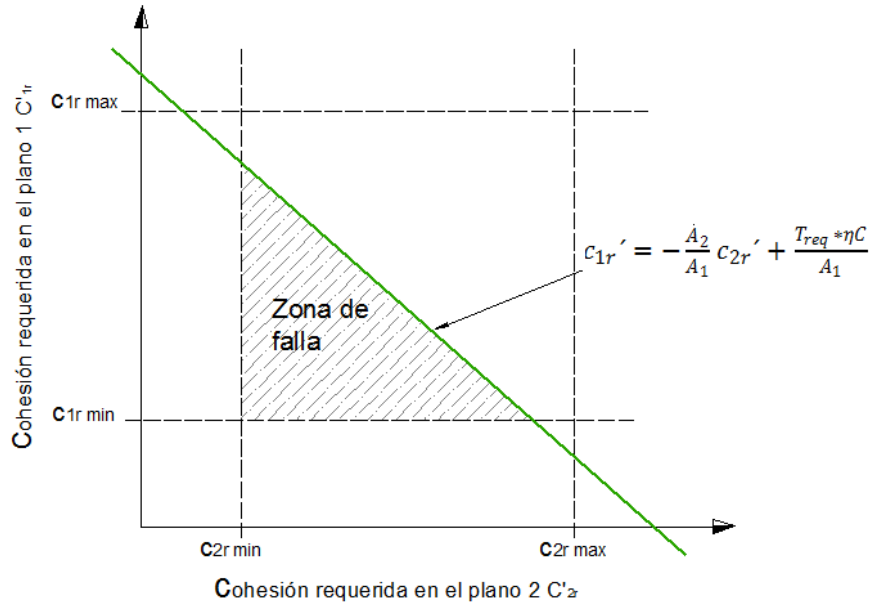


Figura 2-14. Zona de falla para cohesión Fuente: Granados, (2009)

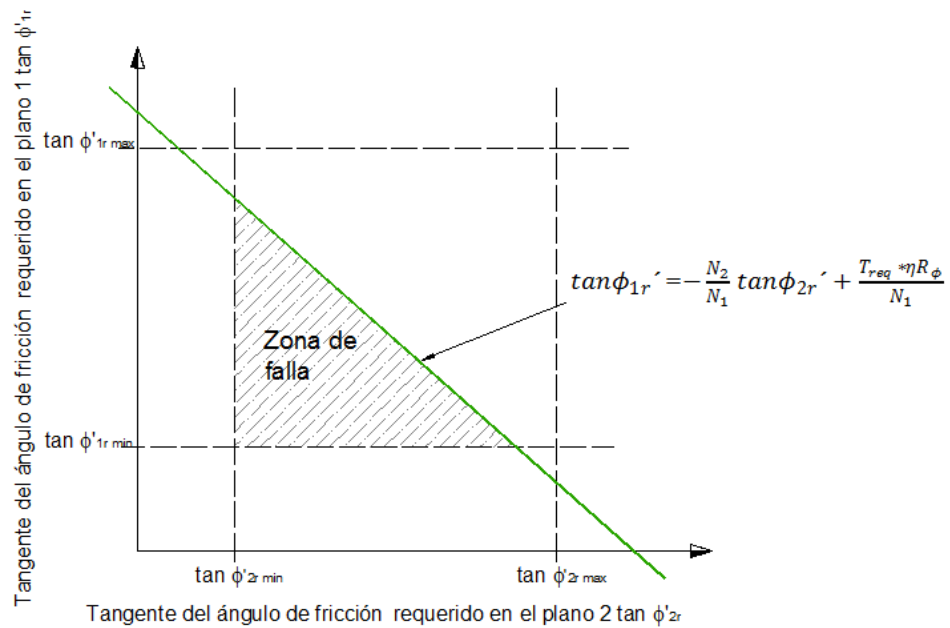


Figura 2-15. Zona de falla para ángulo de fricción Fuente: Granados, (2009)

Teniendo las distribuciones de probabilidad de c_1' , c_2' , ϕ_1' y ϕ_2' , la probabilidad de que $(C < C_r)$ o $(R_\phi < R_{\phi r})$, será el volumen de probabilidades sobre el área de la zona de falla. La probabilidad será definida por las siguientes ecuaciones:

$$P(C < C_r) = \int_A \{P(c_1' < c_{1r}') * P(c_2' < c_{2r}')\} dA \quad (2-50)$$

$$P(R_\phi < R_{\phi_r}) = \int_A \{P(\phi_1' < \phi_{1r}') * P(\phi_2' < \phi_{2r}')\} dA \quad (2-51)$$

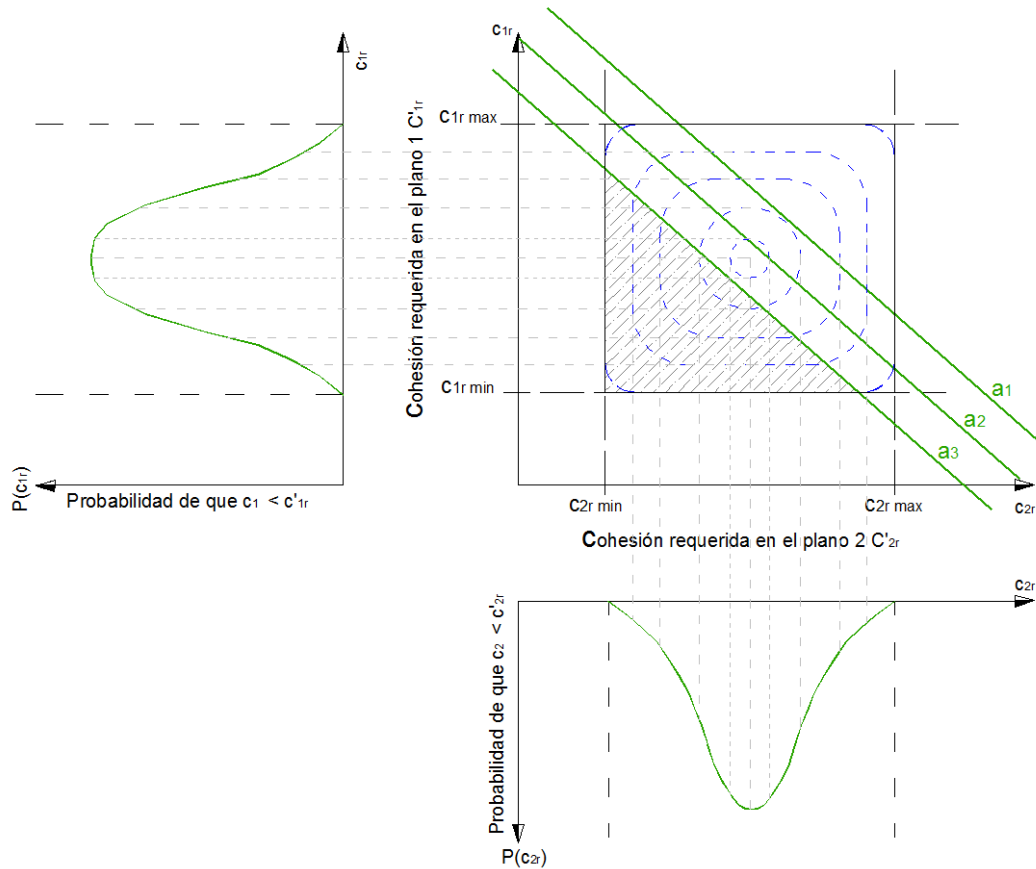


Figura 2-16. Volumen de Probabilidades sobre el área de falla Fuente: Granados, (2009)

La probabilidad de que $T_{dis} < T_{req}$ dado a un valor a de aceleración sísmica horizontal será la probabilidad de que los parámetros de resistencia sean menores que los requeridos. De igual forma se procede con el ángulo de fricción para obtener el volumen parcial de probabilidades.

$$P(C < C_r) = \sum_{i=1}^n V_i \quad (2-52)$$

$$P(R_{\phi} < R_{\phi r}) = \sum_{i=1}^n V_i \quad (2-53)$$

$$P(T_{disp} < T_{req}) = P(C < C_r) * P(R_{\phi} < R_{\phi r}) \quad (2-54)$$

Considerando la probabilidad de ocurrencia de un sismo con determinada intensidad, la probabilidad de falla final será:

$$P_{falla} = [P(C < C_r) * P(R_{\phi} < R_{\phi r})] * P(a > a_d) \quad (2-55)$$

3. Metodología propuesta

Con la revisión de la literatura técnica se identifica que en los últimos años diferentes métodos para el análisis de estabilidad de taludes dentro de un enfoque probabilístico han sido planteados. Sin embargo, pocos han propuesto un análisis estadístico y probabilístico de las variables usualmente empleadas en estabilidad de taludes mediante representaciones gráficas; es por eso que surge la idea de proporcionar una metodología alternativa para el análisis de estabilidad de taludes en roca mediante análisis de probabilidad conjunta y representación en red estereográfica como una herramienta para optimizar los análisis estadísticos y probabilísticos ineludibles dentro del criterio de decisión de los diferentes escenarios en la ingeniería de rocas.

3.1 Metodología para la evaluación probabilística

El presente capítulo muestra una metodología alternativa de implementación de métodos probabilísticos de estabilidad de taludes para los diferentes tipos de mecanismos de falla de taludes en roca, comenzando con planteamientos estadísticos y probabilísticos para los diferentes mecanismos de falla. Se propone involucrar el enfoque a partir de los análisis estadísticos de las orientaciones de las discontinuidades, del ángulo de fricción y la relación de estos en la estabilidad cinemática en cada uno de los mecanismos de falla y finalmente, el cálculo de la probabilidad conjunta de las zonas de falla (planar, en cuña o volteo según sea el caso). En el gráfico de la Figura 3-1, se presenta de manera esquemática la metodología propuesta, en donde se identifica las herramientas preexistentes usadas y que son base en el desarrollo de la metodología que se mostrará más adelante.

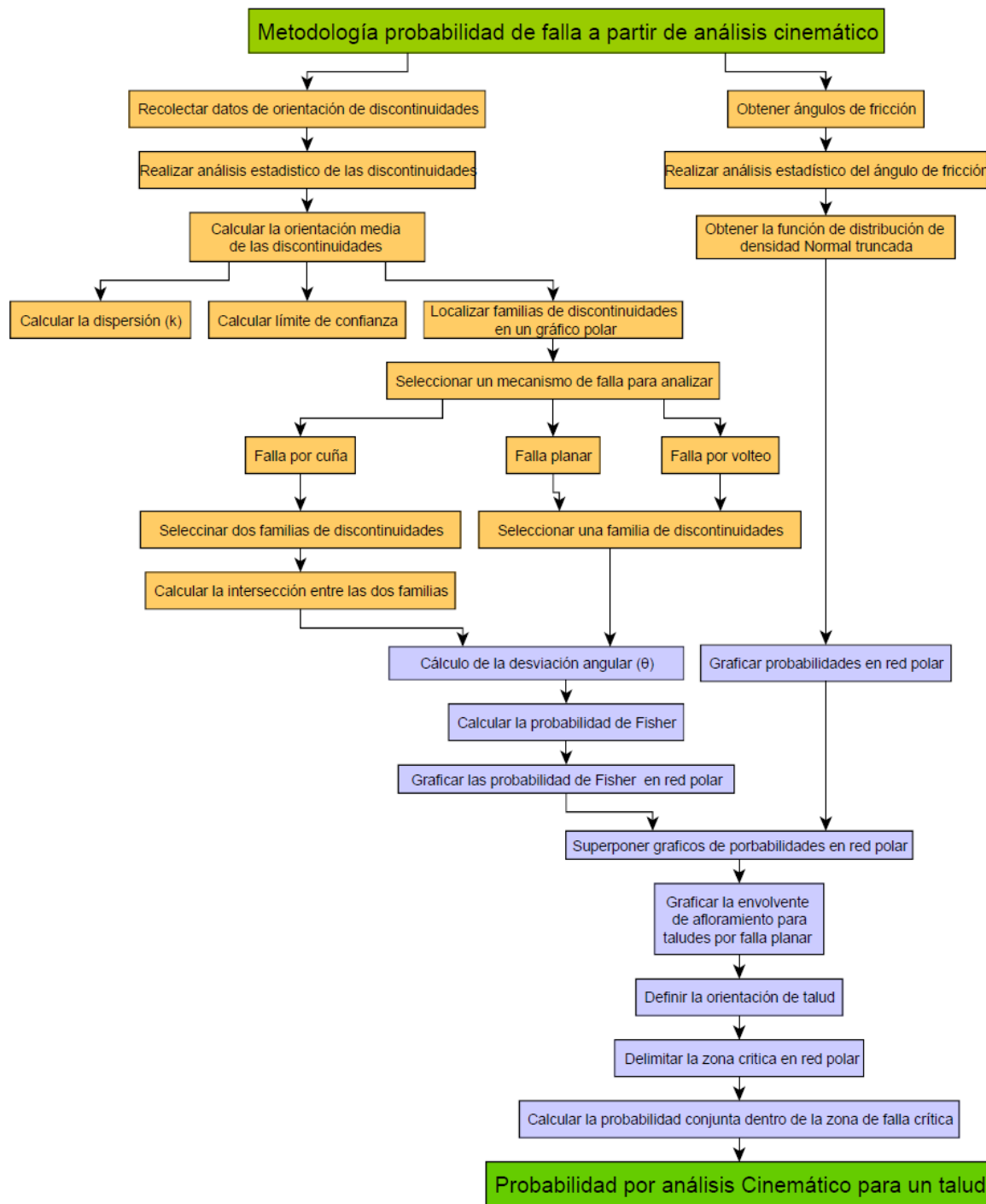


Figura 3-1. Metodología propuesta para análisis probabilístico cinemático de falla. Fuente: propia

La metodología que se está planeando articula y toma ventaja de las herramientas disponibles que usualmente se emplean en el análisis geotécnico de taludes con una

propuesta de evaluación de estabilidad de fallas con control cinemático en taludes en roca mediante la implementación de probabilidad conjunta de variables como la orientación de las discontinuidades y el ángulo de fricción con la representación gráfica en red polar y la definición de contornos de probabilidad.

Esta metodología se diferencia de otros métodos existentes en que plantea un análisis estadístico y probabilístico de las orientaciones de las discontinuidades con la función de probabilidad de Fisher que es una función bivariada, representa el ángulo de fricción no como un solo valor sino como una zona de probabilidad en la red estereográfica definida por la función de densidad de probabilidad normal truncada y limitado por una envolvente de la orientación de un talud crítico permitiendo obtener una probabilidad de falla respecto a la probabilidad de Fisher de cada una de las familias de discontinuidades identificadas; adicionalmente la metodología se complementa con un análisis cinético por confiabilidad de forma gráfica para cada uno de los mecanismos de falla en taludes en roca.

3.1.1 Probabilidad conjunta en red polar

A continuación, se presenta para cada uno de los mecanismos de falla a estudiar el procedimiento que se aplica en cada caso y el fragmento de código desarrollado para cada paso.

- **Procedimiento para falla planar**

1. Inicialmente se deberá contar con datos estructurales de las diferentes discontinuidades del talud rocoso que se quiere estudiar, para los cuales se les realizará análisis estadístico para la obtención de los valores medios representativos de cada una de las familias encontradas según lo descrito en la base teórica del numeral 2.2. Como parte del código desarrollado, a continuación, se presenta el fragmento con el cual se define cada una de las familias de discontinuidades.

```
def agrupar_datos(dataframe: pd.DataFrame) -> pd.DataFrame:
    # Organizar datos por azimut
    sorted_df = dataframe.sort_values('alfa', ascending=True)
    # Hallar la diferencia entre los alfa_i y alfa_cero
    alfa_cero = sorted_df.iloc[0]['alfa']
    sorted_df['d1'] = sorted_df['alfa'] - alfa_cero
    familia = 1
    # Hallar la diferencia entre el dato actual y el anterior
    d2 = sorted_df['d1']
    sorted_df['d2'] = sorted_df['d1'] - d2.shift(1)
```

```

sorted_df['familia'] = 0
# Si la diferencia es mayor a 20 entonces asignar a nueva familia
for idx, row in sorted_df.iterrows():
    if row['d2'] > 22:
        familia = familia + 1
        sorted_df.at[idx, 'familia'] = familia
# print('\nAgrupados\n', sorted_df)
return sorted_df

def cosenos_directores(datos_df: pd.DataFrame) -> pd.DataFrame:
    """
    Calcula los cosenos directores del azimut y buzamiento
    :param datos_df: dataframe. alfa y beta en grados, Familia
    :return: dataframe. l, m, n, Familia
    """
    cos_directores = pd.DataFrame()
    # Cambiar de grados a radianes
    datos_rads_df = np.deg2rad(datos_df[['alfa', 'beta']])
    # Calcular cosenos directores
    cos_directores['l'] = np.cos(datos_rads_df['alfa']) *
np.cos(datos_rads_df['beta'])
    cos_directores['m'] = np.cos(datos_rads_df['beta']) *
np.sin(datos_rads_df['alfa'])
    cos_directores['n'] = np.sin(datos_rads_df['beta'])
    cos_directores['familia'] = datos_df['familia']
    # print('\nCosenos directores\n', cos_directores)
    return cos_directores

def sumatoria_cos_directores(cos_directores_df: pd.DataFrame) ->
pd.DataFrame:
    # Agrupar datos por familia
    group_directores = cos_directores_df.groupby(by='familia')
    # Sumar los cosenos de cada familia
    suma_directores = group_directores.sum()
    # Agregar el número de datos de cada familia
    suma_directores['N'] = group_directores.size()
    # print('\nSumatoria cos directores\n', suma_directores)

    return suma_directores

def calculo_r(suma_directores: pd.DataFrame) -> pd.DataFrame:
    r_df = pd.DataFrame()
    # Elevar la suma de los cosenos directores al cuadrado
    r_df = suma_directores[['l', 'm', 'n']] ** 2
    # Calcular la raíz de la suma de los cuadrados de cada familia
    r_df['R'] = np.sqrt(r_df[['l', 'm', 'n']].sum(axis=1))
    r_df['N'] = suma_directores['N']
    # print('\nRs\n', r_df)
    return r_df

def calculo_representativos(r_df: pd.DataFrame, suma_directores_df:
pd.DataFrame) -> pd.DataFrame:

```

```

# Cargar l, m, n en representativos
representativos_df = suma_directores_df
# Cargar R de r_df en representativos
representativos_df['R'] = r_df['R']
# Dividir l, m,n de cada familia por el R de cada familia
representativos_df = representativos_df[['l', 'm',
'n']].div(representativos_df['R'], axis='index')
# Calcular representativos
representativos_df['beta'] = np.arcsin(representativos_df['n'])

def calcular_alfa_f(row):
    af = np.arccos(row['l'] / np.cos(row['beta']))
    if row['m'] >= 0:
        return af
    else:
        return -af
representativos_df['alfa'] =
representativos_df.apply(calcular_alfa_f, axis=1)
# Cambiar alfa y beta a grados
representativos_df[['alfa', 'beta']] =
np.rad2deg(representativos_df[['alfa', 'beta']])
representativos_df['alfa_p'] =
representativos_df['alfa'].apply(alfa_a_alfa_p)
representativos_df['beta_p'] =
representativos_df['beta'].apply(beta_a_beta_p)
representativos_df['N'] = r_df['N']
representativos_df['R'] = r_df['R']
# print('\nRepresentativos\n', representativos_df[['alfa_p',
'beta_p']])

return representativos_df

```

2. Obteniendo los valores medios o el valor de orientación representativo de cada familia, se grafica en una red estereográfica polar cada una de las familias de discontinuidades como se muestra en la Figura 3-2.

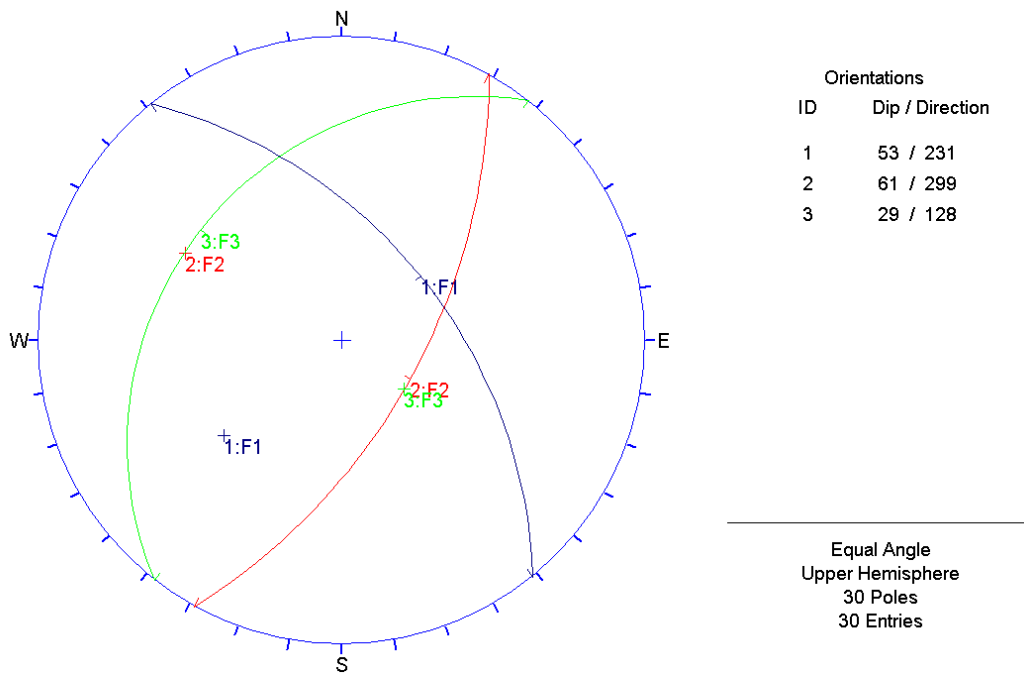


Figura 3-2. Red polar, localización planos de familias de discontinuidades Fuente: Resultados familias de discontinuidades ejercicio Goodman, (1989).

2. Luego, con la ayuda de un programa de graficación de datos en red polar y teniendo los respectivos cálculos de las probabilidades de Fisher de cada una de las orientaciones de la red como se mencionó en el numeral 2.3, se grafican dichos valores y se generan contornos con los valores similares de probabilidad de Fisher. La Figura 3-3 indica contornos que va desde el cero absoluto hasta los valores más altos de probabilidad cercanos a la familia de discontinuidad.

```
def paleom(red_df: pd.DataFrame, alfa_p, beta_p, k) -> pd.DataFrame:
    """
    Calcula la probabilidad usando la función de paleomagnetismo
    :param red_df: dataframe. alfa, beta, Red polar
    :param alfa_p: alfa de la familia
    :param beta_p: beta de la familia
    :param k: coeficiente de la familia
    :return: dataframe. alfa, beta, fp
    """
    df = red_df[['alfa', 'beta']]
    cos_dir_flia = cosenos_directores(pd.DataFrame({'alfa': alfa_p,
    'beta': beta_p}, index=[0]))
    cos_dir = cosenos_directores(red_df)
    cos_theta = (cos_dir * cos_dir_flia.iloc[0]).sum(axis=1)
    df['fp'] = k / (4 * np.pi * np.sinh(k)) * np.exp(k * cos_theta)
    min_fp = df['fp'].max() * 0.01
    df.loc[df['fp'] <= min_fp, 'fp'] = 0
    return df
```

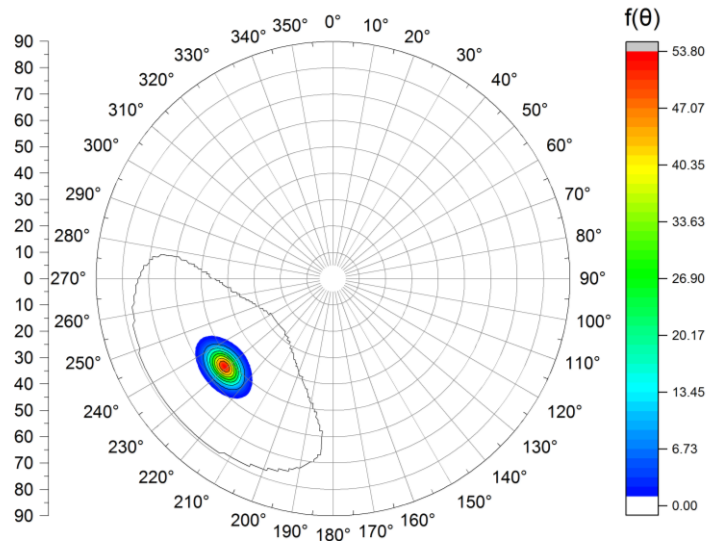



Figura 3-3. Contornos de probabilidades con respecto a familia F1 Fuente: propia

3. Al involucrar el ángulo de fricción dentro del sistema, se tiene que la mejor representación estadística de los datos del ángulo de fricción es la función de densidad de probabilidad Normal truncada, para la cual se calcula inicialmente los descriptores de la función normal, el valor de la función de distribución de los valores máximo y mínimo o los valores del truncamiento, y luego se calcula los valores de la función normal truncada para cada dato; dicho procedimiento fue explicado con más detalle en el numeral 2.4. En la Figura 3-4 se representa una función típica normal truncada para algunos datos de ángulo de fricción.

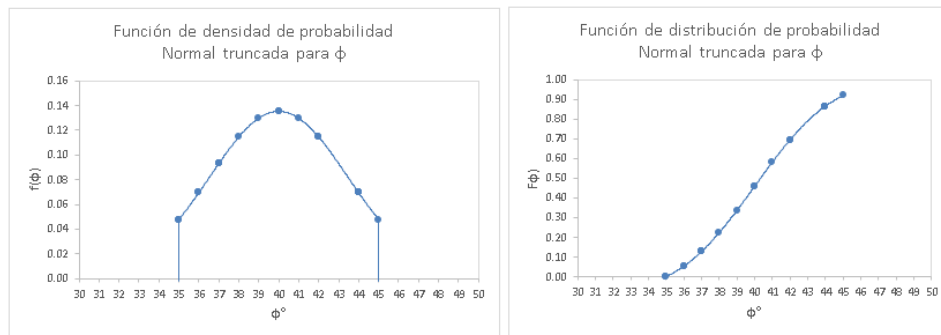


Figura 3-4. Función de densidad de probabilidad para el ángulo de fricción Fuente: propia

4. De la misma forma en la que se grafica la probabilidad de las orientaciones, los valores de la probabilidad del ángulo de fricción se representan en una red polar, como se muestra en la Figura 3-5.

```
def fric(red: pd.DataFrame, phi: pd.DataFrame) -> pd.DataFrame:
    mean = phi.mean().iloc[0]
    std = phi.std().iloc[0]
    minim = phi['phi'].min()
    # maxim = phi['phi'].max()
    maxim = 90
    a, b = (minim - mean) / std, (maxim - mean) / std
    pdf = red[['alfa', 'beta']]
    pdf['pdf'] = truncnorm.pdf(red['beta'], a, b, loc=mean, scale=std)
    # print(pdf)
    return pdf
```

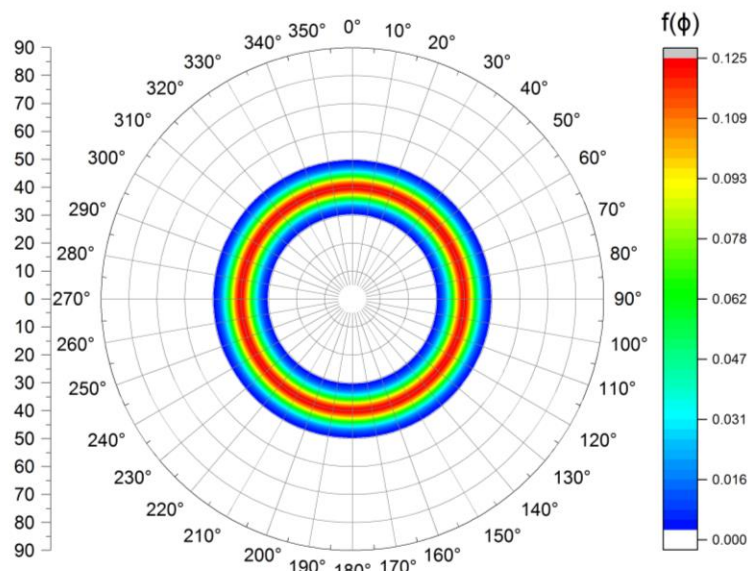


Figura 3-5: Probabilidad de ángulo de fricción. Fuente: propia

5. Continuando con el procedimiento, se realiza el trazado de la envolvente de talud, la cual se obtiene como se describió en el numeral 2.5. Mediante el siguiente fragmento de código se realiza la discretización e identificación de la envolvente que evaluara la zona de falla según la orientación de talud escogida.

```
def beta_min(red_df: pd.DataFrame, paleom_df, alfa_t):
    def iterar(beta_0, beta_1, partes):
        print(f'Buscando beta mínimo entre {beta_0} y {beta_1}')
        paso = (beta_1 - beta_0) // partes
        if paso == 0:
            paso = 1
        suma_env = 0.
        # rango = list()
        # print(rango)
```

```
for b in range(beta_0, beta_1 + 1, paso):
    env_df = envolvente3d(red_df, b, alfa_t)
    env_df['env'] = paleom_df['fp'] * env_df['env']
    suma_1 = suma_env
    suma_env = env_df['env'].sum()
    # print(b, suma_env, suma_1)
    if suma_env > suma_1:
        beta_1 = b
    else:
        if suma_env > 0:
            break
# print(beta_1 - paso, beta_1)
if paso > 1:
    beta_1 = iterar(beta_1 - paso, beta_1, partes)
return beta_1

return iterar(0, 90, 5)
```

6. Con base en las respectivas graficas de probabilidad y la envolvente graficada, se requiere obtener una zona en común para el cálculo de la probabilidad conjunta de falla planar. Se presenta a continuación en la Figura 3-6 la superposición de las gráficas mencionadas para un análisis de talud con la orientación de diseño. Las envolventes se proyectan cada 5° sobre la orientación seleccionada y el trazado de la zona critica de falla se puntualiza con la envolvente que abarque la totalidad de la función de probabilidad de Fisher graficada, según como se muestra en la Figura 3-6 y en la Figura 3-7.

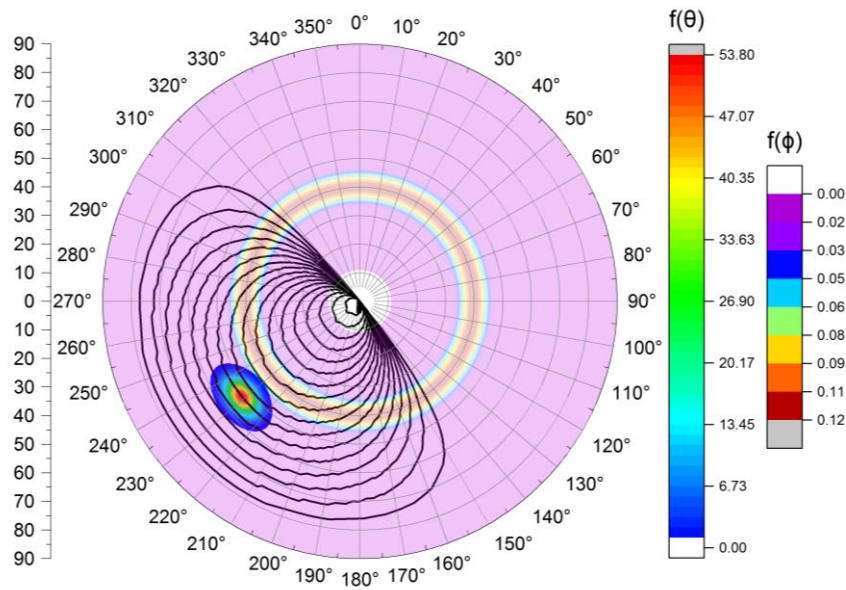


Figura 3-6. Envolturas y cono de probabilidad falla planar. Fuente: propia

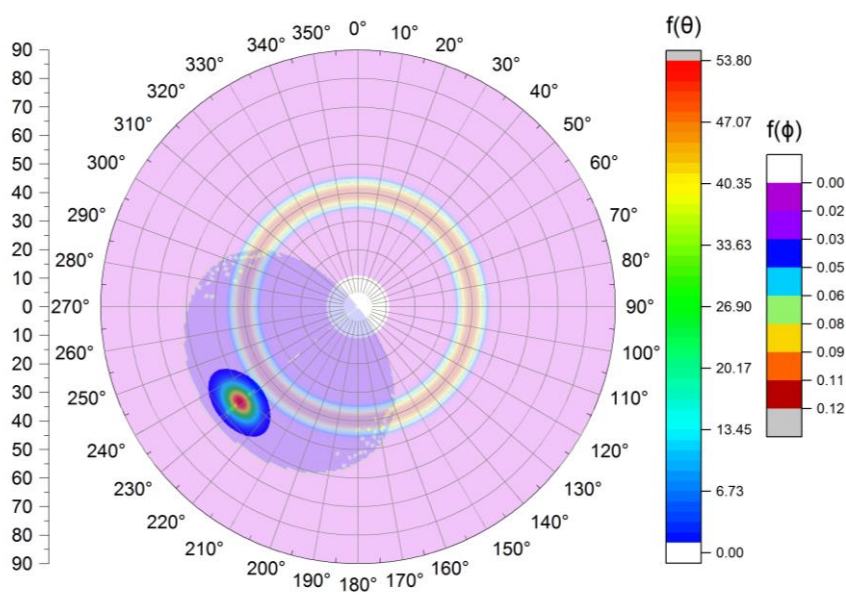


Figura 3-7. Zona de falla critica falla planar. Fuente: propia

7. Para el cálculo de la probabilidad conjunta o probabilidad de falla se tiene, la densidad de probabilidad por la orientación de la familia, la densidad de probabilidad por el ángulo de fricción y la delimitación por la envolvente de talud. La probabilidad conjunta será el producto de dichas probabilidades análogo al concepto de modelos compuestos, el cual

establece que todos los componentes individuales que afectan la probabilidad de la falla deben compilarse para calcular la probabilidad de falla (Glynn, 1979).

$$P_{falla} = \sum (f(\theta) * f(\varphi)) = 1 \quad (3-1)$$

```
def cinematica_planar(datos: DataCtl, datos_out: DataCinematica):
    print('[Cinemática planar]', 'Calculando representativos de
familias')
    familia_df = fam.data_familias(datos.datos_rumbos)
    print('[Cinemática planar]', 'Calculando cono de fricción')
    fric_df = frc.fric(datos.red_polar, datos.datos_friccion)
    datos_out.friccion_df = fric_df[['alfa', 'beta', 'pdf']]
    areas_df = dut.areas(datos.red_polar)

    for idx, row in familia_df.iterrows():
        print(f'Familia {idx}'.center(100, '-'))
        print(f'[Cinemática planar]', 'Calculando probabilidad con la
función paleomagnética')
        paleom_df = pal.paleom(datos.red_polar, row['alfa_p'],
row['beta_p'], row['k'])
        print('[Cinemática planar]', 'Buscando beta de la envolvente')
        beta_envolvente = env.beta_min(datos.red_polar, paleom_df,
datos.cinemat_planar.ang_talud)
        print('Beta de la envolvente:', beta_envolvente)
        env_df = env.envolvente3d(datos.red_polar, beta_envolvente,
datos.cinemat_planar.ang_talud)
        area_envolvente = (env_df['env'] * areas_df['a']).sum()
        print('Área envolvente:', area_envolvente)
        print('[Cinemática planar]', 'Calculando probabilidad conjunta')
        paleom_df['fp_a'] = paleom_df['fp'] * areas_df['a']
        constante = paleom_df['fp_a'].sum()
        paleom_df['fp_a'] = paleom_df['fp_a'] / constante
        paleom_df['conj'] = paleom_df['fp_a'] * env_df['env'] *
fric_df['pdf']
        prob_cojunta = paleom_df['conj'].sum()
        max_level = paleom_df['fp'].max()

        labels = Labels()

        labels.titulo = f'Zona de falla critica falla planar por F{idx}'
        labels.titulo_2 = f'Probabilidad conjunta de falla planar por
F{idx}'

        data_familia = DataCinematicaFamilias(prob_cojunta, max_level,
env_df[['alfa', 'beta', 'env']],
                                                paleom_df[['alfa', 'beta',
'fp', 'conj']], idx, labels)

        if datos_out.calculos_familia is None:
            datos_out.calculos_familia = [data_familia]
        else:
            datos_out.calculos_familia.append(data_familia)
```

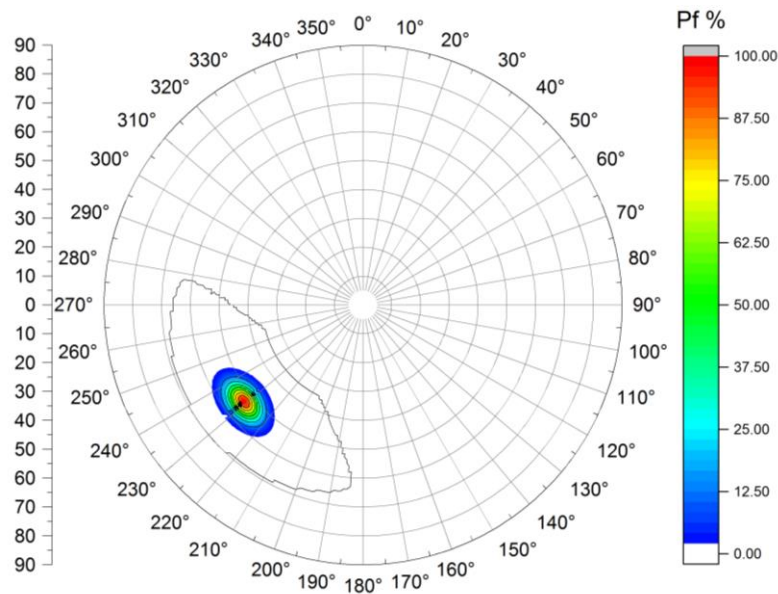


Figura 3-8. Probabilidad conjunta de falla planar Fuente: propia

La Figura 3-8 indica la probabilidad conjunta de falla planar para diferentes orientaciones de talud próximas a la orientación media de la familia F1.

- **Procedimiento para falla en cuña**

El deslizamiento en cuña, al igual que el deslizamiento planar hace parte de los mecanismos más probables en un macizo rocoso dado la influencia de las diferentes orientaciones y familias de discontinuidades presentes en el macizo rocoso, este mecanismo puede comportarse como un deslizamiento planar más complejo.

Al igual que la falla planar, la falla en cuña describe algunas condiciones para que pueda presentarse, y ocurre cuando un bloque de roca se desliza a través de la intersección de al menos dos discontinuidades, las cuales buzcan hacia la cara del talud en un ángulo oblicuo hacia la cara del corte del talud formando un bloque en forma de cuña. Comúnmente, estas cuñas de roca son expuestas por excavaciones que descubren la línea de intersección que corresponde al eje del deslizamiento, precipitando el movimiento del bloque en uno u otro plano de discontinuidad expresa Armas-Zagoya, (2004).

1. Para el análisis de falla por cuña mediante análisis cinemático, inicialmente se calculará la intersección de las familias de discontinuidades, determinando una intersección media representativa y luego se calcula la probabilidad de falla de las diferentes orientaciones por la intersección usando la Probabilidad de Fisher como en el caso de falla planar, finalmente se grafica la probabilidad en la red polar.

```
def interseccion_familias(datos_df: pd.DataFrame) -> pd.DataFrame:
    azimuts_df = rumbo2azimut_df(datos_df)
    familias_df = agrupar_datos(azimuts_df)[['alfa', 'beta', 'familia']]
    idx_familias = range(1, familias_df['familia'].max() + 1)
    perm = combinations(idx_familias, 2)
    inter_df = pd.DataFrame()

    for x, y in perm:
        familia_1 = familias_df[familias_df['familia'] == x]
        familia_2 = familias_df[familias_df['familia'] == y]

        for idx, row in familia_1.iterrows():
            df = familia_2[['alfa', 'beta']].rename(columns={'alfa':
'alfa_b', 'beta': 'beta_b'})
            df[[f'alfa_a', f'beta_a']] = row[['alfa', 'beta']]
            df['inter'] = f'{x}-{y}'
            inter_df = inter_df.append(df, ignore_index=True)

    inter_rad_df = np.deg2rad(inter_df[['alfa_a', 'beta_a', 'alfa_b',
'beta_b']])
    inter_rad_df['inter'] = inter_df['inter']
    inter_rad_df['tan_beta_a'] = np.tan(inter_rad_df['beta_a'])
    inter_rad_df['tan_beta_b'] = np.tan(inter_rad_df['beta_b'])
    inter_rad_df['alfa_ab'] = np.arctan((inter_rad_df['tan_beta_a'] *
np.cos(inter_rad_df['alfa_a']) -
inter_rad_df['tan_beta_b'] *
np.cos(inter_rad_df['alfa_b'])) /
(inter_rad_df['tan_beta_b'] *
np.sin(inter_rad_df['alfa_b']) -
inter_rad_df['tan_beta_a'] *
np.sin(inter_rad_df['alfa_a'])))

    inter_rad_df.loc[inter_rad_df['alfa_ab'] < 0, 'alfa_ab'] =
inter_rad_df.loc[
inter_rad_df['alfa_ab'] < 0, 'alfa_ab'] + np.pi
    inter_rad_df['beta_ab'] = inter_rad_df['tan_beta_b'] *
np.cos(inter_rad_df['alfa_b'] - inter_rad_df['alfa_ab'])
    inter_rad_df.loc[inter_rad_df['beta_ab'] < 0, 'beta_ab'] = -1 *
inter_rad_df.loc[
inter_rad_df['beta_ab'] < 0, 'beta_ab']

    # Aproximar ángulos a cero decimales
    inter_df['alfa_ab'] = np.round(np.rad2deg(inter_rad_df['alfa_ab']))
    inter_df['beta_ab'] = np.round(np.rad2deg(inter_rad_df['beta_ab']))

    # Calculo de cosenos directores
    inter_rad_df['l'] = np.cos(inter_rad_df['alfa_ab']) *
```

```

np.cos(inter_rad_df['beta_ab'])
    inter_rad_df['m'] = np.cos(inter_rad_df['beta_ab']) *
np.sin(inter_rad_df['alfa_ab'])
    inter_rad_df['n'] = np.sin(inter_rad_df['beta_ab'])

# Calculo de sumatoria de cosenos directores
suma_directores_df = sumatoria_cos_directores(
    inter_rad_df[['l', 'm', 'n', 'inter']].rename(columns={'inter':
'familia'}))

# Cálculo de R, representativos y k
r_df = calculo_r(suma_directores_df)
repr_df = calculo_representativos(r_df, suma_directores_df)
k_df = calcular_k(repr_df)

rta_df = pd.DataFrame()
rta_df[['alfa_p', 'beta_p']] = repr_df[['alfa_p', 'beta_p']]
rta_df['k'] = k_df['K']
# print(rta_df)
return rta_df

```

2. Empleando la misma metodología para ángulo de fricción indicado anteriormente, se grafica en red polar la probabilidad de ángulo de fricción, y la envolvente de falla para este caso se proyecta por la orientación de la intersección, luego se grafica dicha zona en red polar y se calcula la probabilidad conjunta para el mecanismo de falla por cuña.

3. El resultado de la metodología propuesta para la intersección por F1-F2 se plasma en la Figura 3-9 a la Figura 3-12, donde se puede identificar la zona crítica para falla en cuña según los contornos y los límites trazados por los condicionamientos antes mencionados.

```

def cinematica_cunia(datos: DataCtl, datos_out: DataCinematica):
    print('[Cinemática por cuña]', 'Calculando representativos de
familias')
    familia_df = fam.interseccion_familias(datos.datos_rumbos)
    print('[Cinemática por cuña]', 'Calculando cono de fricción')
    fric_df = frc.fric(datos.red_polar, datos.datos_friccion)
    areas_df = dut.areas(datos.red_polar)
    datos_out.friccion_df = fric_df[['alfa', 'beta', 'pdf']]

    for idx, row in familia_df.iterrows():
        print(f'Intersección {idx}'.center(100, '-'))
        print(f'[Cinemática por cuña]', 'Calculando probabilidad con la
función paleomagnética')
        paleom_df = pal.paleom(datos.red_polar, row['alfa_p'],
row['beta_p'], row['k'])
        print('[Cinemática por cuña]', 'Buscando beta de la envolvente')
        beta_envolvente = denv.beta_min(datos.red_polar, paleom_df,
datos.cinemat_cunnia.ang_talud)
        print('Beta de la envolvente:', beta_envolvente)
        env_df = denv.envolvente3d(datos.red_polar, beta_envolvente,
datos.cinemat_cunnia.ang_talud)

```



```

area_envolvente = (env_df['env'] * areas_df['a']).sum()
print('Área envolvente:', area_envolvente)
print('[Cinemática por cuña]', 'Calculando probabilidad
conjunta')
paleom_df['fp_a'] = paleom_df['fp'] * areas_df['a']
constante = paleom_df['fp_a'].sum()
paleom_df['fp_a'] = paleom_df['fp_a'] / constante
paleom_df['conj'] = paleom_df['fp_a'] * env_df['env'] *
fric_df['pdf']
prob_cojunta = paleom_df['conj'].sum()
max_level = paleom_df['fp'].max()

labels = Labels()

labels.titulo = f'Zona de falla critica falla en cuña por {idx}'
labels.titulo_2 = f'Probabilidad conjunta falla en cuña por
{idx}'

datos_familia = DataCinematicaFamilias(prob_cojunta, max_level,
env_df[['alfa', 'beta', 'env']],
paleom_df[['alfa',
'beta', 'fp', 'conj']], idx, labels)

print('Probabilidad conjunta:', prob_cojunta)

if datos_out.calculos_familia is None:
    datos_out.calculos_familia = [datos_familia]
else:
    datos_out.calculos_familia.append(datos_familia)

```

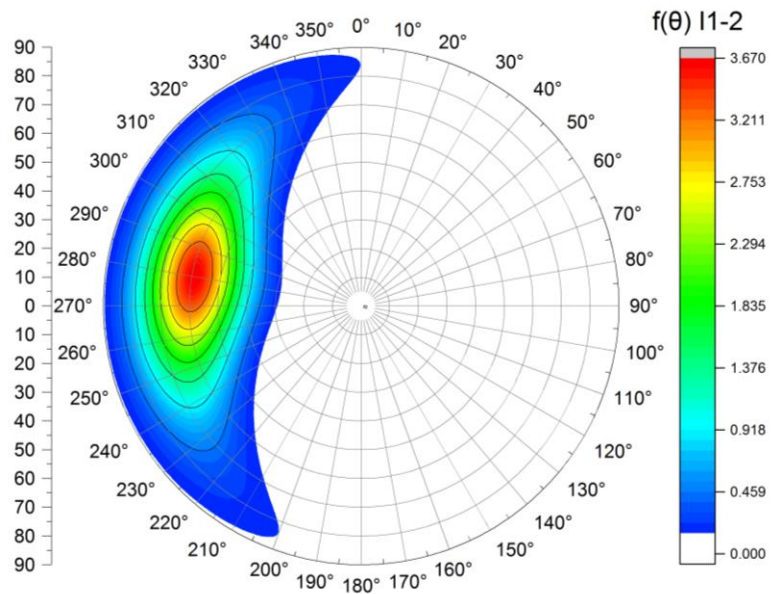


Figura 3-9. Contornos de probabilidades con respecto a intersección F1-F2. Fuente: propia

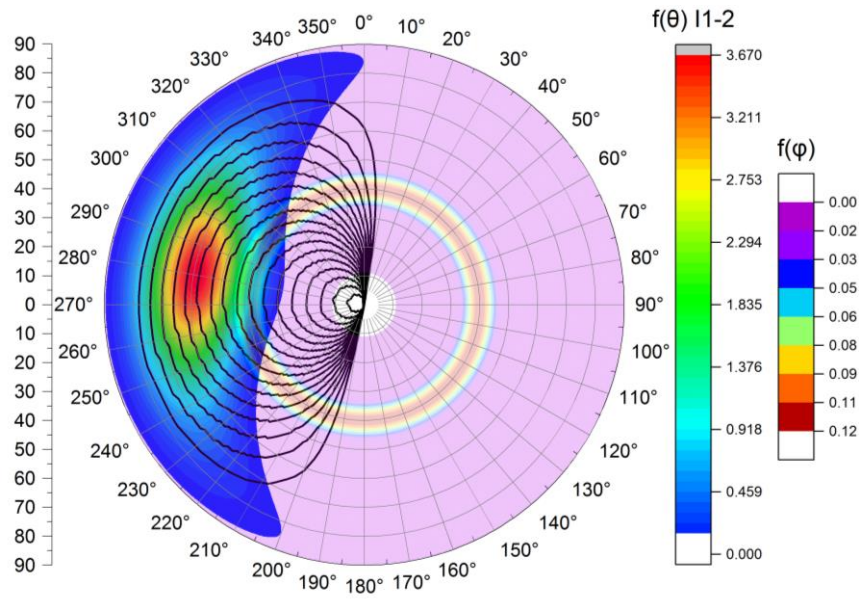


Figura 3-10. Envoltentes y cono de probabilidad falla en cuña por F1-F2. Fuente: propia

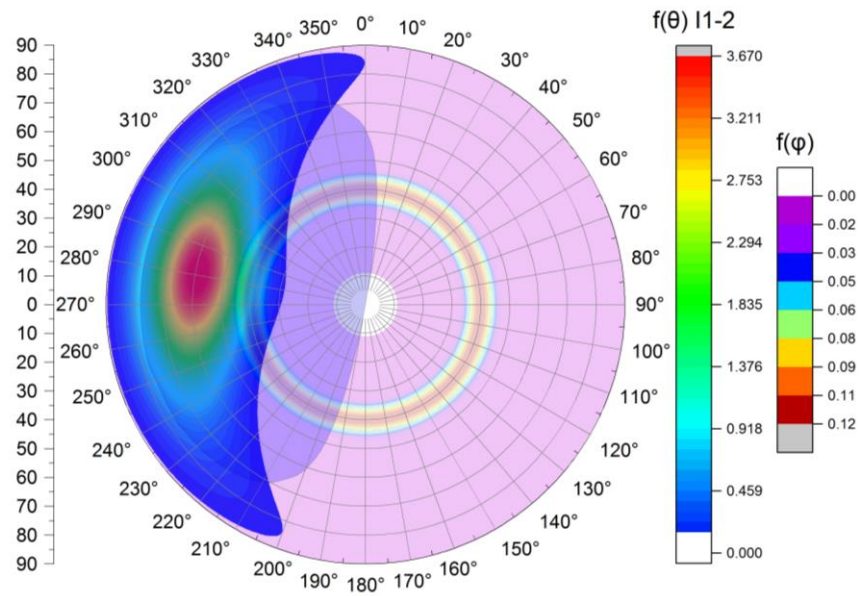
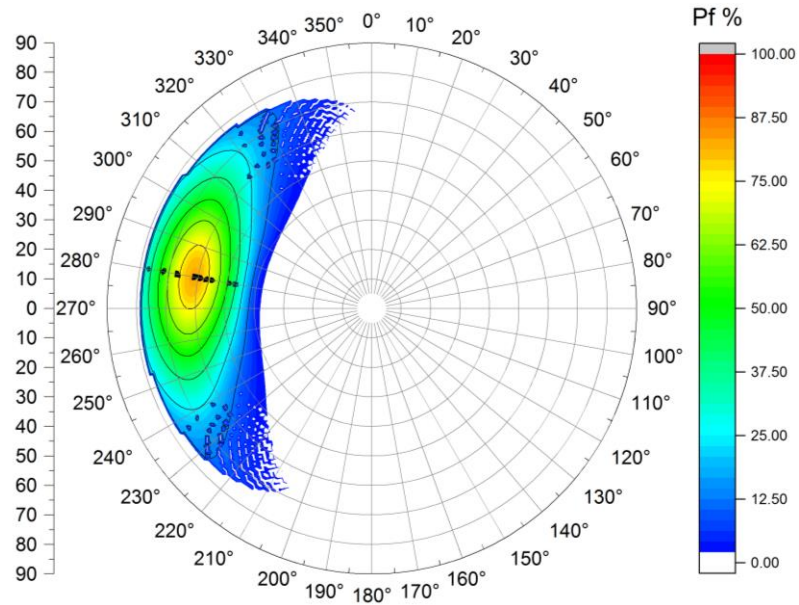


Figura 3-11. Zona de falla crítica falla en cuña por F1-F2. Fuente: propia



. Figura 3-12. Probabilidad conjunta falla en cuña por F1-F2. Fuente: propia

El procedimiento presentado para falla planar se aplica para el mecanismo de falla por volteo como se indicó en la Figura 3-1.

3.2 Análisis cinético de confiabilidad

Para el análisis de estabilidad de taludes en roca es necesario realizar la evaluación cinemática y cinética. Realizado el análisis probabilístico para la condición cinemática con respecto a la orientación de los taludes, la orientación de las discontinuidades para los diferentes mecanismos de falla, obtener una probabilidad de falla y determinar si es posible la inestabilidad cinemática, se plantea como complemento dentro de la metodología el análisis cinético de confiabilidad para cada uno de los mecanismos de falla a estudiar y donde se involucrarán las fuerzas actuantes en el macizo rocoso. A continuación, se presenta el procedimiento planteado según la metodología propuesta.

- **Procedimiento para falla planar**

Para el desarrollo de la metodología se plantea el siguiente procedimiento para llegar al resultado final de una probabilidad conjunta en el análisis probabilístico mediante estabilidad cinética para falla planar.

1. Dentro del análisis se tendrán los datos de laboratorio con resultados de ángulo de fricción y cohesión de varias muestras a los cuales se les realiza análisis estadístico.
2. Teniendo los datos se realiza la representación gráfica de los datos mediante histogramas que da una representación de la distribución de los datos y una idea de la posible función de densidad a usar en el análisis probabilístico.
3. Con los datos de ángulo de fricción, se calcula un valor de cohesión requerido C_r para un factor de seguridad igual a 1 con el uso de la ecuación (2-29).

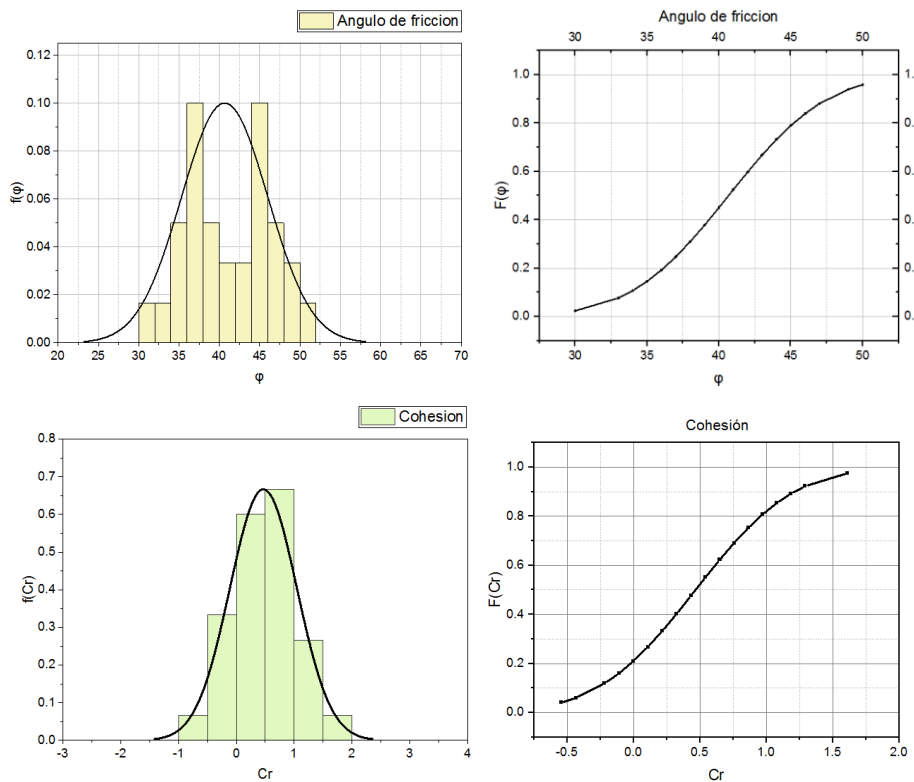


Figura 3-13. Histogramas y funciones de probabilidad Normal para ángulo de fricción y cohesión. Fuente: propia

4. Para los datos del ángulo de fricción y para los valores de C_r , se calcula para cada variable la función de densidad y la función de distribución, que según algunos autores y lo registrado en la Tabla 2-1 para estos parámetros las funciones que pueden representar mejor el comportamiento son la función de distribución normal, normal truncada y función Beta.

Luego como se expresa en la Figura 2-6 en el diagrama de funciones de probabilidad conjunta, se tiene por un lado la función de densidad del ángulo de fricción y por el otro lado la función de densidad de la cohesión.

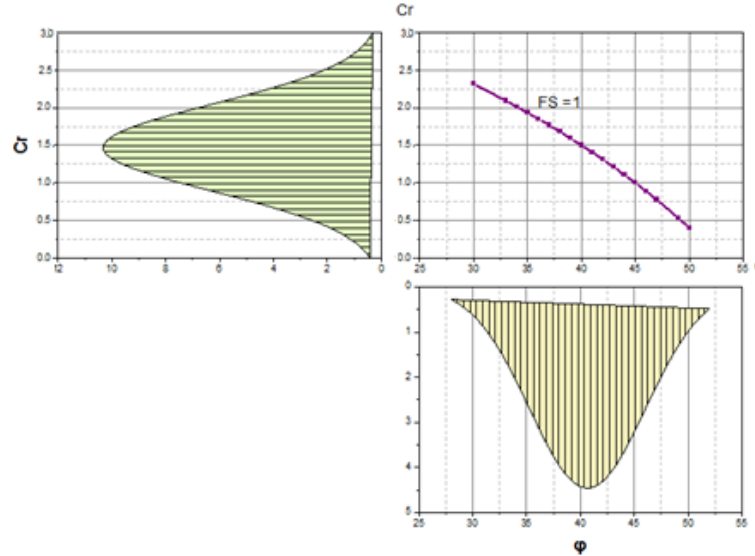


Figura 3-14. Funciones de densidad y FS= 1. Fuente: propia

5. Operando dichos valores como lo señala la ecuación (2-32) se obtiene el volumen resultante de probabilidad conjunta. En el siguiente fragmento del código se indica el proceso que se llevó a cabo en el cálculo de la probabilidad conjunta.

```
def cinetica_planar(datos: DataCtl, datos_out: DataCinetica):
    # Construir los ejes de la red phi vs cr
    minim = datos.datos_friccion['phi'].min()
    maxim = datos.datos_friccion['phi'].max()
    step = 0.01
    datos.cinetic_planar.theta = 89 if datos.cinetic_planar.theta > 89
else datos.cinetic_planar.theta
    theta_rad = np.deg2rad(datos.cinetic_planar.theta)
    axes = pd.DataFrame({'phi': np.arange(minim, maxim + step, step)})
    axes['cr'] = calculo_cr(axes['phi'], theta_rad,
    datos.cinetic_planar.h, datos.cinetic_planar.gamma)

    # Construir la red
    index = pd.MultiIndex.from_product([axes['phi'], axes['cr']])
    red_df = pd.DataFrame(index=index).reset_index()

    # Normalizar ejes de 0 a 1
    red_df['x_phi'] = (red_df['phi'] - minim) / (maxim - minim)
    minim = red_df['cr'].min()
    maxim = red_df['cr'].max()
    red_df['x_cr'] = (red_df['cr'] - minim) / (maxim - minim)

    # Calcular las distribuciones Beta
    red_df['pdf_phi'] = beta.pdf(red_df['x_phi'],
```

```

datos.cinetic_planar.alfa, datos.cinetic_planar.beta)
  red_df['pdf_cr'] = beta.pdf(red_df['x_cr'],
datos.cinetic_planar.alfa, datos.cinetic_planar.beta)

# Calcular probabilidad conjunta
red_df['prob'] = red_df['pdf_phi'] * red_df['pdf_cr']

```

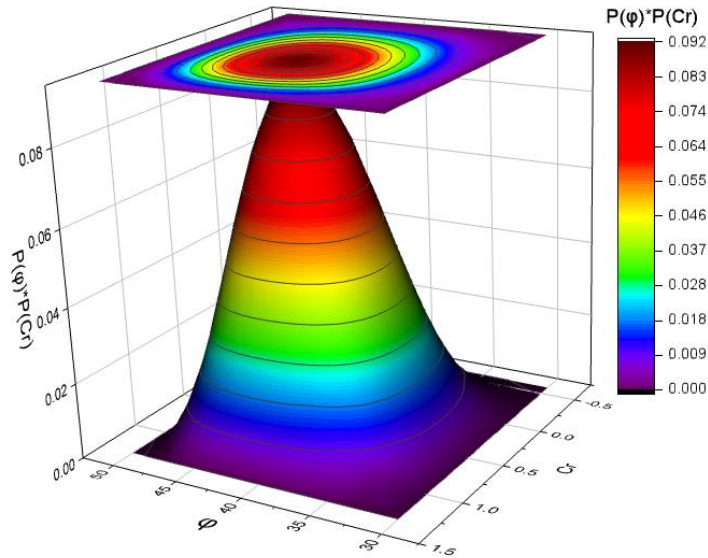


Figura 3-15. Volumen de probabilidad conjunta con funciones de densidad Beta. Fuente: propia

6. Para un Factor de seguridad igual a 1 se obtendrá la probabilidad de falla como el resultado del cálculo del volumen por debajo de la curva de Factor de seguridad multiplicado por la probabilidad de excedencia del sismo para un periodo de retorno de 475 años como una variable adicional en la evaluación de la probabilidad conjunta.

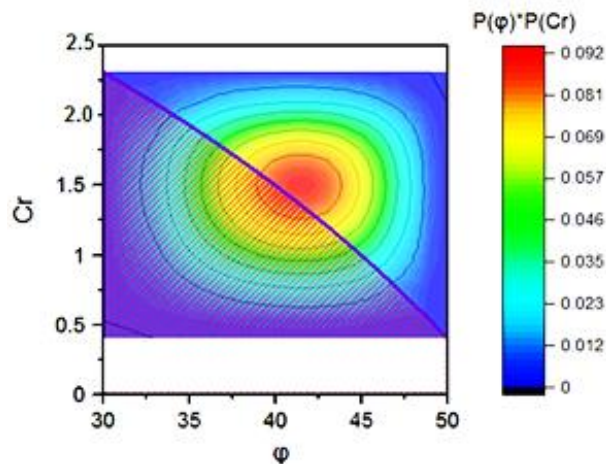


Figura 3-16. Probabilidad conjunta y límite de FS= 1, zona de falla. Fuente: propia

- **Procedimiento para falla por Volteo**

Para el desarrollo de la metodología se sugiere continuar con los siguientes pasos para la obtención de la probabilidad de falla por análisis cinético para volteo con enfoque probabilístico.

1. Al igual que en la metodología anterior, con los datos de los valores de ángulo de fricción se calculará la función de densidad de probabilidad, que para el caso de ángulo de fricción varios autores recomiendan la función Normal, normal Truncada o distribución Beta. En las siguientes graficas se indica la función de densidad y distribución de probabilidad de los datos que se han venido analizando.

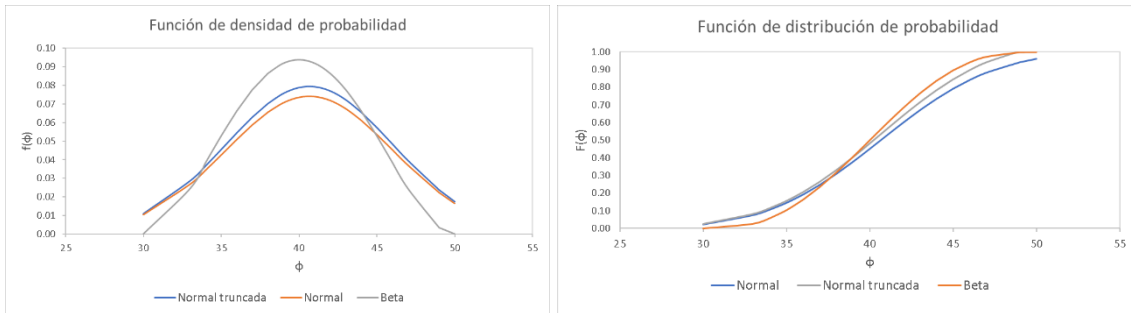


Figura 3-17. Función de densidad de probabilidad y función de distribución de probabilidad
Fuente: propia

2. Luego se calcula la función de densidad de probabilidad para la otra variable a estudiar que en este caso es la inclinación de la superficie de deslizamiento o volteo.

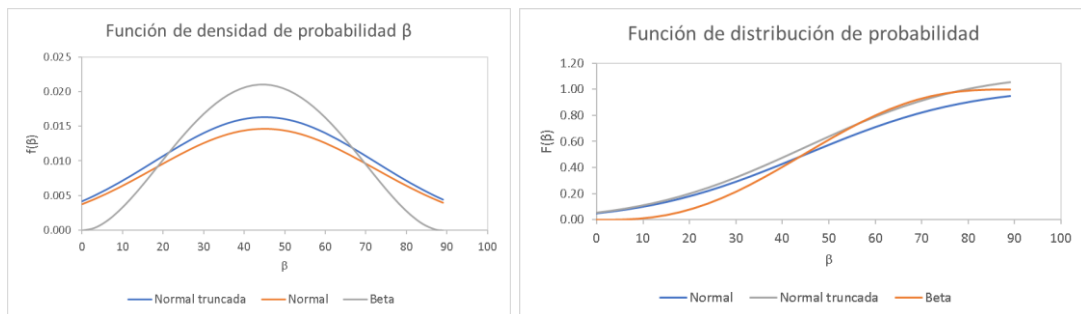


Figura 3-18. Función de densidad y distribución para superficie de deslizamiento o volteo.
Fuente: propia

3. Calcular la probabilidad conjunta de dos funciones para obtener la variable V de la ecuación (2-38) que será el producto de las probabilidades de cada una de las variables. Gráficamente se representaría como en la Figura 3-19 y en la Figura 3-20 se representa el volumen de probabilidades resultante.

```
def cinetica_volteo(datos: DataCtl, datos_out: DataCinetica):
    # Datos de inclinación
    step = 0.1
    beta_df = pd.DataFrame({'beta': np.arange(0, 90 + step, step)})

    # Datos fricción
    maxim = datos.datos_friccion['phi'].max()
    minim = datos.datos_friccion['phi'].min()
    df = pd.DataFrame({'phi': np.arange(minim, maxim + step, step)})

    # Normalización de variables
    df['x_p'] = (df['phi'] - minim) / (maxim - minim)
    beta_df['x_b'] = beta_df['beta'] / 90
    theta = datos.cinetic_volteo.theta / 90 if
datos.cinetic_volteo.theta <= 90 else 1

    # Creación de la red
    index = pd.MultiIndex.from_product([df['x_p'], beta_df['x_b']])
    red_df = pd.DataFrame(index=index).reset_index()

    # Calculo de distribuciones de probabilidad
    red_df['f_b'] = beta.pdf(red_df['x_b'], datos.cinetic_volteo.alfa,
datos.cinetic_volteo.beta)
    red_df['f_p'] = beta.pdf(red_df['x_p'], datos.cinetic_volteo.alfa,
datos.cinetic_volteo.beta)

    # Calculo de probabilidad conjunta
    red_df['prob'] = red_df['f_p'] * red_df['f_b']
    constant = red_df['prob'].sum()
    red_df['prob_n'] = red_df['prob'] / constant
```

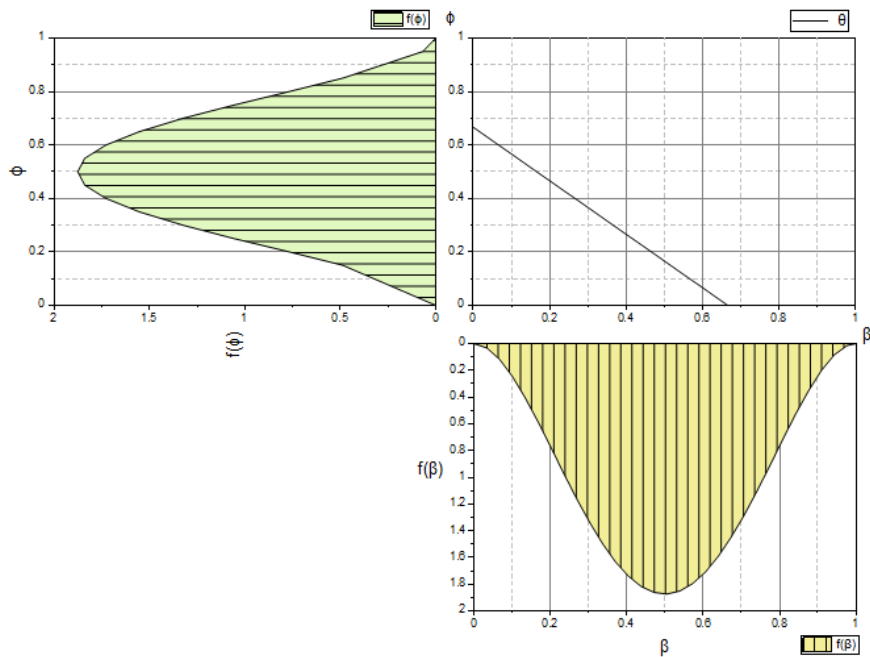



Figura 3-19. Funciones de densidad ángulo de fricción y de la inclinación de superficie de falla. Fuente: propia

```
def cinetica_volteo(datos: DataCtl, datos_out: DataCinetica):
    # Datos de inclinación
    step = 0.1
    beta_df = pd.DataFrame({'beta': np.arange(0, 90 + step, step)})

    # Datos fricción
    maxim = datos.datos_friccion['phi'].max()
    minim = datos.datos_friccion['phi'].min()
    df = pd.DataFrame({'phi': np.arange(minim, maxim + step, step)})

    # Normalización de variables
    df['x_p'] = (df['phi'] - minim) / (maxim - minim)
    beta_df['x_b'] = beta_df['beta'] / 90
    theta = datos.cinetic_volteo.theta / 90 if
datos.cinetic_volteo.theta <= 90 else 1

    # Creación de la red
    index = pd.MultiIndex.from_product([df['x_p'], beta_df['x_b']])
    red_df = pd.DataFrame(index=index).reset_index()

    # Calculo de distribuciones de probabilidad
    red_df['f_b'] = beta.pdf(red_df['x_b'], datos.cinetic_volteo.alfa,
datos.cinetic_volteo.beta)
    red_df['f_p'] = beta.pdf(red_df['x_p'], datos.cinetic_volteo.alfa,
datos.cinetic_volteo.beta)

    # Calculo de probabilidad conjunta
    red_df['prob'] = red_df['f_p'] * red_df['f_b']
    constant = red_df['prob'].sum()
    red_df['prob_n'] = red_df['prob'] / constant
```

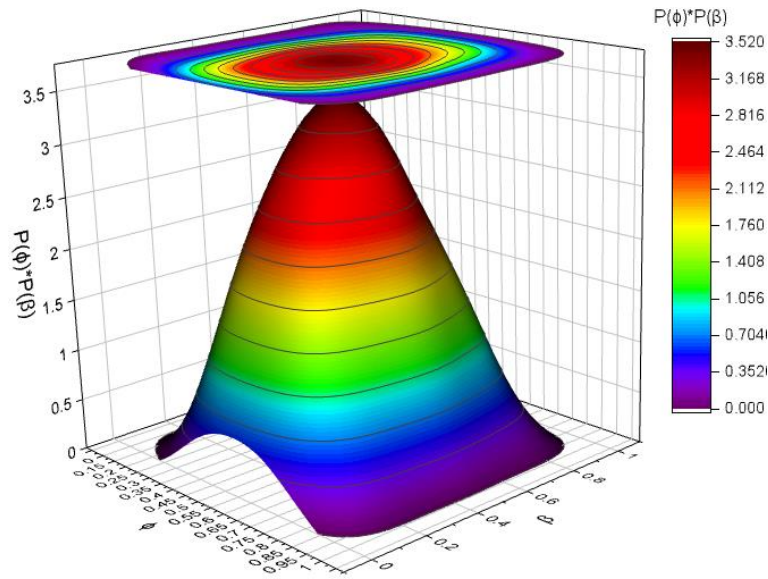


Figura 3-20. Volumen probabilidad conjunta. Fuente: propia

4. Luego para conseguir la probabilidad de falla se calculará el volumen de la parte del volumen por debajo de la envolvente que traza la inclinación del talud θ con respecto a la inclinación de la superficie de deslizamiento o volteo β y el ángulo de fricción ϕ .

En la Figura 3-21 se muestra un avista en planta del volumen de probabilidad conjunta y la envolvente de inclinación el talud, que para este caso es de $\theta= 60^\circ$.

```
# Calculo probabilidad conjunta dado beta
red_df['prob_f'] = (red_df['x_b'] <= theta - red_df['x_p']) *
red_df['prob_n']
volumen = red_df['prob_f'].sum()
prob_falla = du.prob_acc_sismica(datos.cinetic_volteo.t_retorno,
datos.cinetic_volteo.periodo) * volumen
datos_out.axes_df = df[['x_p']]
datos_out.axes2_df = beta_df[['x_b']]
datos_out.red_df = red_df[['x_p', 'x_b', 'prob_n', 'prob_f']]
datos_out.probabilidad = prob_falla
```

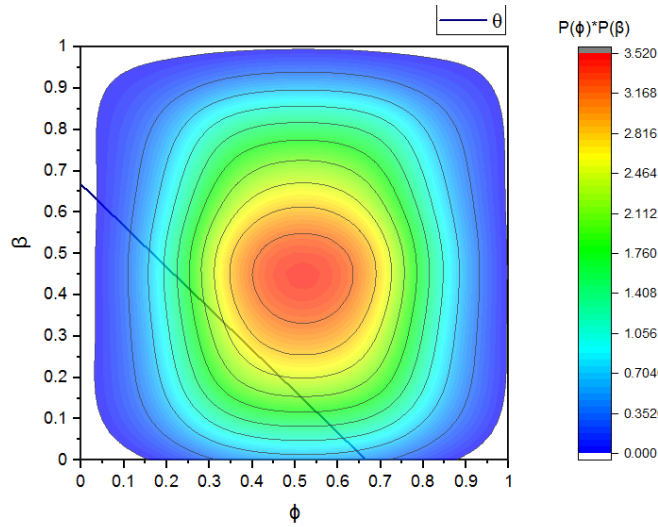


Figura 3-21. Probabilidad conjunta, inclinación de superficie de falla 60°. Fuente: propia

5. Continuando con el procedimiento mostrado en la Figura 2-10, se generará la función de probabilidad de la aceleración y la probabilidad de excedencia de la curva de amenaza para la probabilidad de falla, haciendo uso de los datos del modelo nacional de amenaza sísmica para Colombia, en donde se especifica para cada sitio o región específica de Colombia la curva de amenaza sísmica y el espectro de amenaza para diferentes periodos de retorno.

- **Procedimiento para falla en cuña**

1. Teniendo los datos de las variables a analizar en este caso, cohesión y ángulo de fricción, se procede a obtener las zonas de falla de las variables de cada uno de los planos que conforman la cuña, se tendrá en cuenta los valores máximos y mínimos de cada grupo de datos, y asumiendo un factor de seguridad igual a 1 se obtendrá la curva del límite de la zona con la ecuación (2-44) para la cohesión y la ecuación (2-45) para el ángulo de fricción.

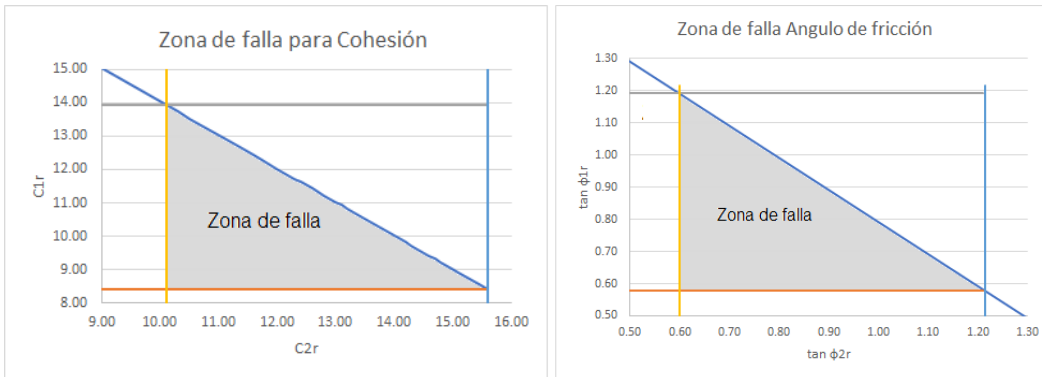


Figura 3-22. Zonas de falla para cohesión y ángulo de fricción de los dos planos. Fuente: propia

2. Luego la representación de las funciones de densidad de cada una de las variables que para este caso se trabajara con la función Beta con parámetros de la función $\alpha = \beta = 3$. Cada uno de los valores de la función de densidad de las variables se operan de tal manera que se obtenga el volumen de probabilidad de cohesión y de ángulo de fricción. En las siguientes figuras se representa las funciones de distribución y la zona de falla demarcada para un Factor de seguridad de 1.

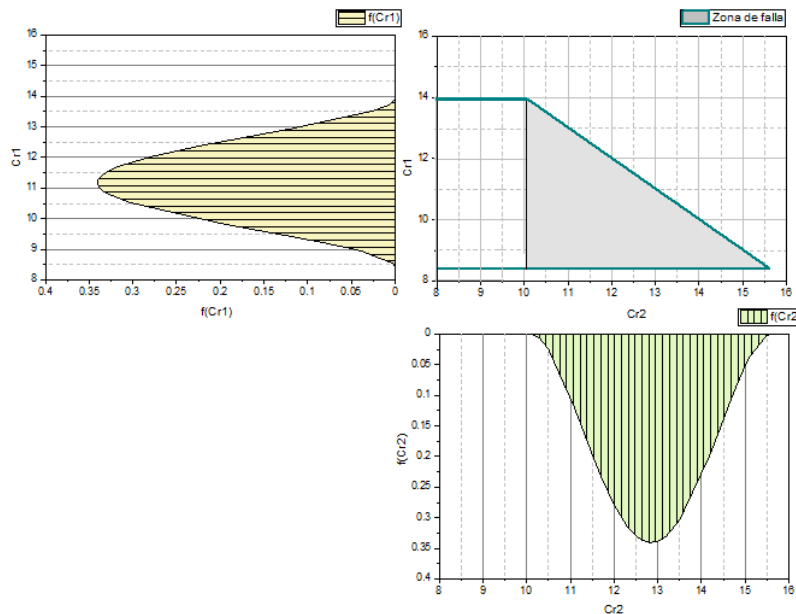


Figura 3-23. Funciones de densidad sobre zona de falla para cálculo de volumen de probabilidad para cohesión Fuente: propia

Y el resultado de la probabilidad final de la cohesión será el volumen de la sección de la zona de falla.

```

def cinetica_cunia_cohesion(datos: DataCtl, datos_out: DataCinetica):
    num_divisiones = 100

    # Cálculo de phi1 a partir de phi_2
    df = pd.DataFrame({'Cr2': datos.datos_cohesion['Cr2']})
    c_phi = datos.cinetic_cunnia.plano1.c *
datos.cinetic_cunnia.plano1.A + \
        datos.cinetic_cunnia.plano2.c * datos.cinetic_cunnia.plano2.A
    r_phi = datos.cinetic_cunnia.plano1.N *
np.tan(np.deg2rad(datos.cinetic_cunnia.plano1.phi)) \
        + datos.cinetic_cunnia.plano2.N *
np.tan(np.deg2rad(datos.cinetic_cunnia.plano2.phi))
    t_req = c_phi + r_phi
    df['Cr1'] = (t_req * datos.cinetic_cunnia.n_c * c_phi -
datos.cinetic_cunnia.plano2.A * df['Cr2']) / \
        datos.cinetic_cunnia.plano1.A

    # Calculo de los valores máximos y mínimos de Cr1
    mean1 = df['Cr1'].mean()
    std1 = df['Cr1'].std()
    minim1 = mean1 - std1 * np.sqrt(28 / 4)
    maxim1 = mean1 + std1 * np.sqrt(28 / 4)

    # Calculo de los valores máximos y mínimos de Cr2
    mean2 = datos.datos_cohesion['Cr2'].mean()
    std2 = datos.datos_cohesion['Cr2'].std()
    minim2 = mean2 - std2 * np.sqrt(28 / 4)
    maxim2 = mean2 + std2 * np.sqrt(28 / 4)

    # Calculo de red Cr1, Cr2
    step1 = (maxim1 - minim1) / num_divisiones
    cr1_range = np.arange(minim1, maxim1 + 0.9 * step1, step1)
    step2 = (maxim2 - minim2) / num_divisiones
    cr2_range = np.arange(minim2, maxim2 + 0.9 * step2, step2)

    index = pd.MultiIndex.from_product([cr2_range, cr1_range],
names=['Cr2', 'Cr1'])
    red_df = pd.DataFrame(index=index).reset_index()
    red_df['x2'] = (red_df['Cr2'] - minim2) / (maxim2 - minim2)
    red_df['x1'] = (red_df['Cr1'] - minim1) / (maxim1 - minim1)

    # Calculo de distribuciones
    red_df['beta1'] = beta.pdf(red_df['x1'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)
    red_df['beta2'] = beta.pdf(red_df['x2'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)

    # Matriz de Cohesión
    red_df['prob'] = red_df['beta1'] * red_df['beta2']
    constant = red_df['prob'].sum()
    red_df['prob'] = red_df['prob'] / constant
    red_df = red_df.round(6)

    # Selección de la probabilidad
    red_df['prob_f'] = (red_df['Cr1'] <= (-datos.cinetic_cunnia.plano2.A
* red_df['Cr2'] +

```

```

                                t_req *
datos.cinetic_cunnia.n_c * c_phi) / datos.cinetic_cunnia.plano1.A) \
    * red_df['prob']

# Guardar los datos de salida
datos_out.eje1, datos_out.eje2, datos_out.eje_z1, datos_out.eje_z2 =
'Cr2', 'Cr1', 'prob', 'prob_f'
datos_out.axes_df = pd.DataFrame({'Cr2': cr2_range})
datos_out.axes2_df = pd.DataFrame({'Cr1': cr1_range})
datos_out.red_df = red_df[['Cr2', 'Cr1', 'prob', 'prob_f']]
datos_out.probabilidad = red_df['prob_f'].sum()

```

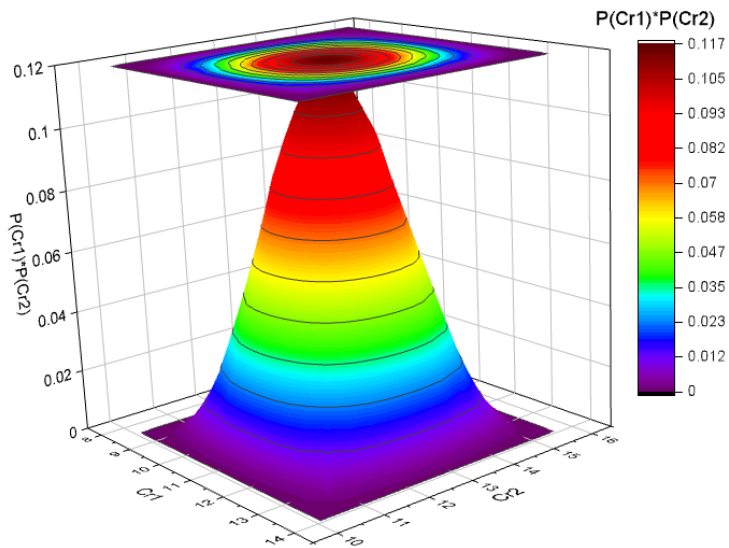


Figura 3-24. Volumen de probabilidad final para cohesión. Fuente: propia

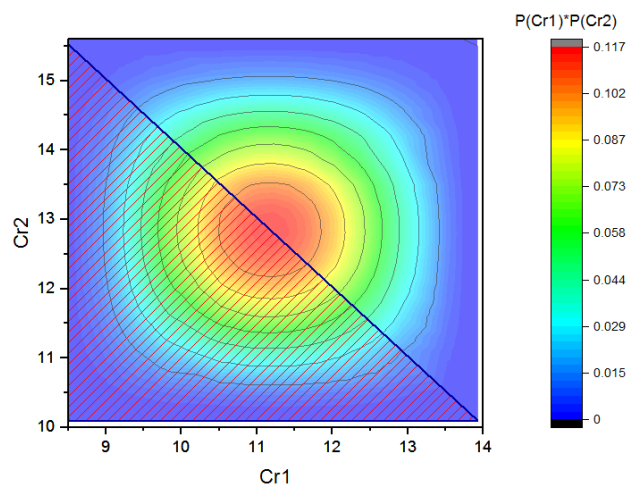


Figura 3-25. Visa en planta zona de falla sobre volumen de probabilidad final para cohesión. Fuente: propia

3. Y en el caso de ángulo de fricción las funciones de densidad serán las siguientes y de igual forma que con la cohesión se calculara un volumen de probabilidad final para el ángulo de fricción.

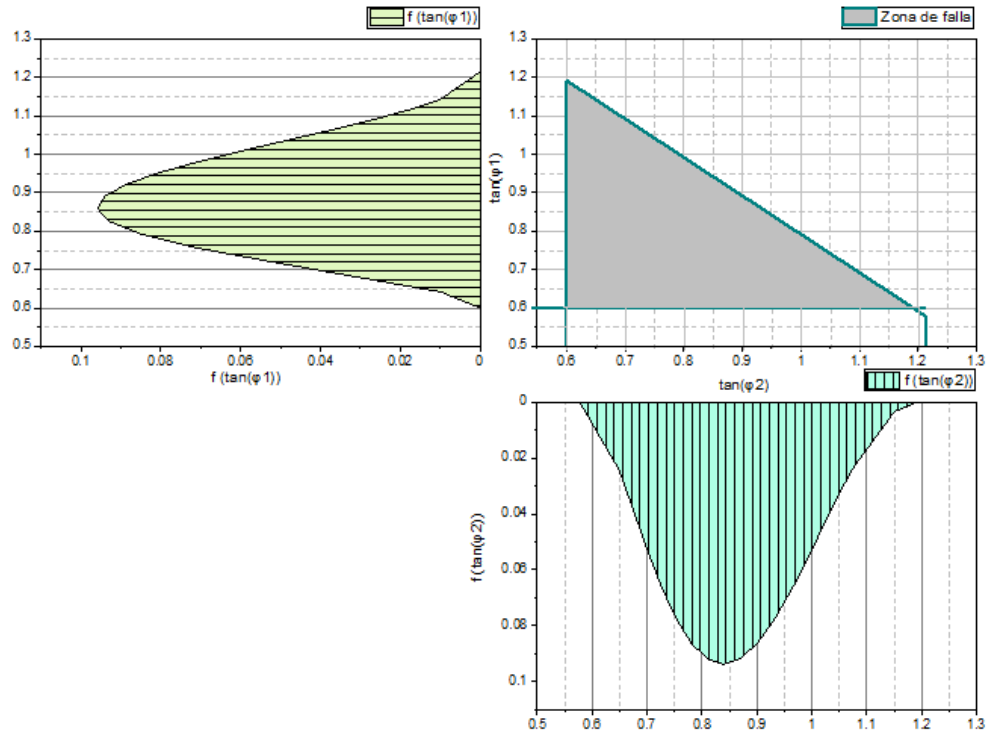


Figura 3-26. Funciones de densidad sobre zona de falla para cálculo de volumen de probabilidad para ángulo de fricción. Fuente: propia

```
def cinetica_cunia_friccion(datos: DataCtl, datos_out: DataCinetica):
    num_divisiones = 100

    # Cálculo de phi1 a partir de phi_2
    df = pd.DataFrame({'p2': datos.datos_friccion['phi']})
    df['tan_p2'] = np.tan(np.deg2rad(datos.datos_friccion['phi']))
    r_phi = datos.cinetic_cunnia.plano1.N *
    np.tan(np.deg2rad(datos.cinetic_cunnia.plano1.phi)) \
        + datos.cinetic_cunnia.plano2.N *
    np.tan(np.deg2rad(datos.cinetic_cunnia.plano2.phi))
    t_req = datos.cinetic_cunnia.plano1.c *
    datos.cinetic_cunnia.plano1.A + \
        datos.cinetic_cunnia.plano2.c *
    datos.cinetic_cunnia.plano2.A + r_phi
    df['tan_p1'] = - datos.cinetic_cunnia.plano2.N * df['tan_p2'] /
    datos.cinetic_cunnia.plano1.N + \
        t_req * datos.cinetic_cunnia.n_phi * r_phi /
    datos.cinetic_cunnia.plano1.N
    df['p1'] = np.rad2deg(np.arctan(df['tan_p1']))
```

```

# Calculo de los valores máximos y mínimos de phi_1
mean1 = df['p1'].mean()
std1 = df['p1'].std()
minim1 = mean1 - std1 * np.sqrt(28 / 4)
maxim1 = mean1 + std1 * np.sqrt(28 / 4)

# Calculo de los valores máximos y mínimos de phi2
mean2 = datos.datos_friccion['phi'].mean()
std2 = datos.datos_friccion['phi'].std()
minim2 = mean2 - std2 * np.sqrt(28 / 4)
maxim2 = mean2 + std2 * np.sqrt(28 / 4)

# Calculo de red p1, p2
step1 = (maxim1 - minim1) / num_divisiones
p1_range = np.arange(minim1, maxim1 + 0.9 * step1, step1)
step2 = (maxim2 - minim2) / num_divisiones
p2_range = np.arange(minim2, maxim2 + 0.9 * step2, step2)
tan_p1 = np.tan(np.deg2rad(p1_range))
tan_p2 = np.tan(np.deg2rad(p2_range))

index = pd.MultiIndex.from_product([p2_range, p1_range],
names=['p2', 'p1'])
red_df = pd.DataFrame(index=index).reset_index()
red_df['x2'] = (red_df['p2'] - minim2) / (maxim2 - minim2)
red_df['x1'] = (red_df['p1'] - minim1) / (maxim1 - minim1)

# Calculo de tangentes de la red
red_df['tan_p1'] = np.tan(np.deg2rad(red_df['p1']))
red_df['tan_p2'] = np.tan(np.deg2rad(red_df['p2']))

# Calculo de distribuciones
red_df['beta1'] = beta.pdf(red_df['x1'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)
red_df['beta2'] = beta.pdf(red_df['x2'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)

# Matriz de fricción
red_df['prob'] = red_df['beta1'] * red_df['beta2']
constant = red_df['prob'].sum()
red_df['prob'] = red_df['prob'] / constant
red_df = red_df.round(6)

# Selección de la probabilidad
red_df['prob_f'] = (red_df['tan_p1'] <= (-
datos.cinetic_cunnia.plano2.N * red_df['tan_p2'] +
t_req *
datos.cinetic_cunnia.n_phi * r_phi) /
datos.cinetic_cunnia.plano1.N) * red_df['prob']

datos_out.eje1, datos_out.eje2, datos_out.eje_z1, datos_out.eje_z2 =
'tan_p2', 'tan_p1', 'prob', 'prob_f'
datos_out.axes_df = pd.DataFrame({'tan_p2': tan_p2})
datos_out.axes2_df = pd.DataFrame({'tan_p1': tan_p1})
datos_out.red_df = red_df[['tan_p2', 'tan_p1', 'prob', 'prob_f']]
datos_out.probabilidad = red_df['prob_f'].sum()

```

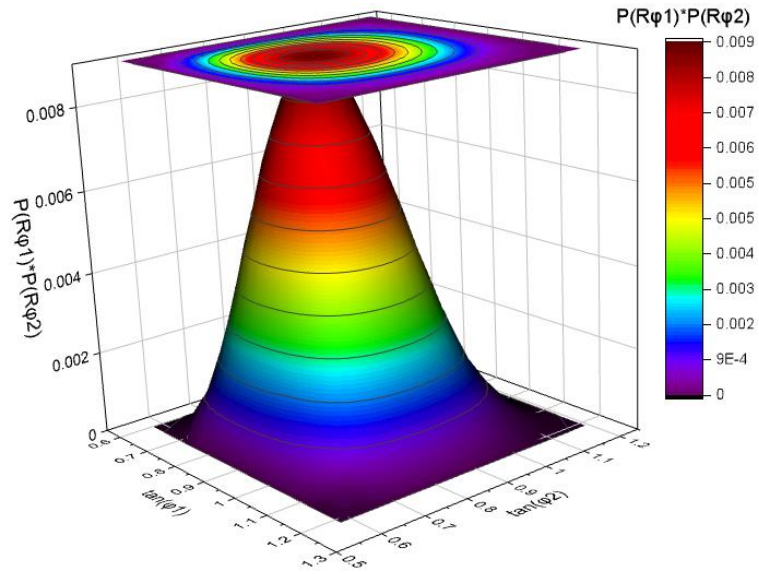



Figura 3-27. Volumen de probabilidad final para ángulo de fricción. Fuente: propia

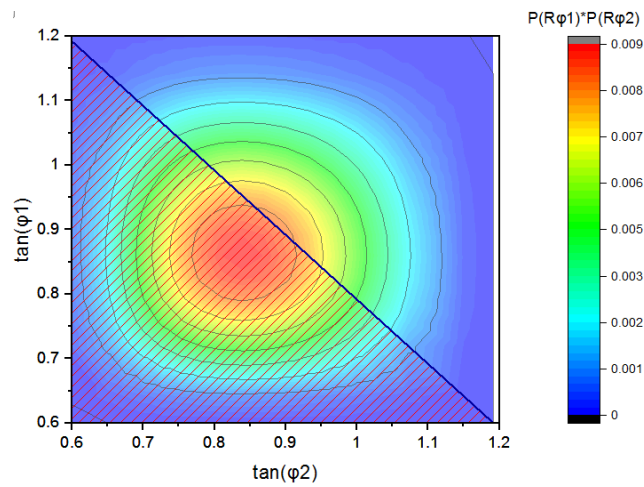


Figura 3-28. Vista en planta zona de falla sobre volumen de probabilidad final para ángulo de fricción. Fuente: propia

Finalmente, la probabilidad de falla tendrá en cuenta la probabilidad calculada de cada una de las variables y la probabilidad de excedencia del sismo como una variable adicional en la evaluación de la probabilidad conjunta.

```
def cinetica_cunia(datos: DataCtl, datos_out: DataOut):
    prob_friccion = cinetica_cunia_friccion(datos,
    datos_out.cinetica_cunia_friccion)
    prob_cohesion = cinetica_cunia_cohesion(datos,
    datos_out.cinetica_cunia_cohesion)
    probabilidad = prob_cohesion * prob_friccion *
    du.prob_acc_sismica(datos.cinetic_cunnia.t_retorno,
    datos.cinetic_cunnia.periodo)
```


4. Caso de estudio y aplicación de la metodología

Con el propósito de aplicar la metodología planteada, se selecciona un talud de una mina a cielo abierto de la cual se contará con información disponible de orientación y parámetros de resistencia de un caso de estudio.

4.1 Área de estudio

Para caso de estudio se tomará información de un macizo rocoso de la Mina El Pedregal la cual se localiza en el municipio de Une – Cundinamarca, a cargo de la Gravillera Albania S.A del Grupo Mincivil, de los taludes de la mina se realiza explotación para agregados para concreto. El macizo pertenece a la formación Areniscas de Une y se ubica en un monoclinal con buzamiento en dirección noroccidental e inclinación variable entre 18° y 30°. En el límite oriental del monoclinal se extiende un escarpe de una longitud de aproximadamente 1500 m, en cuyo extremo norte se ubica el frente de exploración de la mina. La Superficie expuesta con inclinación semi-vertical, con rumbo variable entre 280° y 300°, altura promedio de 25 metros y longitud aproximada 270 metros (Peñuela, 2019).



Figura 4-1. Esquema localización Mina el Pedregal. Fuente: Google Earth

Esta mina se encuentra en un macizo rocoso sedimentario, formado principalmente por intercalaciones de estratos de cuarzo-arenita de grano fino a grueso y lutitas carbonosas con variaciones de arena de grano mediano y moderada cementación; en la zona inferior intercalaciones de secuencias de areniscas de grano fino a muy fino y lutitas carbonosas de bajo espesor (Peñuela, 2019). La mina ha operado desde mediados de la década de 1990 hasta el presente. Holcim Colombia SA operó la mina hasta 2013, luego Gravillera Albania SA tomó el control de la operación (R. Hernández – Carrillo et al, 2019).

4.2 Análisis de datos

Los datos que se utilizarán en la aplicación de la metodología son tomados de la base de información recopilada como parte del desarrollo de la investigación “Adquisición y evaluación de datos geométricos de macizos rocosos a partir de imágenes tridimensionales para su uso en análisis geotécnicos” de Peñuela, J.L., Beltran-Calvo G., Hernandez-Carrillo, R. (2019).

4.2.1 Análisis estadístico de las discontinuidades

La base de datos disponible corresponde a mediciones de dos sectores de la mina, bloque norte y bloque sur, para los cuales se les realizara análisis estadístico para la definición de los valores medios que representen las familias de discontinuidades principales de cada bloque del macizo rocoso. Aplicando la metodología presentada en el numeral 2.2 para los dos bloques analizados los resultados se indican en la Tabla 4-1 y la Tabla 4-2.

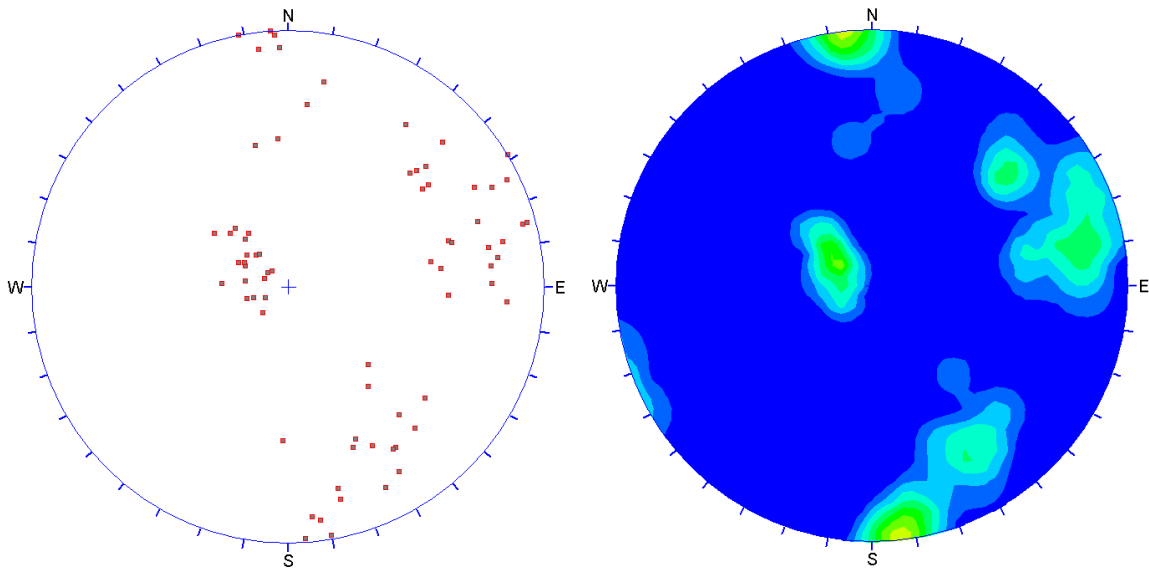


Figura 4-2. Datos estructurales Bloque Norte. Fuente: propia+ Dips

Tabla 4-1. Resumen análisis estadístico Bloque Norte

Familia	R	Planos		Polos		N	Kf
		Azimut (°)	Buzamiento (°)	Azimut (°)	Buzamiento (°)		
1	26.584	64	76	249	14	25	71.78
2	18.59	150	74	330	16	19	47.11
3	17.83	293	24	113	66	20	9.22
4	6.81	352	80	172	10	7	38.67

Fuente: propia

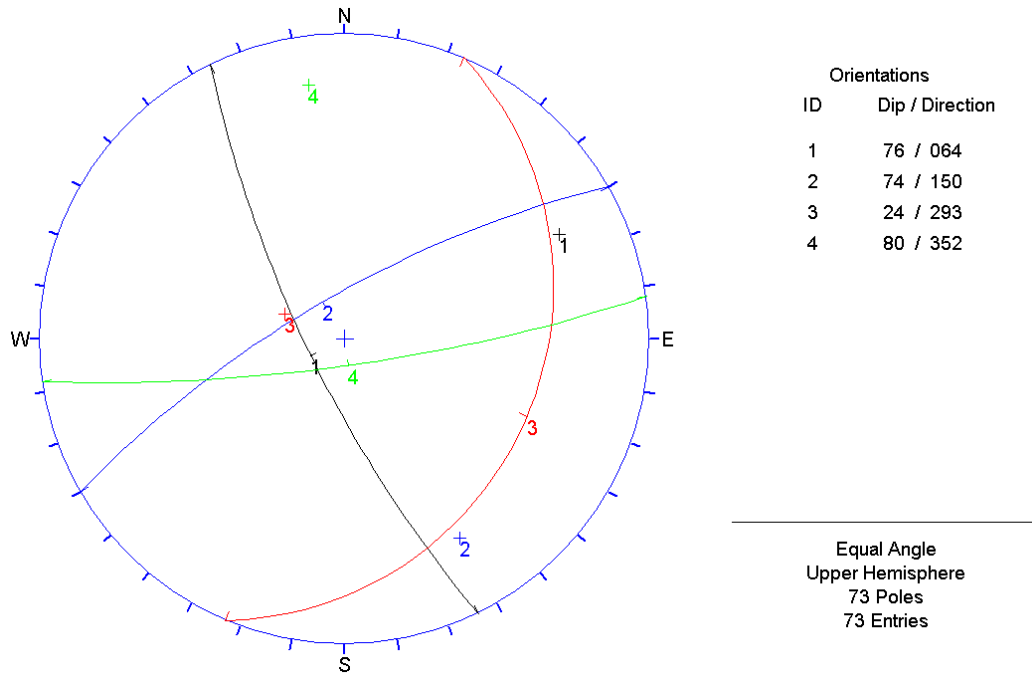


Figura 4-3 Familias de discontinuidades Bloque Norte. Fuente: propia

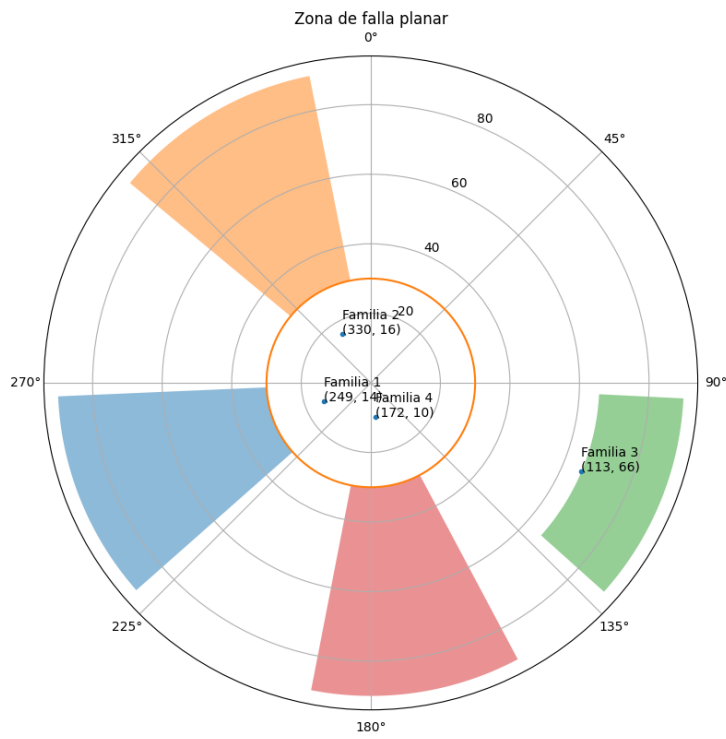


Figura 4-4. Zonificación falla planar Bloque Norte. Fuente: propia

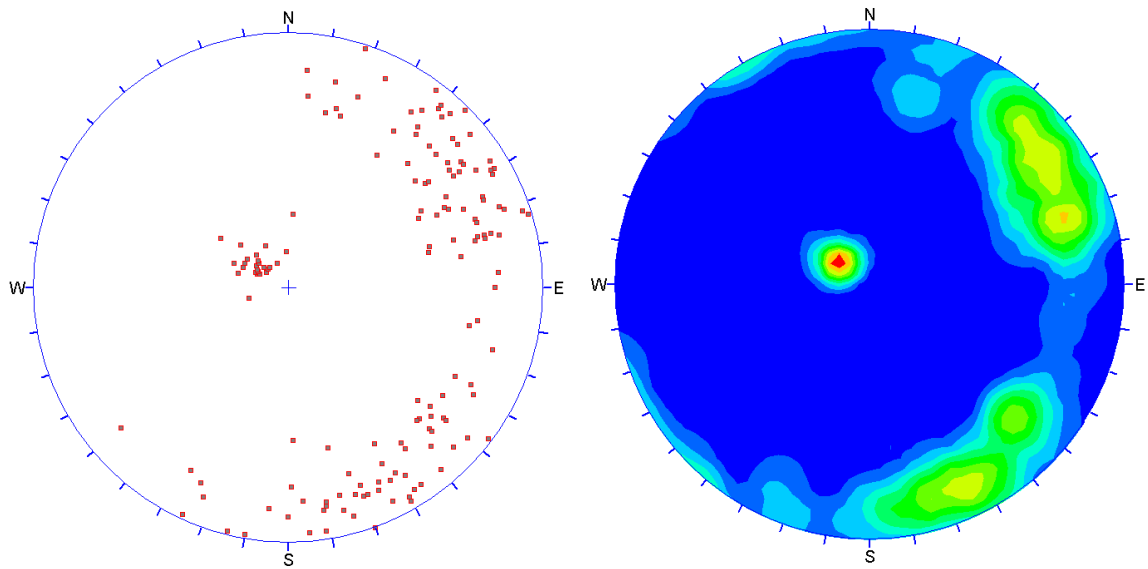


Figura 4-5. Datos estructurales Bloque Sur. Fuente: propia+ Dips.

Tabla 4-2. Resumen análisis estadístico Bloque Sur

Familia	R	Planos		Polos		N	Kf
		Azimut (°)	Buzamiento (°)	Azimut (°)	Buzamiento (°)		
1	74.14	57	78	234	12	75	87.36
2	63.49	151	81	328	9	64	125.61
3	24.39	308	20	124	71	26	16.16

Fuente: propia

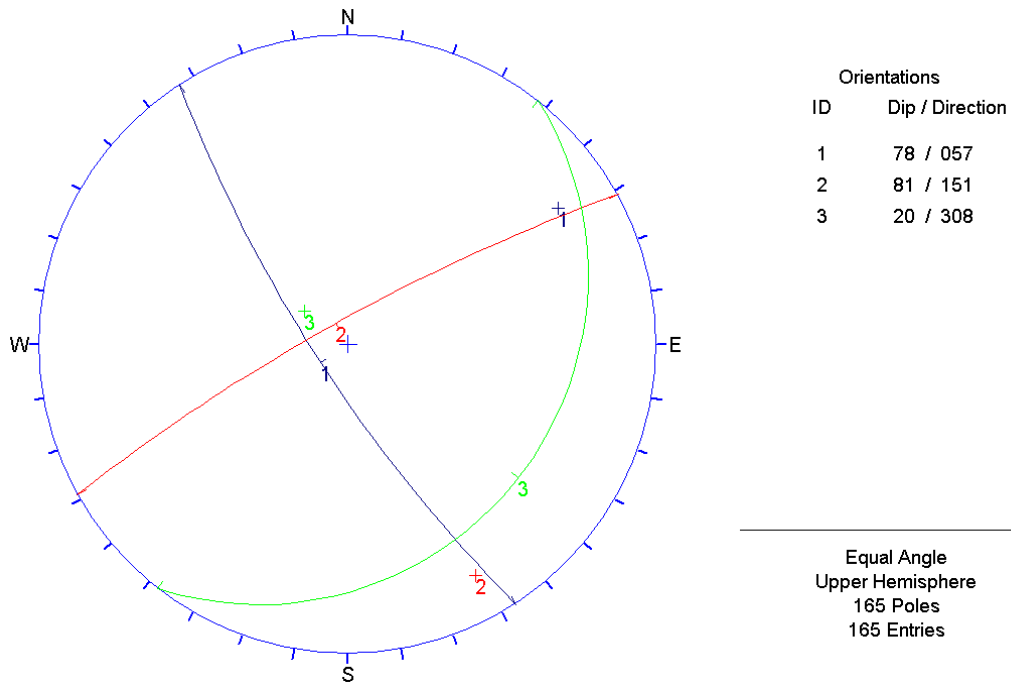


Figura 4-6 Familias de discontinuidades Bloque Sur. Fuente: propia

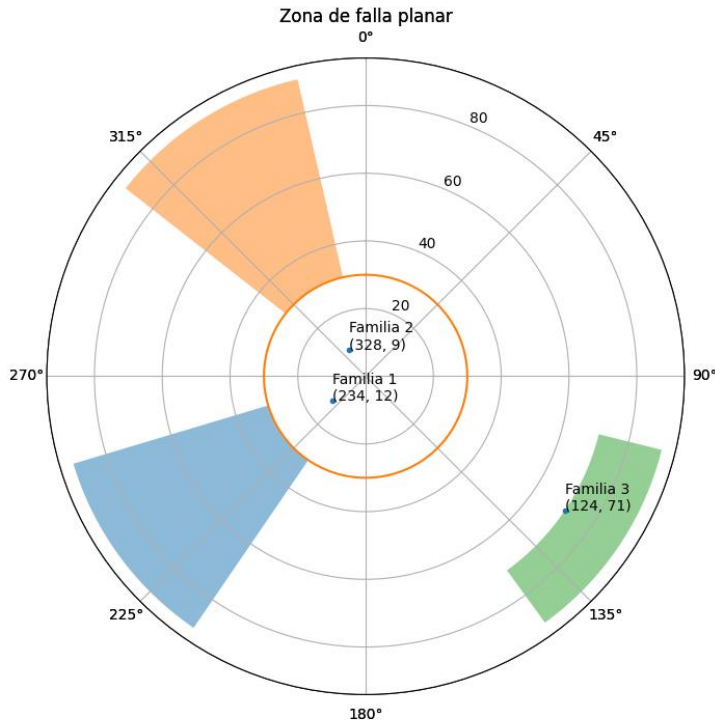


Figura 4-7. Zonificación falla planar Bloque Sur. Fuente: propia

4.2.2 Análisis estadístico del ángulo de fricción

Como información secundaria se tiene resultados de ensayos de corte directo que han sido ejecutados en diferentes campañas y se indican en la Tabla 4-3.

Tabla 4-3. Resumen de resultados de ensayos de corte por discontinuidades

Descripción	Peso Unitario g/cm ³	C Pico (Kg/cm ²)	Ø Pico (°)	Año
Lutita Carbonosa	2.462	0.48	31.8	2016
Lutita Carbonosa	2.531	0.69	22.0	2016
Lutita Carbonosa	2.531	0.97	18.5	2016
Lutita Carbonosa	2.594	1.59	34.4	2016
Lutita Carbonosa	2.604	0.90	20.4	2017
Lutita Carbonosa	2.594	0.68	36.7	2017
Lutita Carbonosa	2.591	1.49	37.4	2017
Limolita gris oscura	2.601	1.87	42.1	2017

Fuente: Peñuela, J.L., Beltran-Calvo G., Hernandez-Carrillo, R. (2019)

El análisis del ángulo de fricción se realiza según lo indicado en el numeral 2.4, en el cual se propone el análisis de la variable con una distribución normal truncada la cual representa la tendencia y permite limitar por valores máximos y mínimos de los datos recolectados por laboratorio o referencias bibliográficas, permitiendo de igual manera llevar la distribución de probabilidad a una representación gráfica en la red polar.

En la Figura 4-8 se representa la distribución normal truncada y la probabilidad acumulada de los datos de ángulo de fricción para el caso de estudio.

Tabla 4-4. Datos y cálculo de función de densidad y distribución del ángulo de fricción

Distribución normal		Media		30.41	
		Desv. estándar		8.910	
Min ϕ	18.5	Dist. Normal acumulada		0.091	
Max ϕ	42.1	Dist. Normal acumulada		0.905	
N°	ϕ	Normal		Normal truncada	
		Función de densidad	Función de distribución	Función de densidad	Función de distribución
1	31.8	0.044	0.562	0.054	0.521
2	22	0.029	0.173	0.035	0.091
3	18.5	0.018	0.091	0.022	0.000
4	34.4	0.041	0.673	0.050	0.643
5	20.4	0.024	0.131	0.029	0.044
6	36.7	0.035	0.760	0.043	0.739
7	37.4	0.033	0.784	0.040	0.766
8	42.1	0.019	0.905	0.023	0.900

Fuente: propia

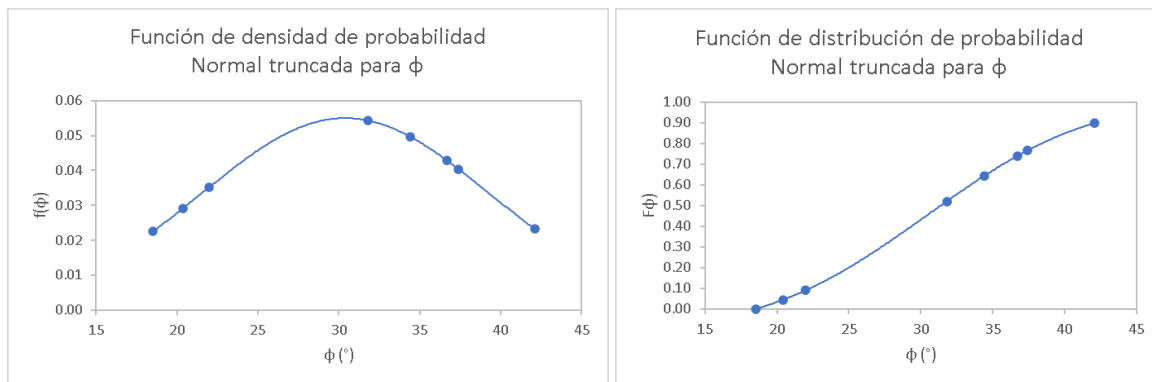


Figura 4-8. Distribución normal truncada para el ángulo de fricción Fuente: propia

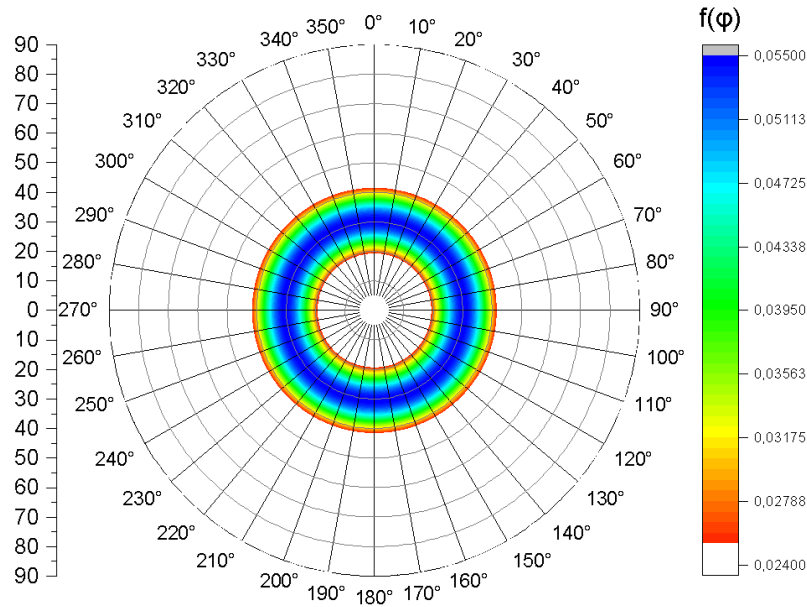


Figura 4-9. Probabilidad de ángulo de fricción. Fuente: propia

La Figura 4-9 indica un valor de probabilidad máximo entre los ángulos 18° y 42° aproximadamente, con un valor medio en 30°, permitiendo así identificar zonas de mayor y menor probabilidad de los datos de ángulo de fricción con los que se analizan en este trabajo. Usualmente, el ángulo de fricción se representa como un círculo de un único valor en la red polar, con esta representación de probabilidad del ángulo de fricción se alcanza a abarcar una zona amplia de la influencia de los valores del ángulo de fricción.

4.2.3 Probabilidad conjunta en red polar - Bloque Norte

Procediendo con la metodología propuesta, para cada uno de los bloques a analizar se define una orientación de talud crítica para el cual se le realiza el cálculo de la probabilidad conjunta entre la probabilidad de Fisher calculada para cada familia de discontinuidades obtenidas y la probabilidad del ángulo de fricción calculado y graficado en la red polar.

- Falla planar - Bloque Norte

Se define para el bloque Norte el análisis para un talud con orientación de 215°, y empleando la herramienta desarrollada, a continuación, desde la Figura 4-10 a la Figura 4-15 se presentan los resultados arrojados para el bloque de estudio.

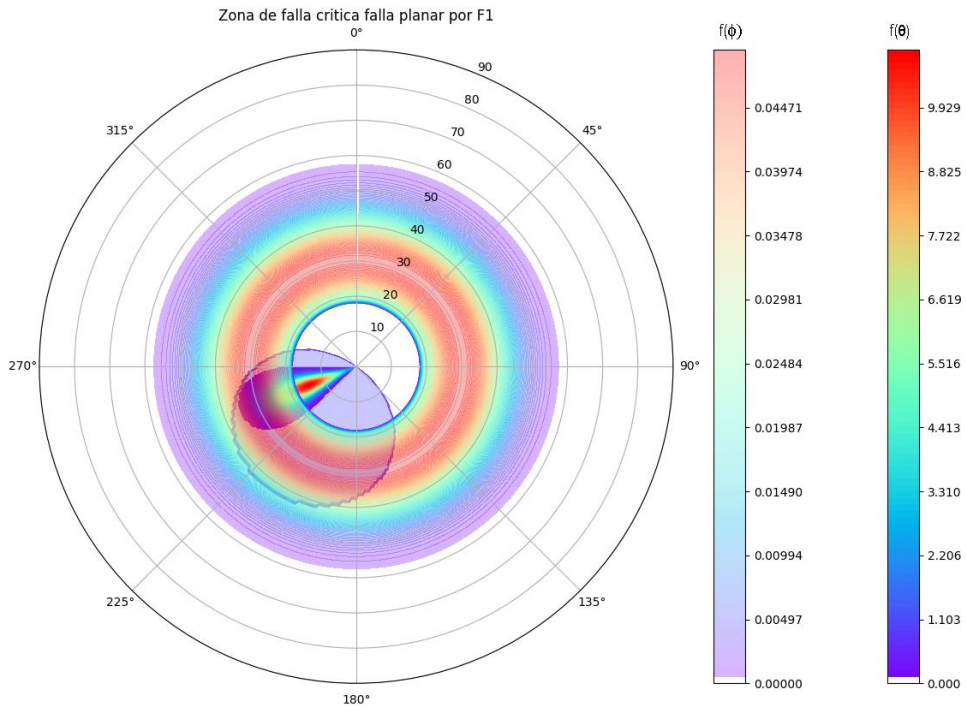


Figura 4-10. Contornos de probabilidades falla planar para F1

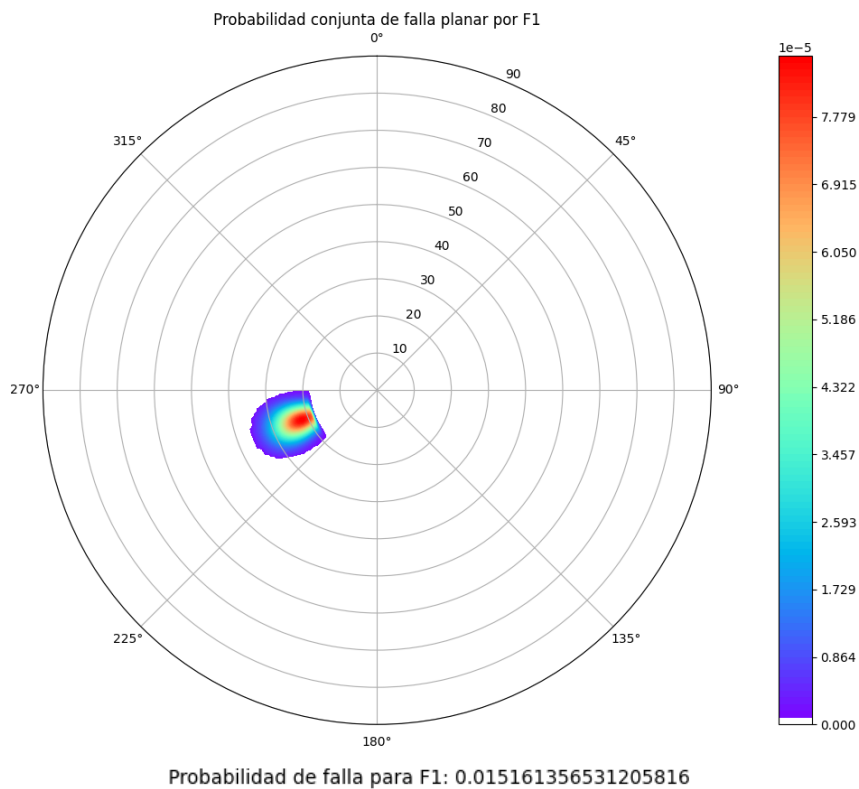


Figura 4-11. Zona de falla critica por falla planar para F1

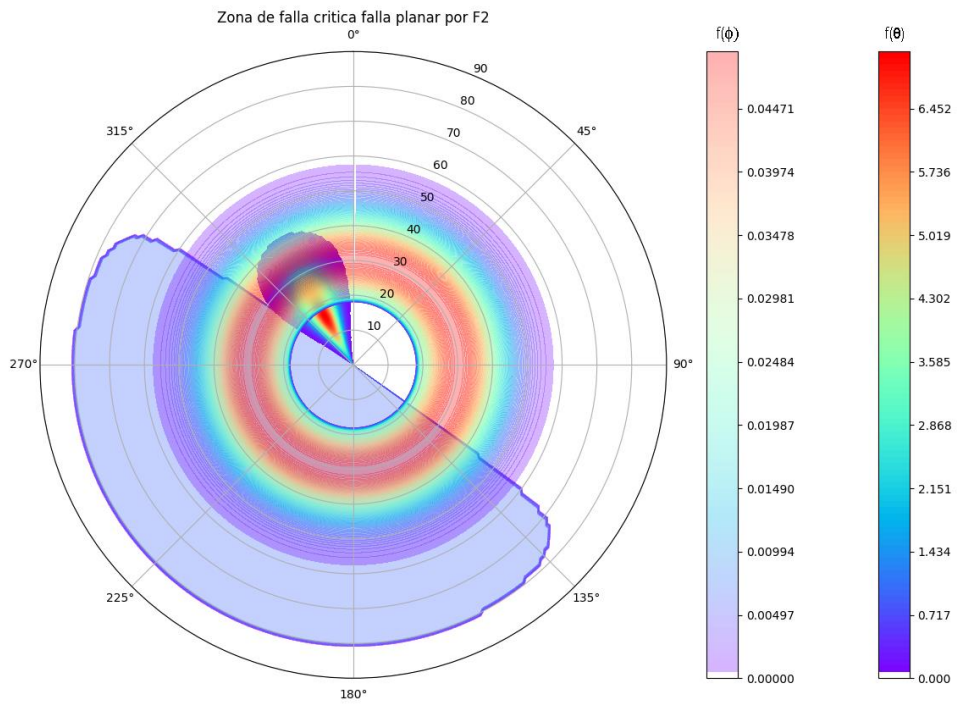


Figura 4-12. Contornos de probabilidades falla planar para F2

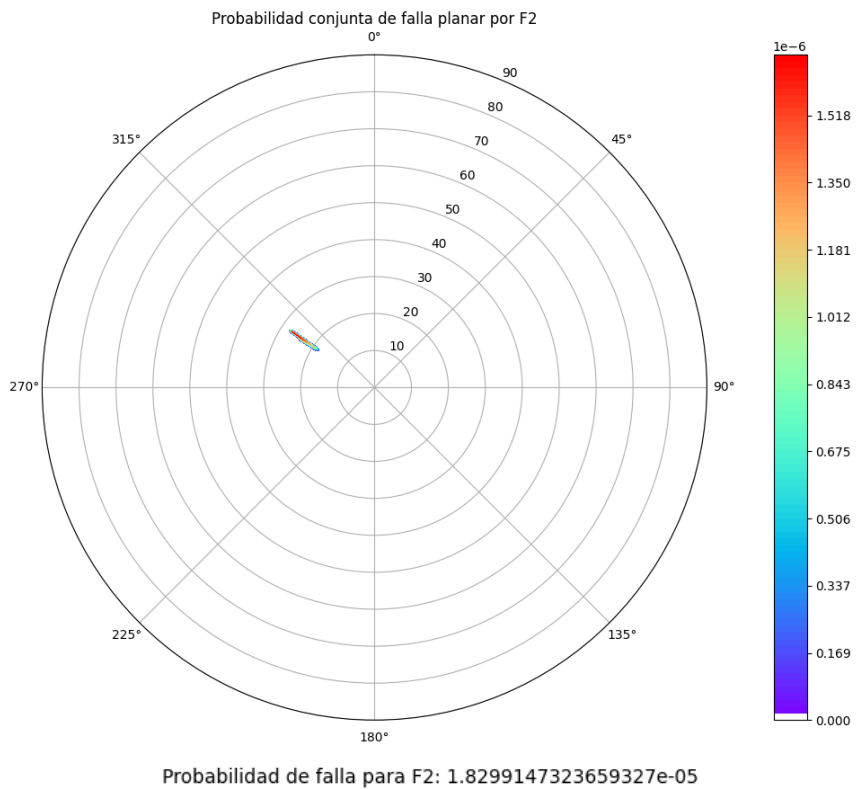


Figura 4-13. Zona de falla critica por falla planar para F2

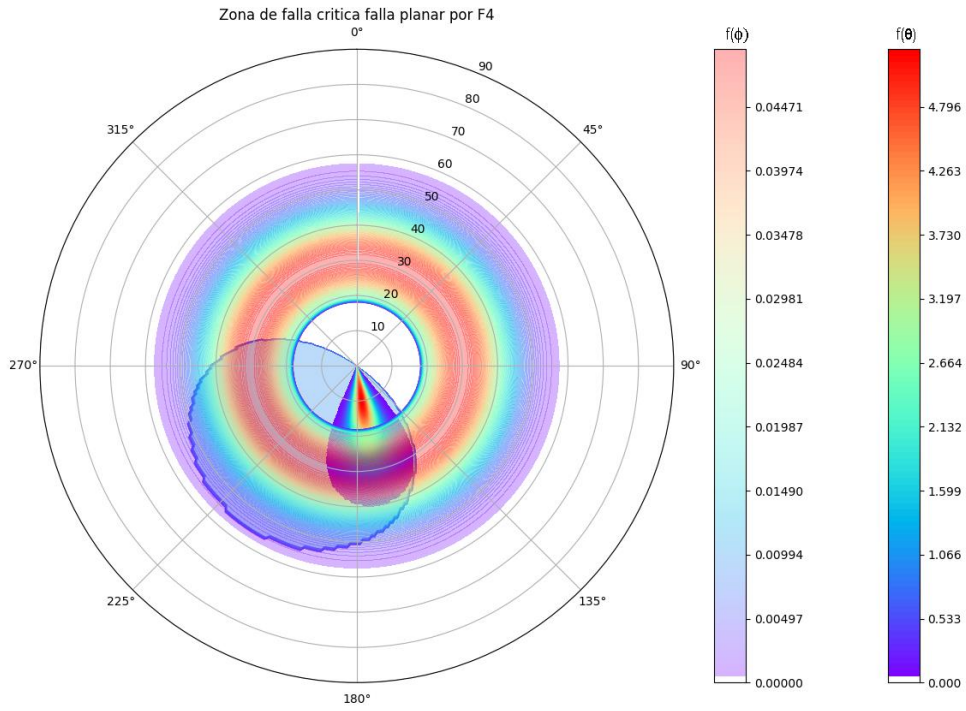
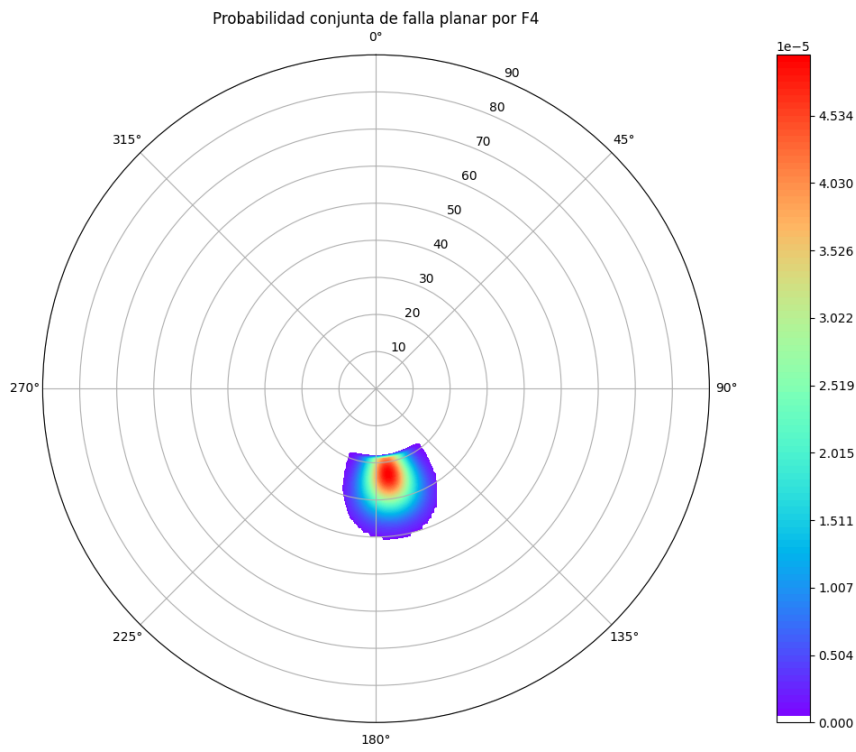


Figura 4-14. Contornos de probabilidades falla planar para F4



Probabilidad de falla para F4: 0.016140659437980334

Figura 4-15. Zona de falla critica por falla planar para F4

De las cuatro familias de discontinuidades se obtienen los contornos de probabilidad conjunta, y habiendo definido la orientación de talud crítico para el bloque se obtiene la zona de falla crítica por falla planar donde se evidencia que la Familia 1 y 4 están influenciadas por las envolventes del talud. Con este procedimiento se obtienen probabilidades de falla de 0.015 y 0.016 para las familias 1 y 4 respectivamente.

- Falla por volteo – Bloque Norte

Para la misma orientación de talud con orientación de 215° se realiza el cálculo de la probabilidad de falla por volteo, en la Figura 4-16 y hasta la Figura 4-19 se muestran los contornos y zonas de falla para las familias que alcancen a ser influenciadas por las envolventes de la orientación del talud definido.

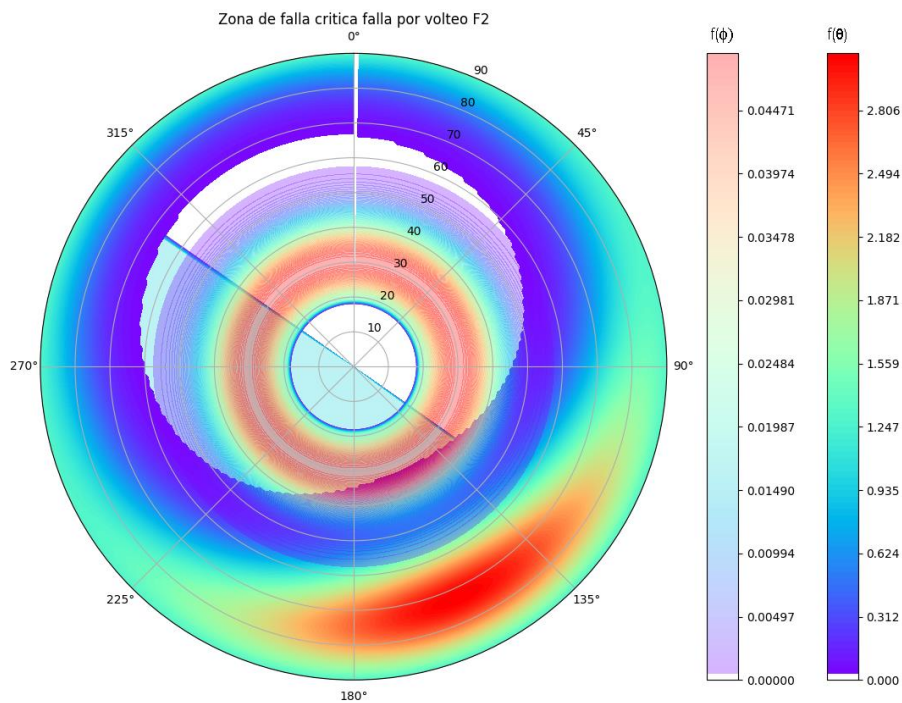


Figura 4-16. Contornos de probabilidades, falla por volteo para F2

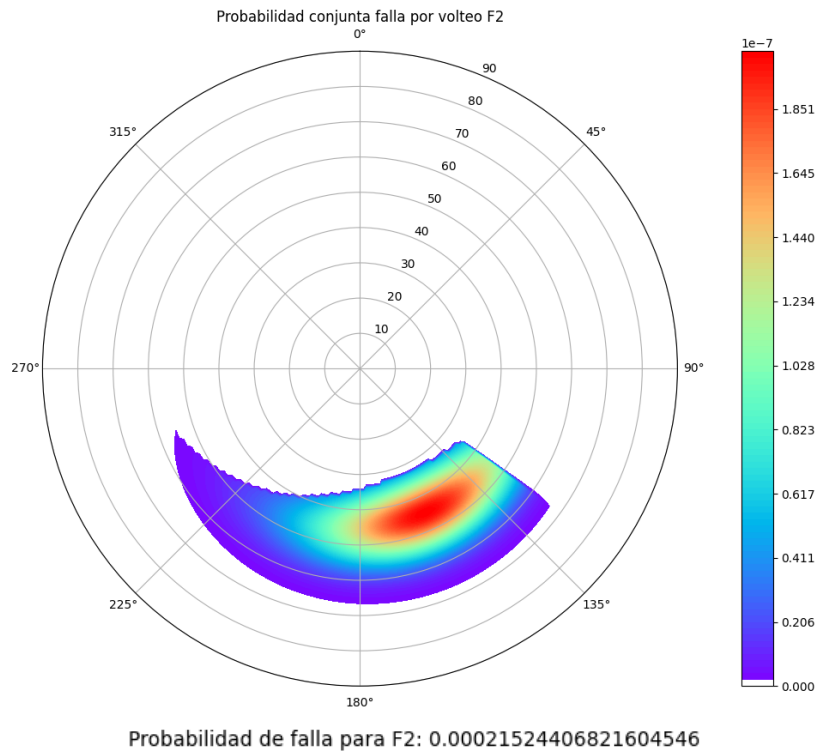


Figura 4-17. Zona de falla crítica por falla por volteo para F2

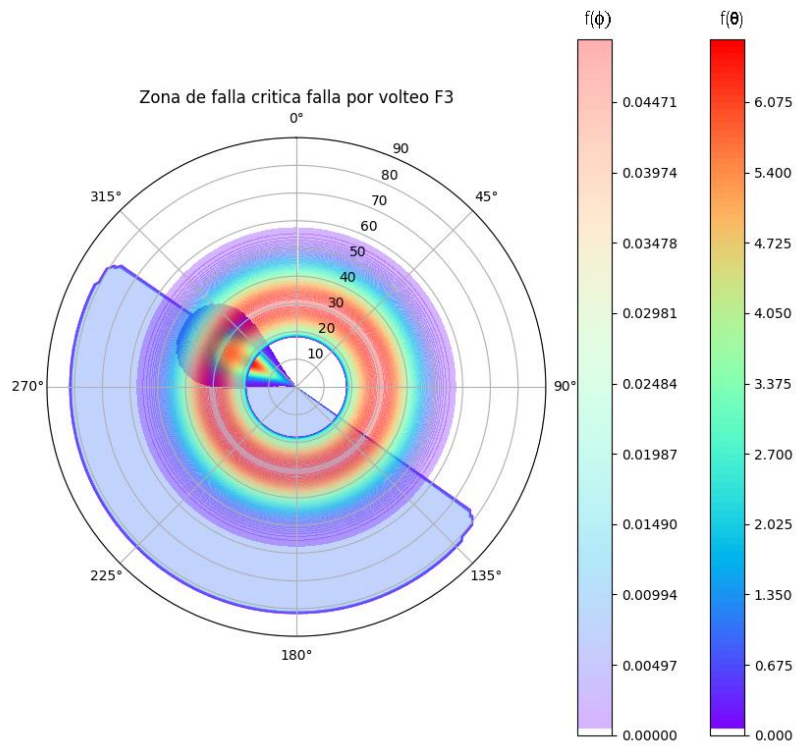


Figura 4-18. Contornos de probabilidades, falla por volteo para F3

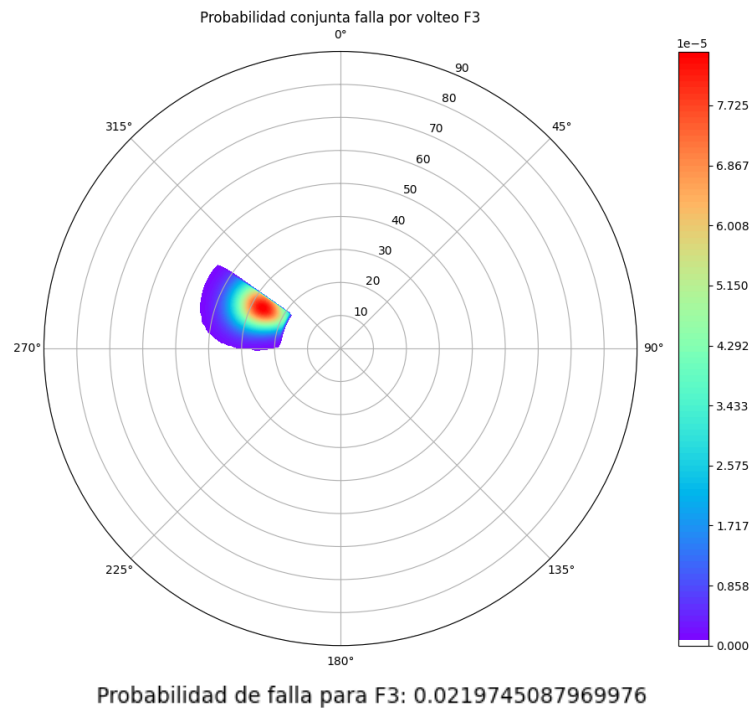


Figura 4-19. Zona de falla crítica por falla por volteo para F3

Con el análisis, se encuentra que las familias que son afectadas son las familias 2 y 3 con probabilidades de 0.000215 y 0.0219 respectivamente.

- Falla por cuña - Bloque Norte

Para las intersecciones que se generen de las familias de discontinuidades se realiza el cálculo de la probabilidad conjunta para la orientación crítica de talud, para el caso se presenta los resultados obtenidos para la intersección entre las familias 2 y 3 en la Figura 4-20 y en la Figura 4-21.

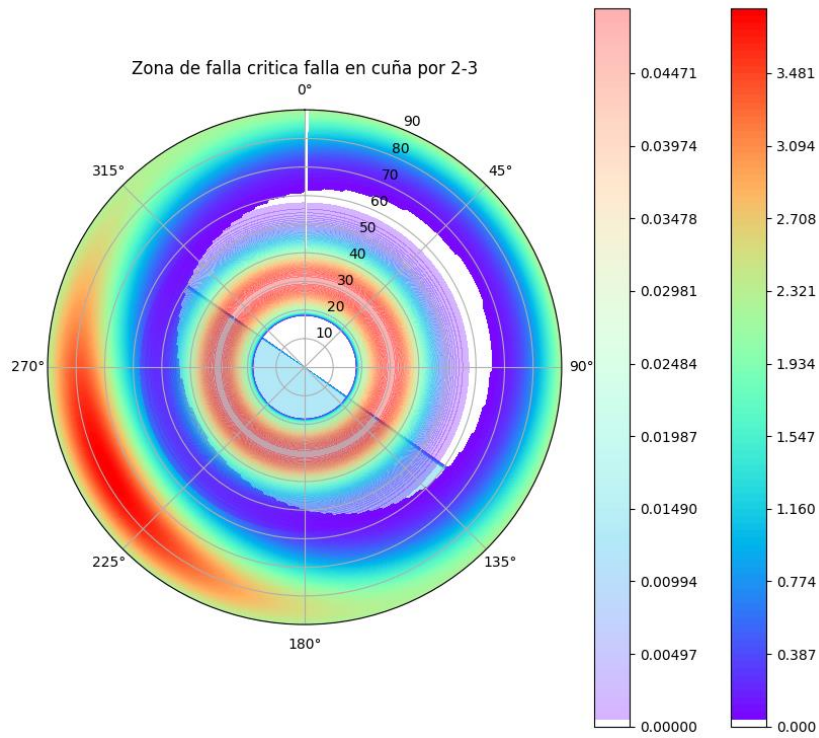
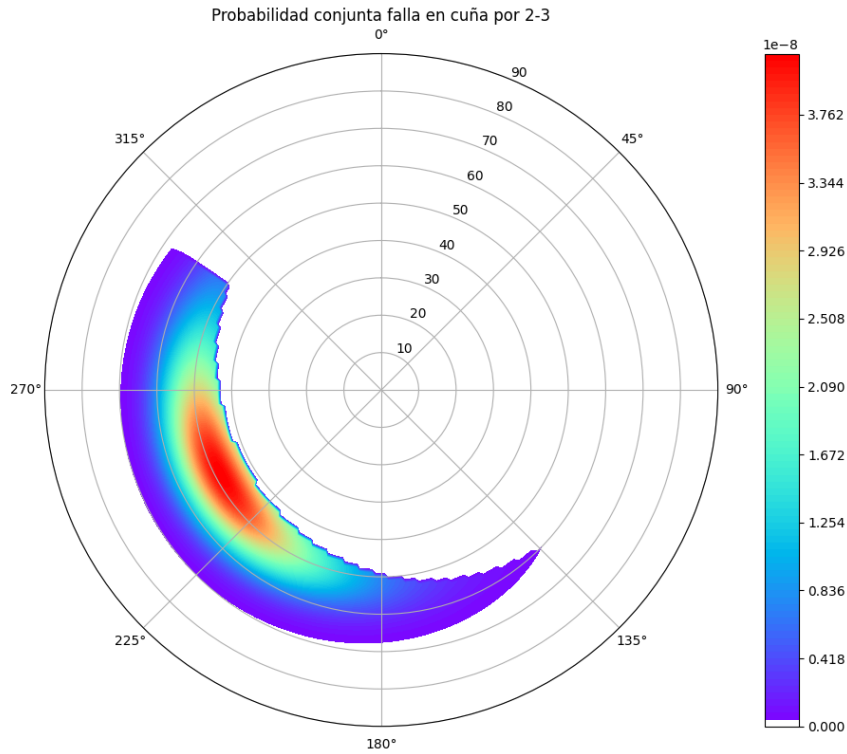


Figura 4-20. Zona de falla crítica, falla en cuña por F2-F3



Probabilidad de falla para F2-3: 4.4574643705864955e-05

Figura 4-21. Probabilidad conjunta, falla en cuña por F2-F3

4.2.4 Análisis por confiabilidad Bloque Norte

Luego con la metodología planteada en el numeral 3.2, se realiza el análisis por confiabilidad inicialmente para el bloque norte y luego de la misma forma para el bloque sur. Para el cálculo se tendrán los datos de resistencia obtenidos en el análisis cinemático.

- Falla planar - Bloque Norte

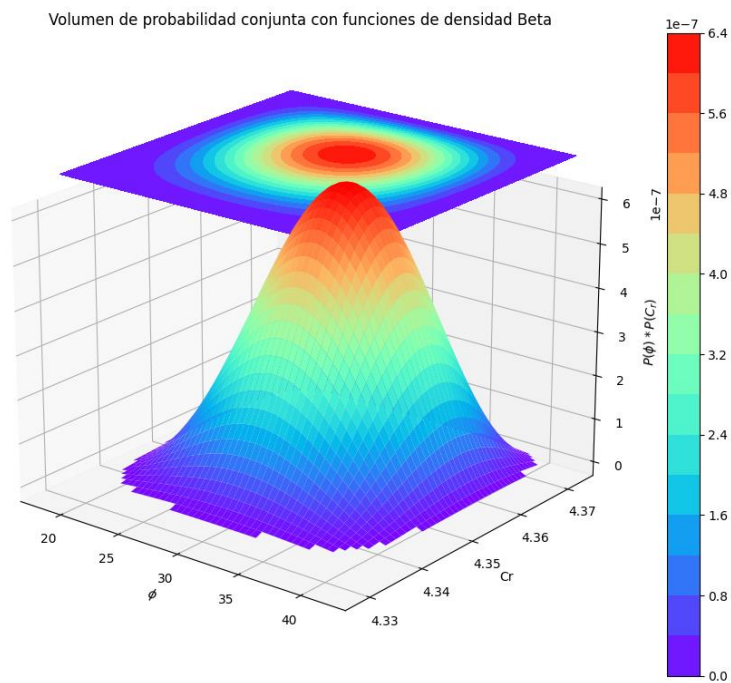


Figura 4-22. Volumen de probabilidad conjunta con funciones, falla planar

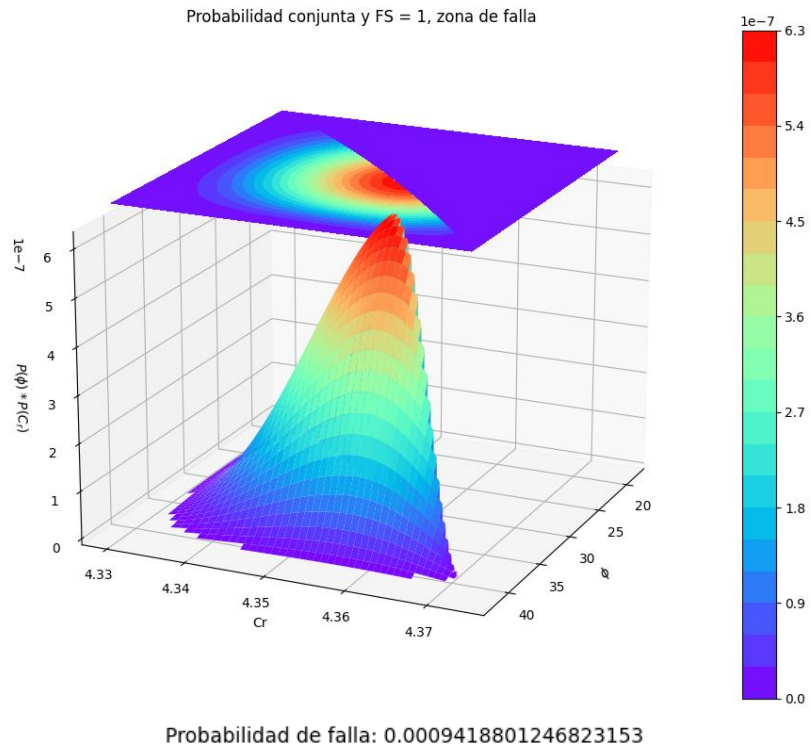


Figura 4-23. Probabilidad conjunta y FS= 1, zona de falla

- Falla por volteo - Bloque Norte

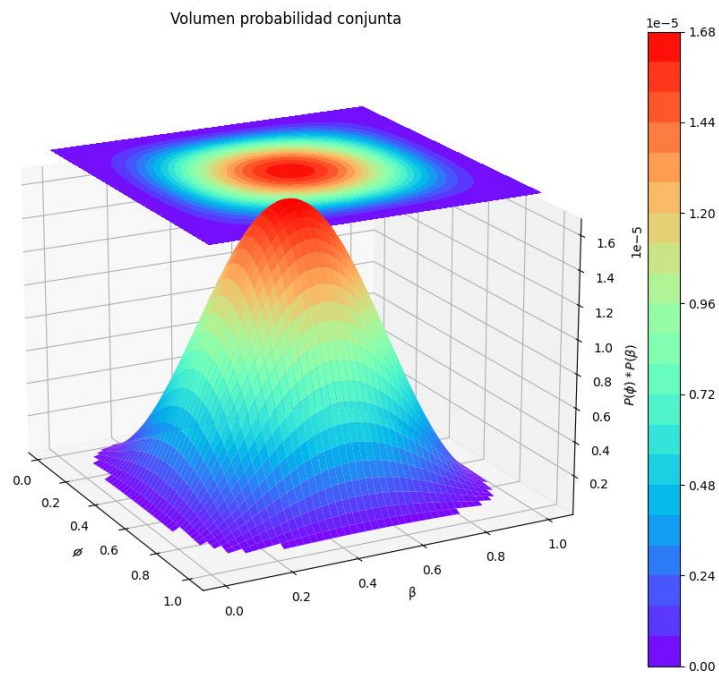


Figura 4-24. Volumen probabilidad conjunta, falla por volteo

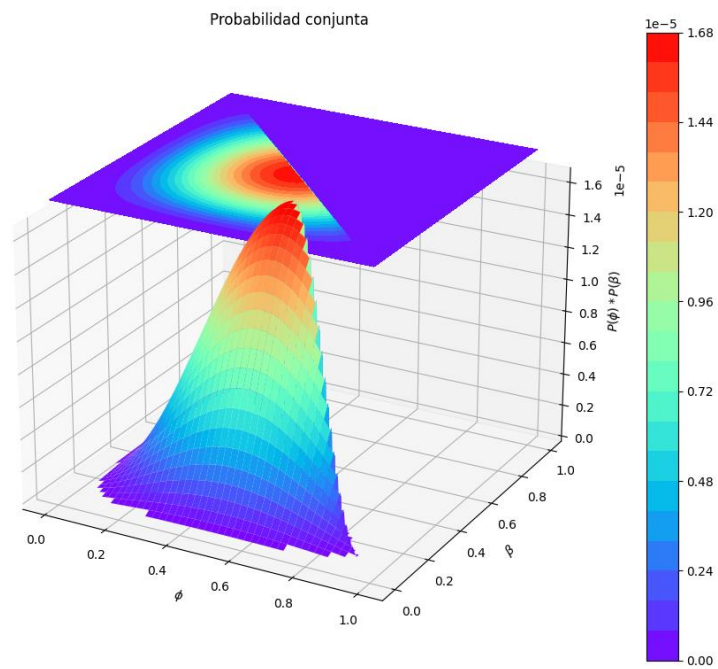


Figura 4-25. Probabilidad conjunta falla por volteo

- Falla por cuña - Bloque Norte

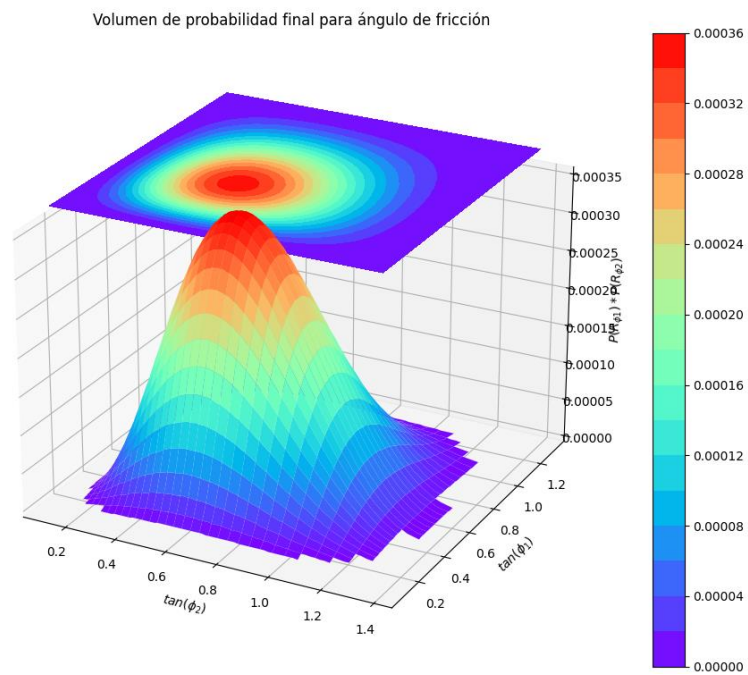


Figura 4-26. Volumen de probabilidad final para ángulo de fricción

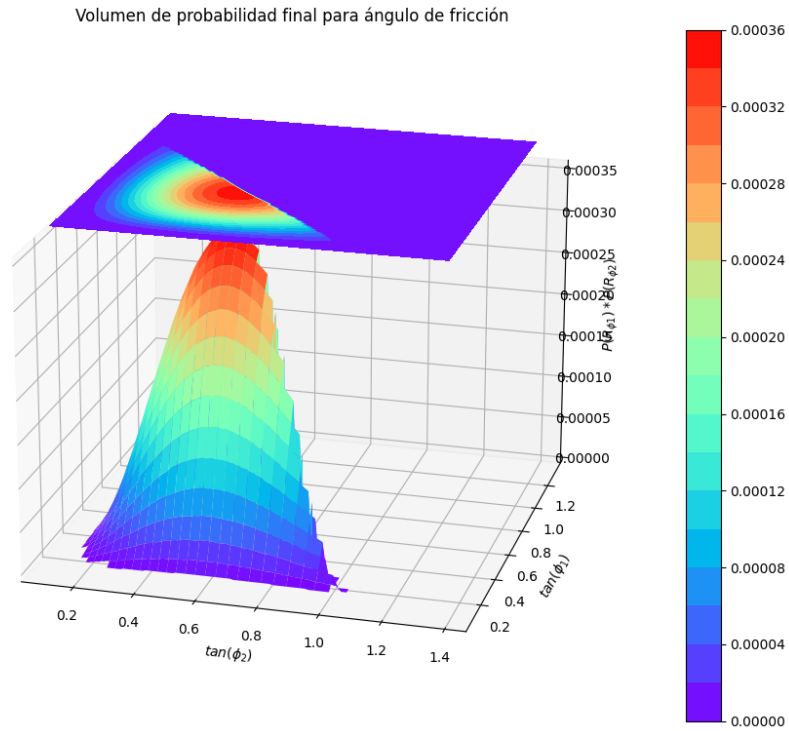


Figura 4-27. Zona de falla sobre volumen de probabilidad final para ángulo de fricción.

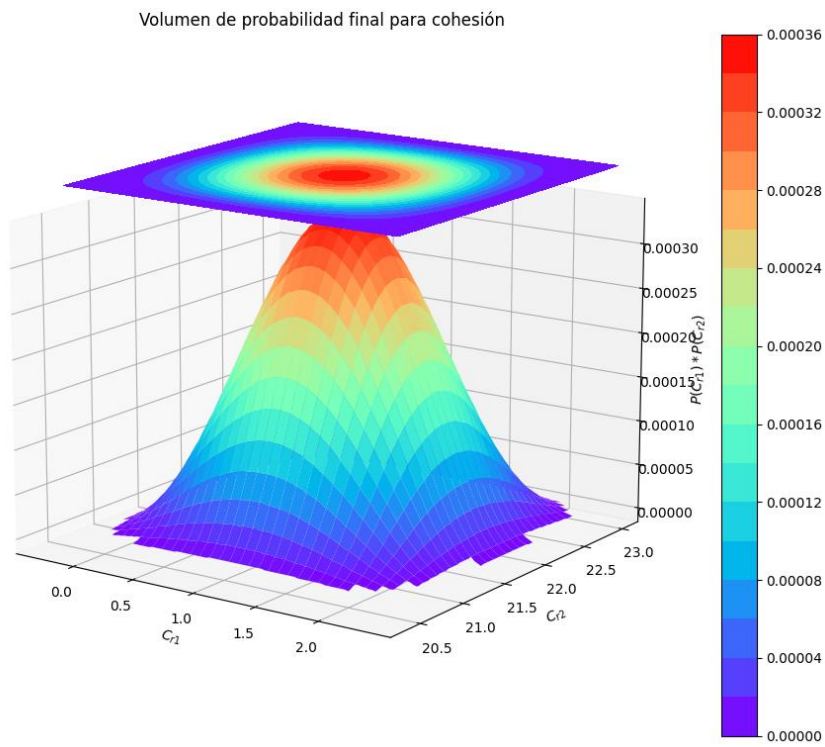


Figura 4-28. Volumen de probabilidad final para cohesión

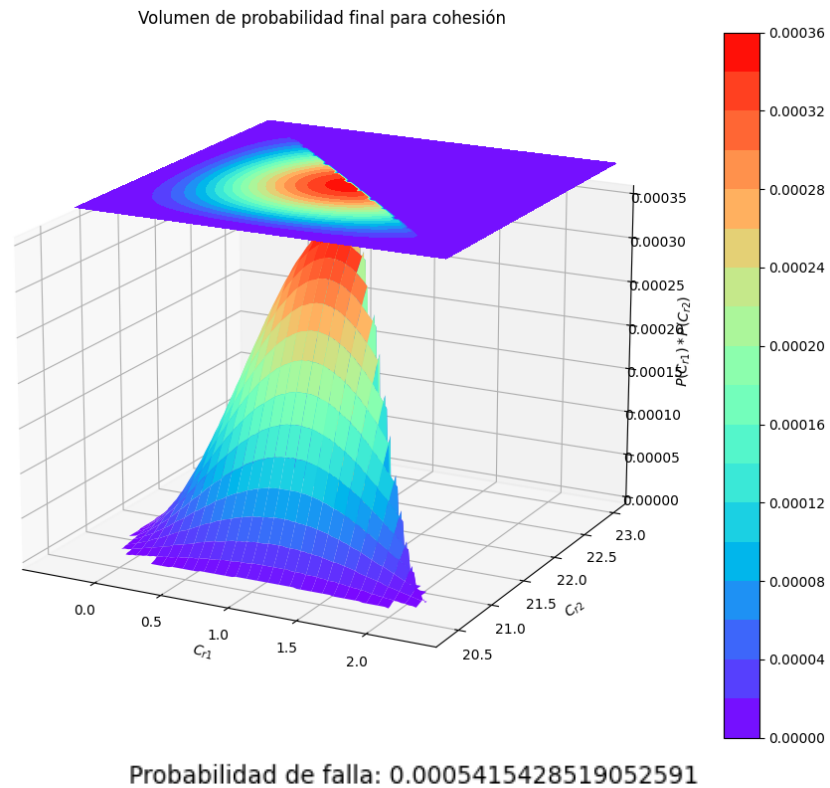


Figura 4-29. Zona de falla sobre volumen de probabilidad final para cohesión

4.2.5 Probabilidad conjunta en red polar - Bloque sur

Para el bloque sur se define la orientación de 230° para talud crítico, y siguiendo la misma metodología plateada se obtiene los siguientes resultados para los diferentes mecanismos de falla.

- Falla planar - Bloque Sur

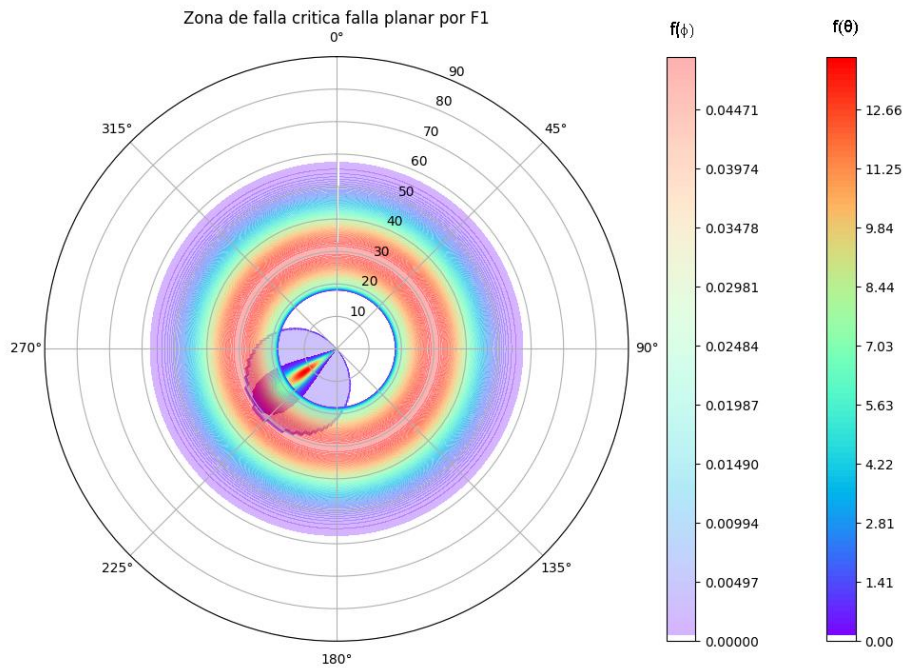


Figura 4-30. Contornos de probabilidades, falla planar para F1

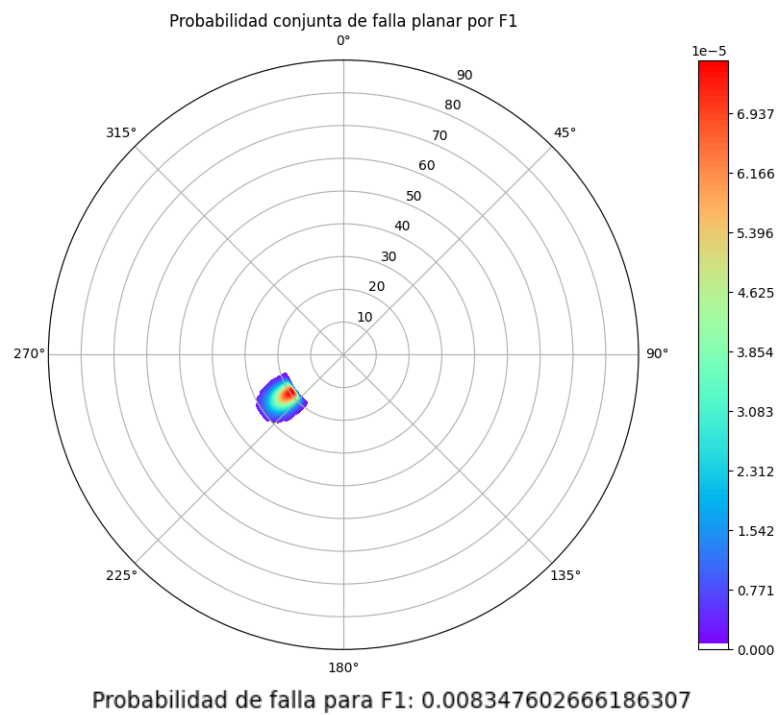


Figura 4-31. Zona de falla crítica por falla planar para F1

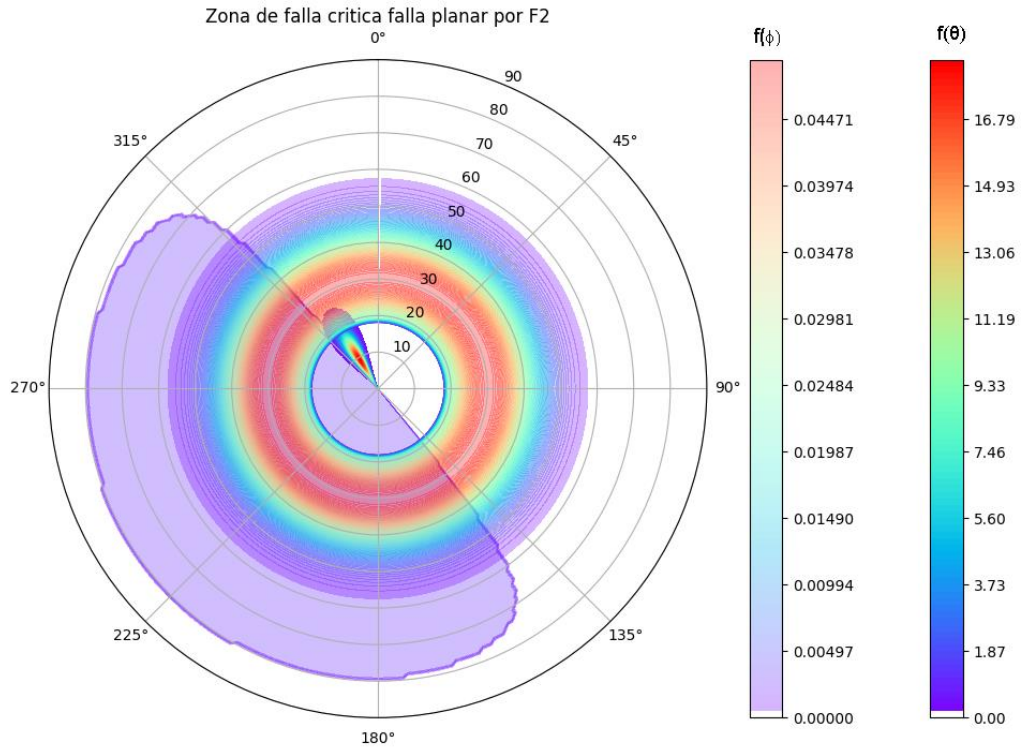
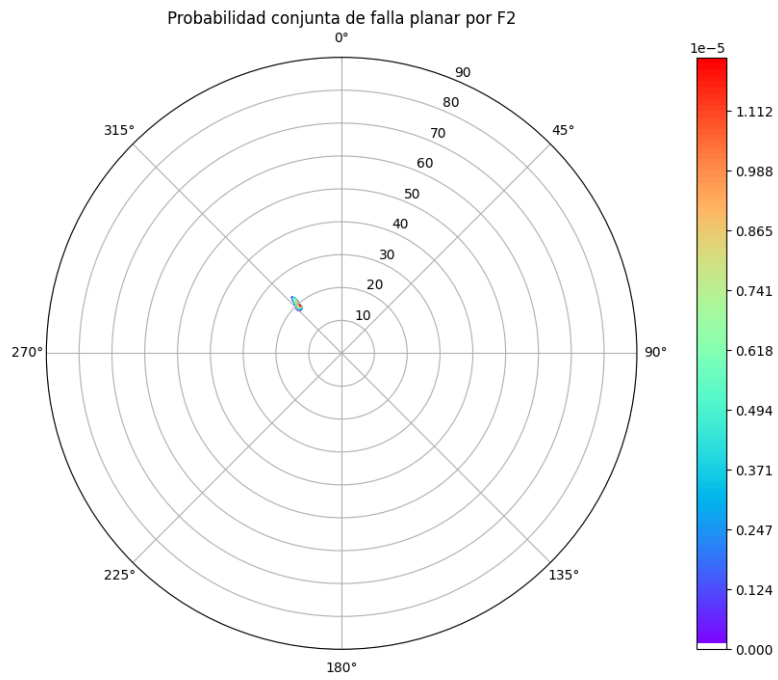


Figura 4-32. Contornos de probabilidades, falla planar para F2



Probabilidad de falla para F2: 9.205237601192856e-05

Figura 4-33. Zona de falla critica por falla planar para F2

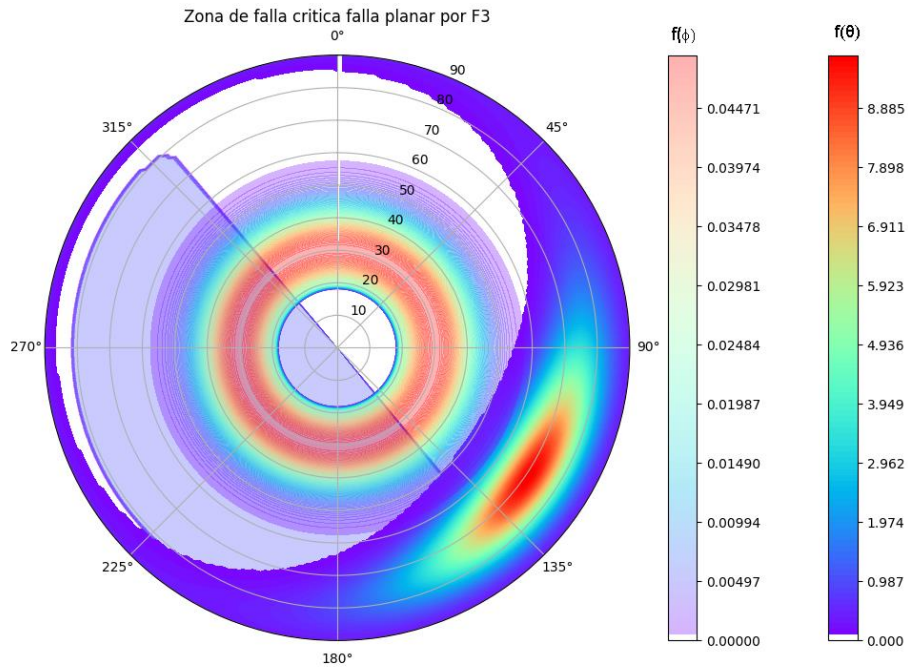
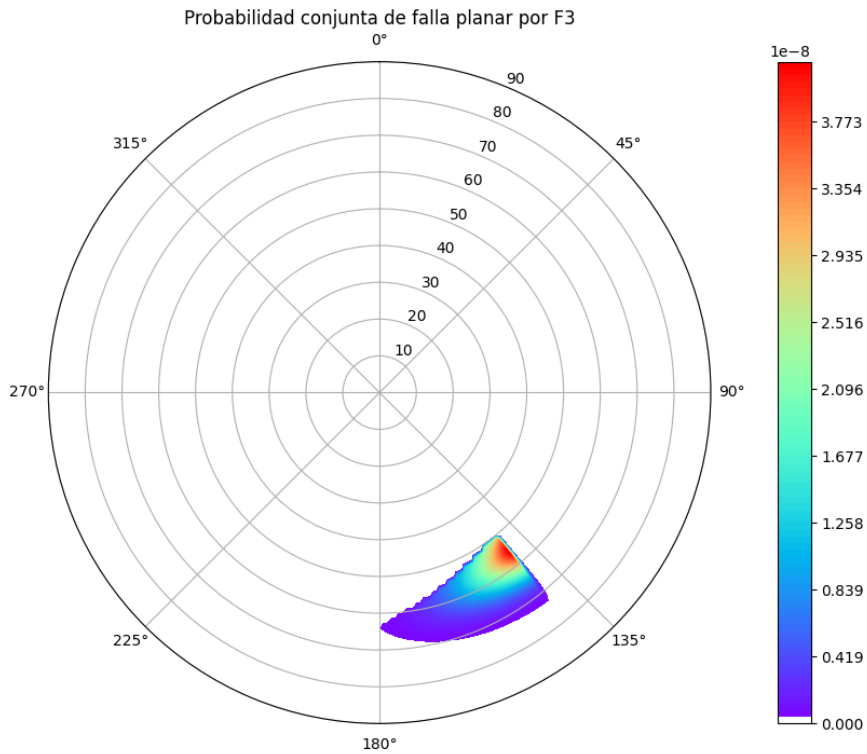


Figura 4-34. Contornos de probabilidades falla planar para F3



Probabilidad de falla para F3: 5.379855704121436e-06

Figura 4-35. Zona de falla critica por falla planar para F3

La probabilidad conjunta calculada para cada una de las familias se encuentra de 0.008 para la Familia 1 y probabilidades muy bajas para familia 2 y 3 del orden de 5.37×10^{-6} .

- Falla por volteo – Bloque Sur

Para la misma orientación de talud de 230° se realiza el cálculo de la probabilidad por falla por volteo, en donde la familia 3 es la única que esta influenciada por las envolventes de la orientación del talud crítico. En la Figura 4-36 y Figura 4-37 se muestran los resultados.

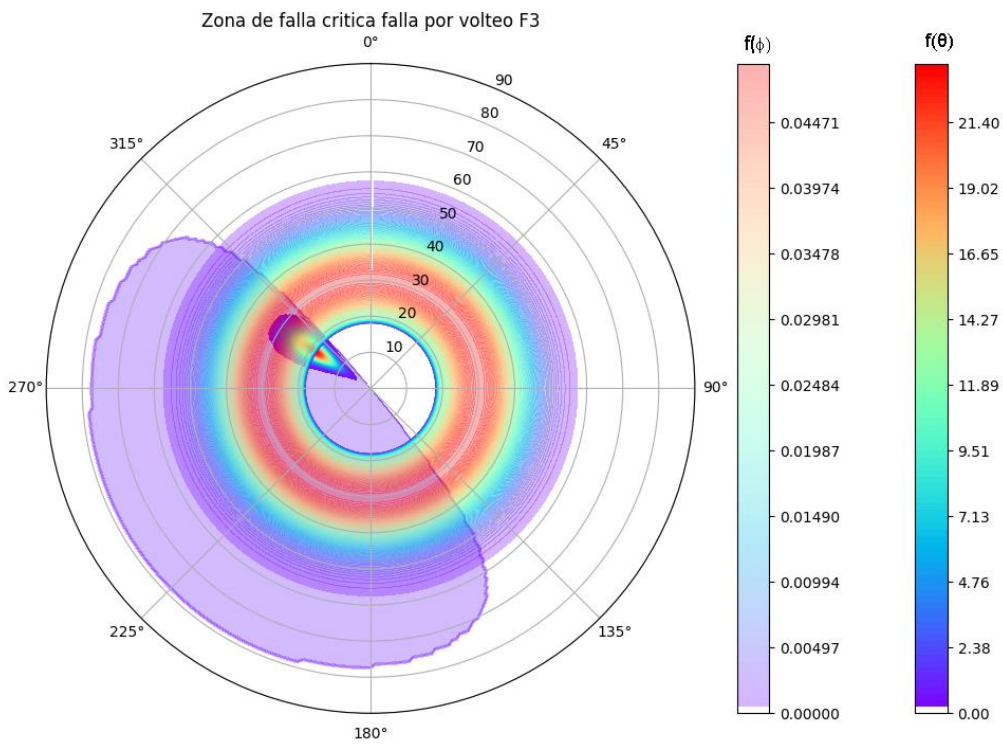
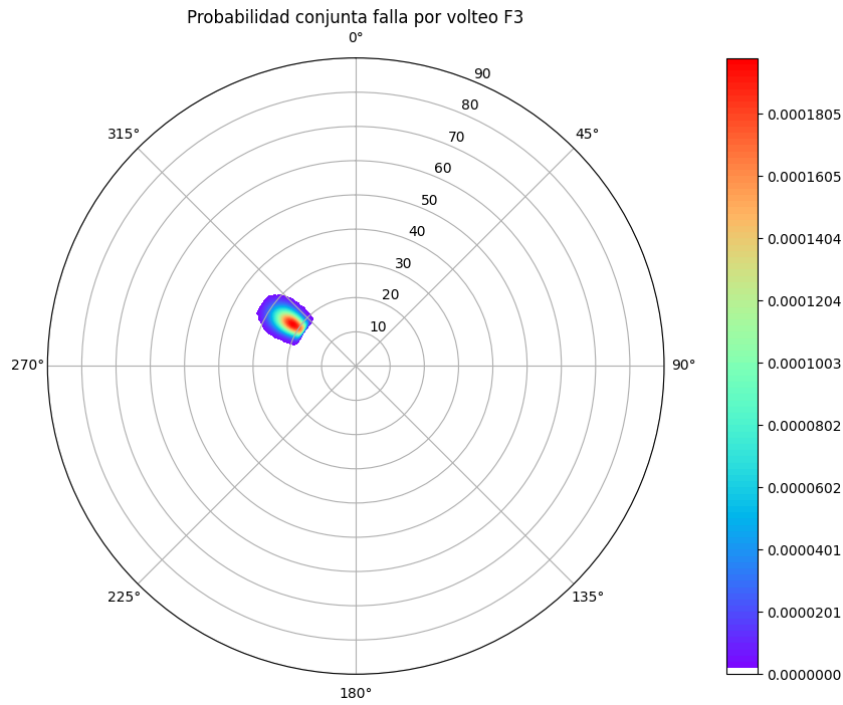


Figura 4-36. Contornos de probabilidades falla por volteo para F3



Probabilidad de falla para F3: 0.02053344513642291

Figura 4-37. Zona de falla critica por falla por volteo para F3

- Falla por cuña - Bloque Sur

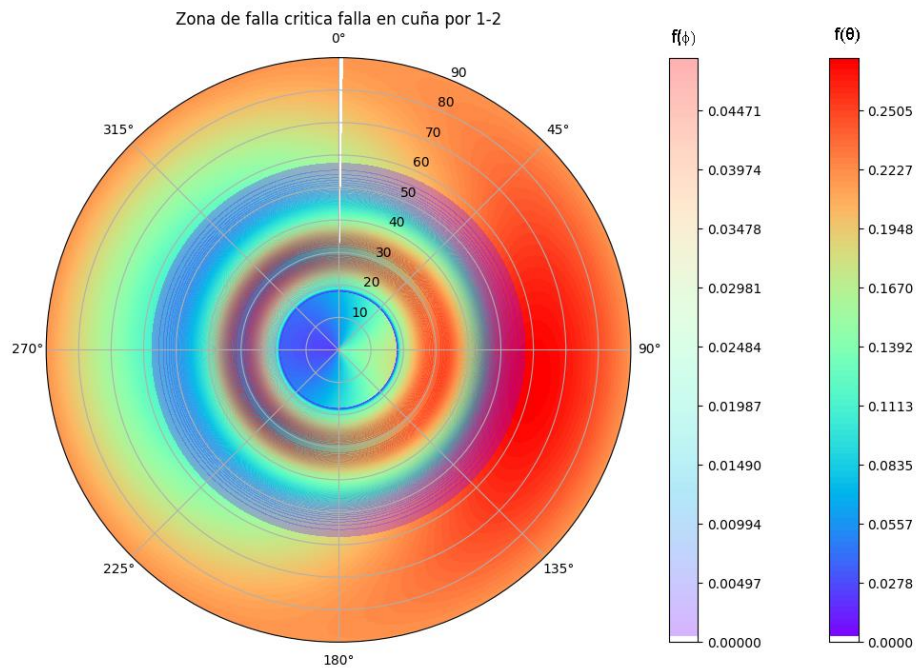
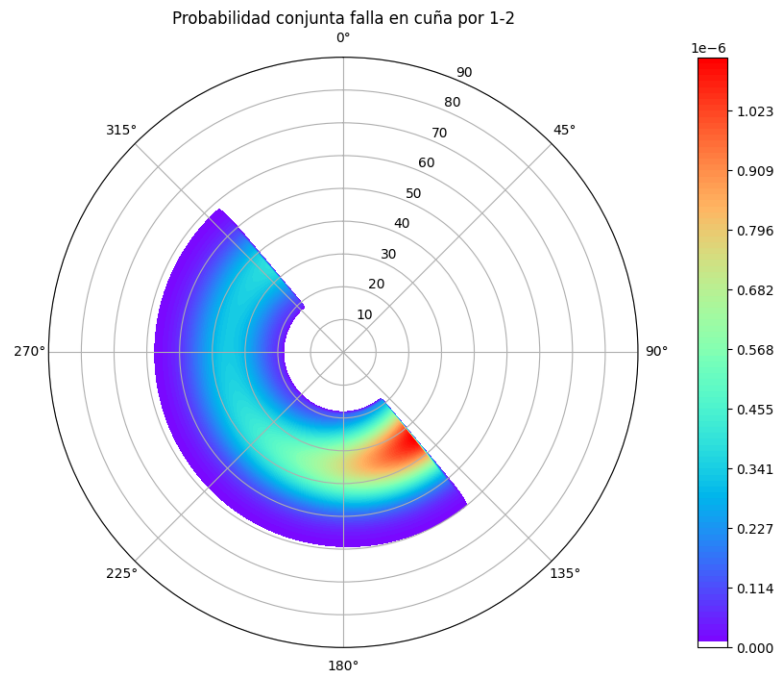


Figura 4-38. Zona de falla critica falla en cuña por F1-F2



Probabilidad de falla para F1-2: 0.002022738775402215

Figura 4-39. Probabilidad conjunta falla en cuña por F1-F2

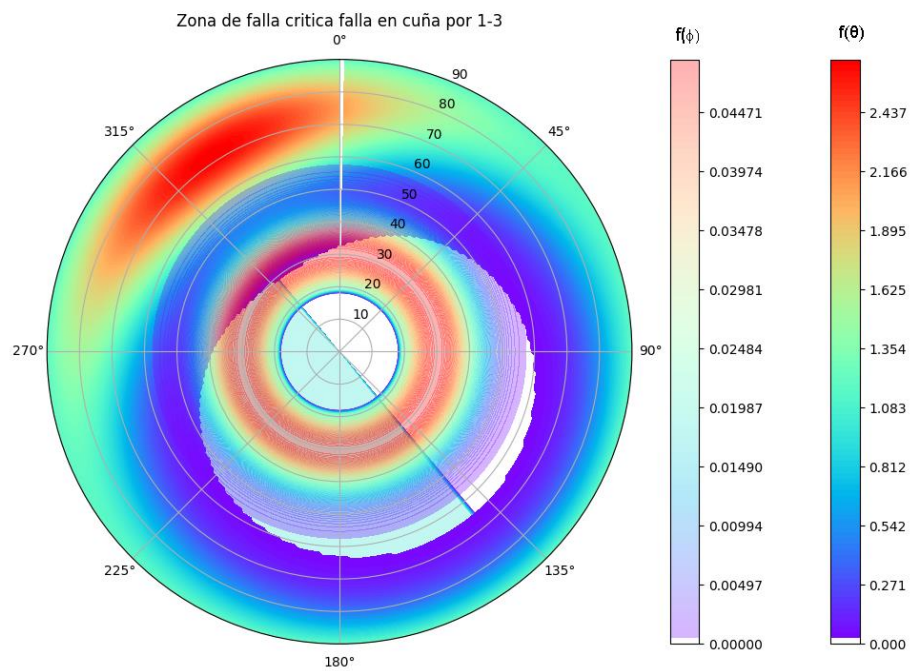
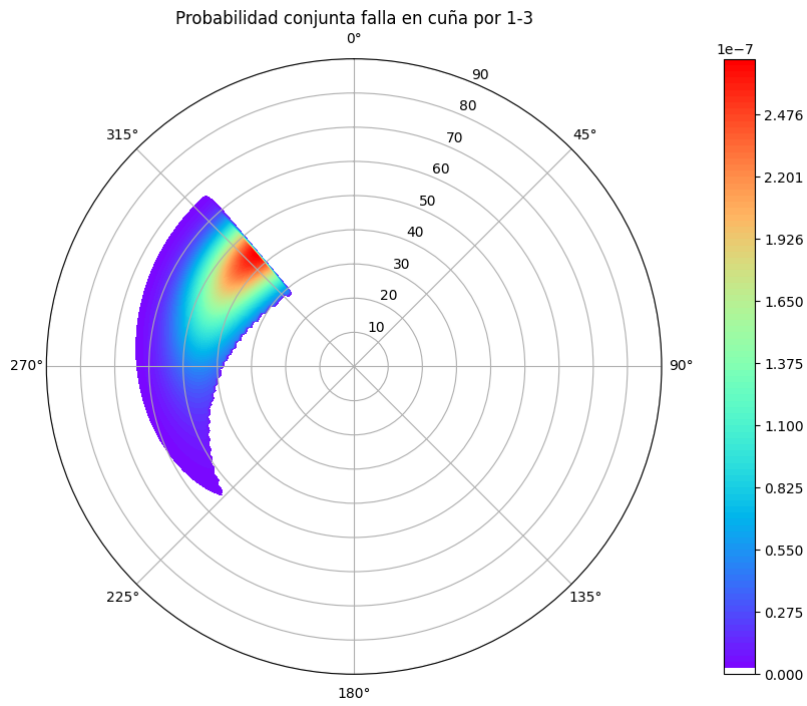


Figura 4-40. Zona de falla critica falla en cuña por F1- F3



Probabilidad de falla para F1-3: 0.00015323812166304968

Figura 4-41. Probabilidad conjunta falla en cuña por F1-F3

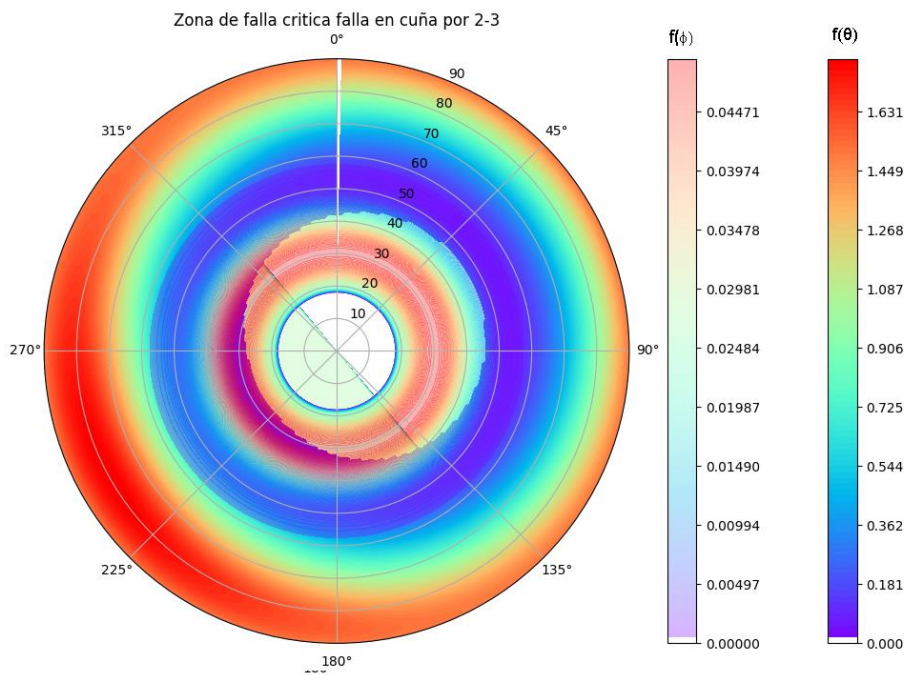
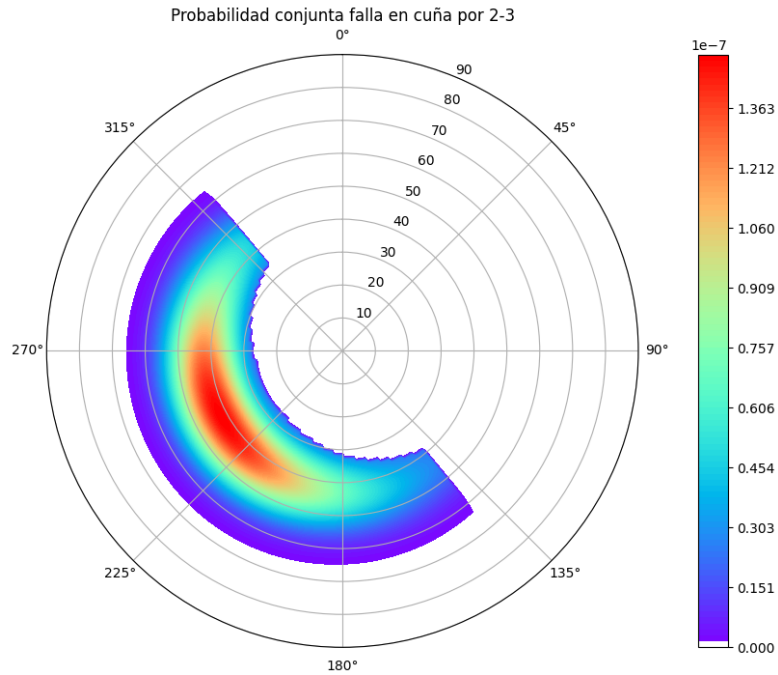


Figura 4-42. Zona de falla critica falla en cuña por F2- F3



Probabilidad de falla para F2-3: 0.0003248044851102126

Figura 4-43. Probabilidad conjunta falla en cuña por F1-F3

4.2.6 Análisis por confiabilidad - Bloque Sur

- Falla planar - Bloque Sur

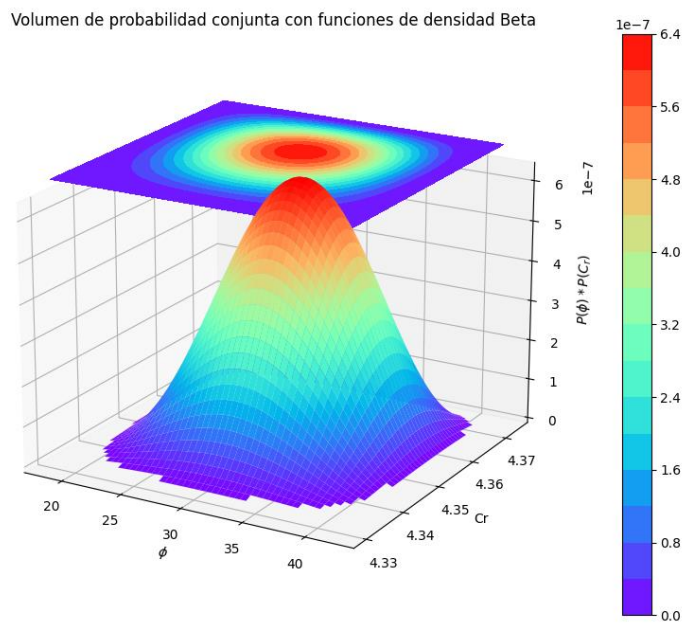


Figura 4-44. Volumen de probabilidad conjunta con funciones falla planar

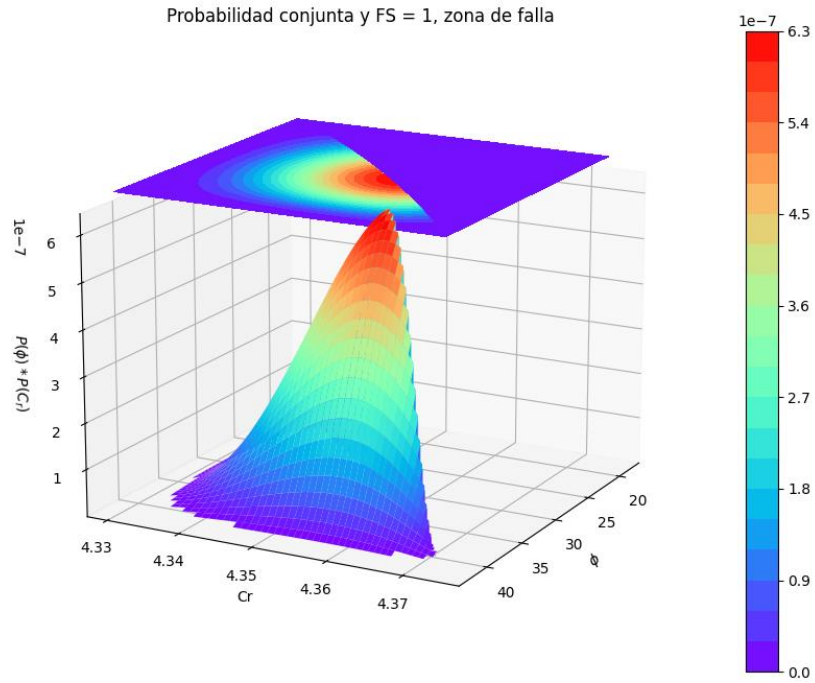


Figura 4-45. Probabilidad conjunta y FS= 1, zona de falla

- Falla por volteo - Bloque Sur

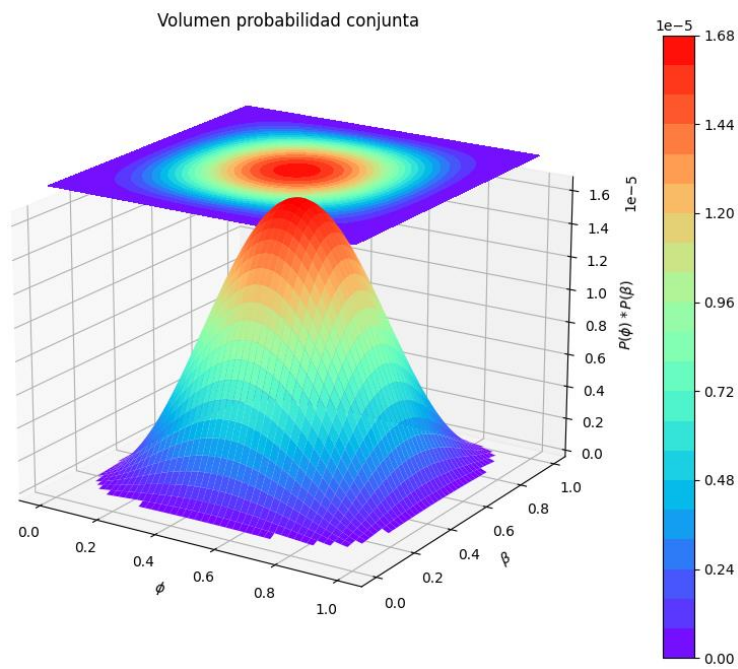
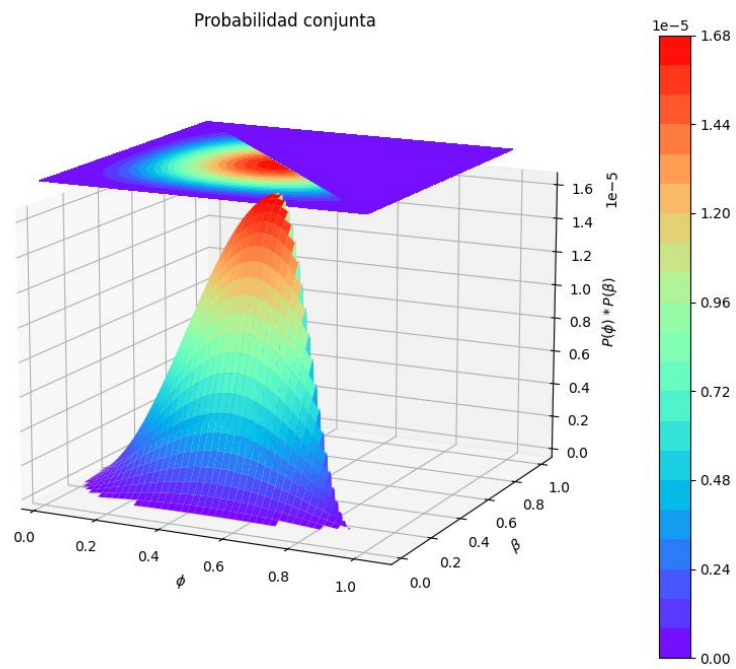


Figura 4-46. Volumen probabilidad conjunta falla por volteo



Probabilidad de falla: 0.0010526031303575954

Figura 4-47. Probabilidad conjunta falla por volteo

- Falla por cuña - Bloque Sur

Volumen de probabilidad final para ángulo de fricción

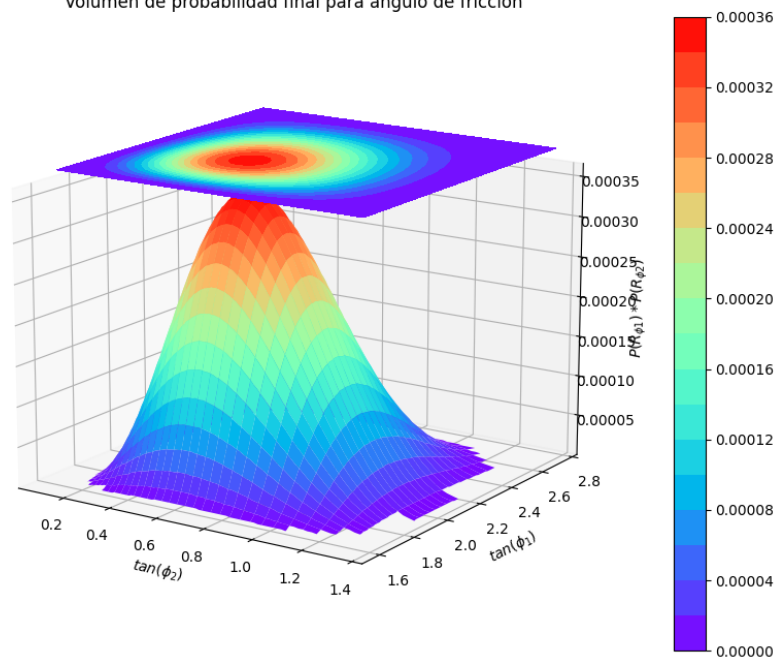


Figura 4-48. Volumen de probabilidad final para ángulo de fricción

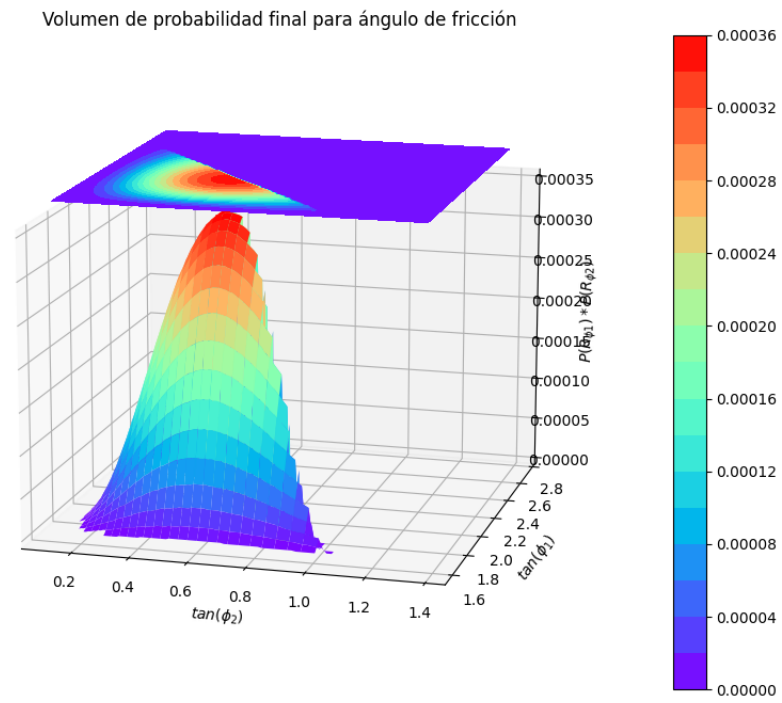


Figura 4-49. Zona de falla sobre volumen de probabilidad final para ángulo de fricción

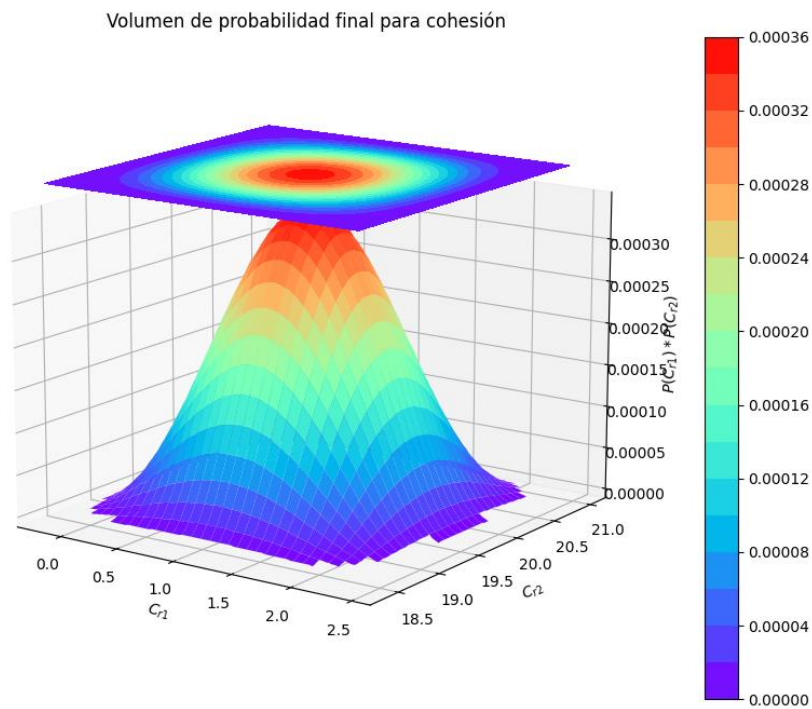


Figura 4-50. Volumen de probabilidad final para cohesión

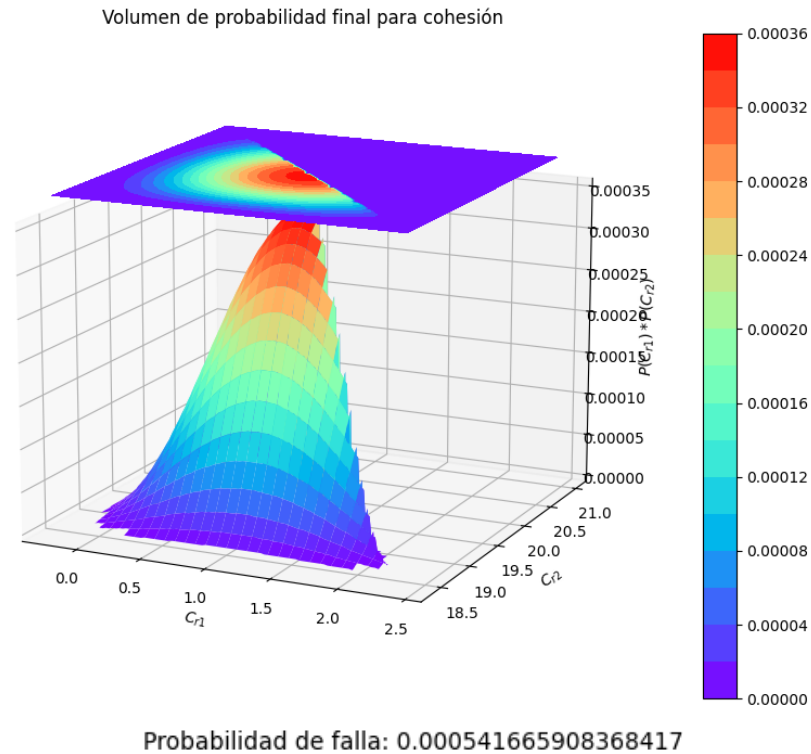


Figura 4-51. Zona de falla sobre volumen de probabilidad final para cohesión

4.3 Análisis de resultados

Aplicada la metodología de la evaluación probabilística de estabilidad utilizando probabilidad conjunta para los bloques norte y sur de la mina del caso de estudio, se consiguen los siguientes resultados de la probabilidad de falla:

- Los resultados obtenidos para la probabilidad de falla mediante el análisis cinemático en los diferentes mecanismos de falla y las familias de discontinuidades del bloque norte, analizados dentro del rango de orientación de 215° , varían entre $5.8E-7$ y 0.02 . En cuanto a la condición cinemática, los resultados oscilan entre $2.36E-06$ y 0.0072 .

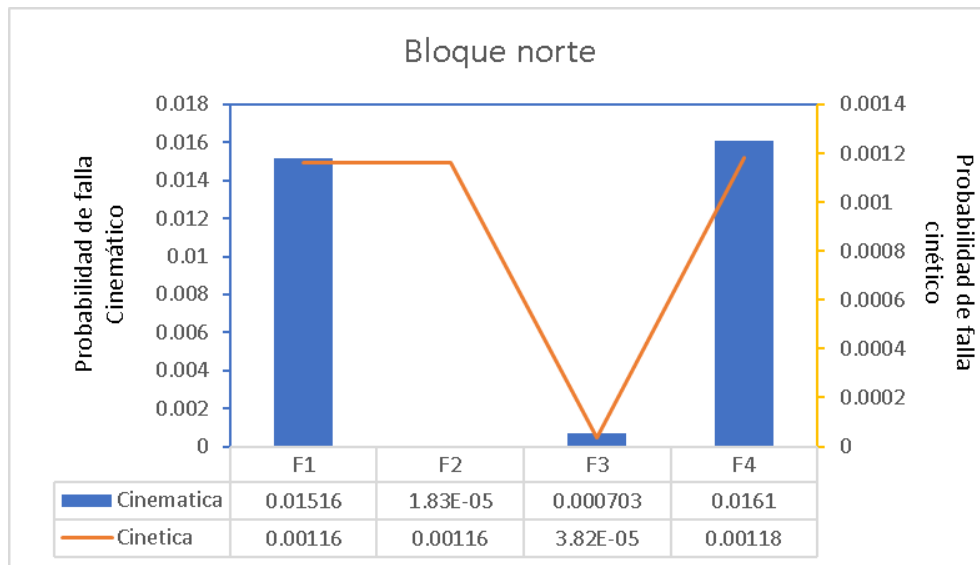


Figura 4-52. Probabilidad de falla mecanismo de falla planar Bloque Norte

Con respecto a los resultados del análisis cinemático del bloque norte en falla planar, se identifica que podrá existir mayor probabilidad de falla en las familias 1 y 4, esto dado a que la envolvente de la orientación del talud crítico 215° con el cual se analiza la probabilidad conjunta de falla, índice directamente en la probabilidad de Fisher calculada para dichas familias, caso contrario de las otras 2 familias que se localizan fuera del rango de influencia de las envolventes de la orientación de talud asignada. Para el bloque norte en el análisis probabilístico cinético se obtiene que la tendencia de la probabilidad de falla es similar a la cinemática, pero de menor magnitud; estas diferencias en las probabilidades de falla cinemática y cinética se presentan por las características específicas del talud, como la geometría, las propiedades geotécnicas de la roca y demás variables.

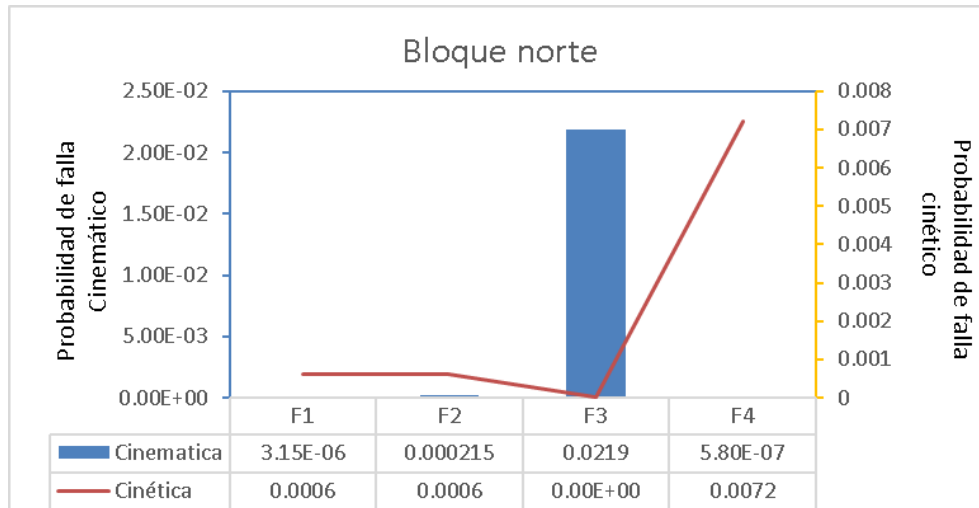


Figura 4-53. Probabilidad de falla mecanismo de falla por volteo Bloque Norte

Para el caso de la falla por volteo, se identifica que la familia 3 tiene mayor probabilidad de falla en el análisis cinemático para la orientación de talud crítico definido para el bloque norte, análogo a la tendencia por falla planar. Para este caso se obtienen probabilidades de falla menores en condición cinemática en comparación a los resultados por probabilidad cinética, esto se debe a que la probabilidad cinemática se basa en la geometría del talud, orientación de planos de falla y característica geomecánicas del talud, sin tener en cuenta los efectos de las fuerzas actuantes como el sismo dentro del análisis probabilístico de falla.

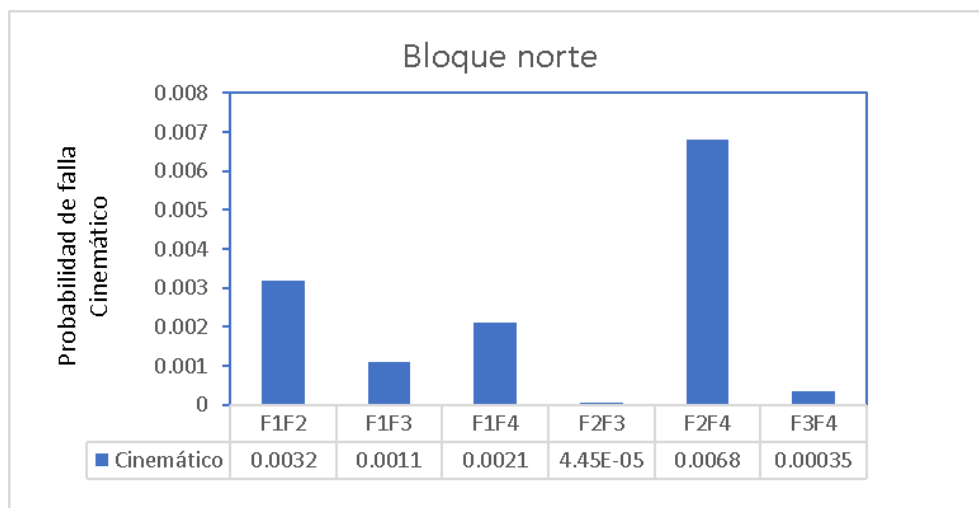


Figura 4-54. Probabilidad de falla mecanismo de falla por cuña Bloque Norte

El resultado de análisis cinemático con enfoque probabilístico en falla por cuña para el bloque norte muestra valores de probabilidad entre $4.45E-5$ y 0.0068 . Con la probabilidad mayor la cuña formada por las familias 2 y 4.

- Para el bloque Sur con orientación de talud crítico de 230° , se tienen probabilidades en condición cinemática de 0.0204 a $5.37E-6$ y en condición cinética de 0.0016 a $9.24 E-7$.

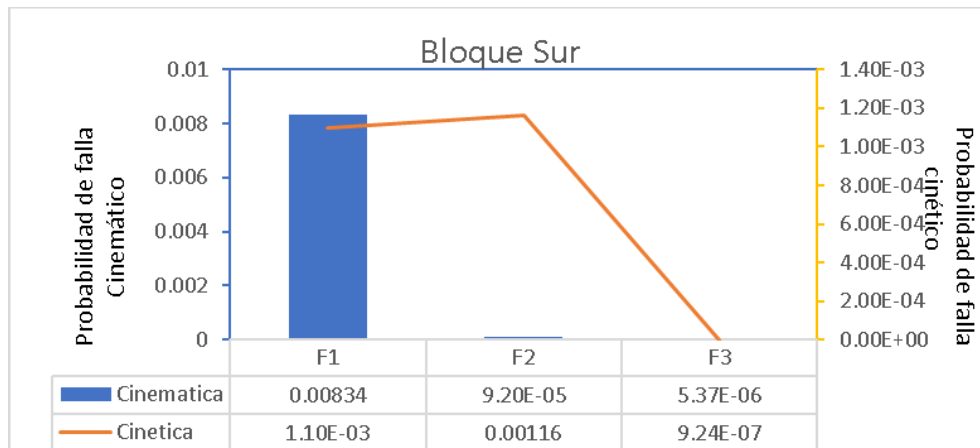


Figura 4-55. Probabilidad de falla mecanismo de falla planar Bloque Sur

Con un comportamiento similar al del bloque norte, se obtiene tendencia similar en las probabilidades, pero la probabilidad cinemática mayor a la cinética.

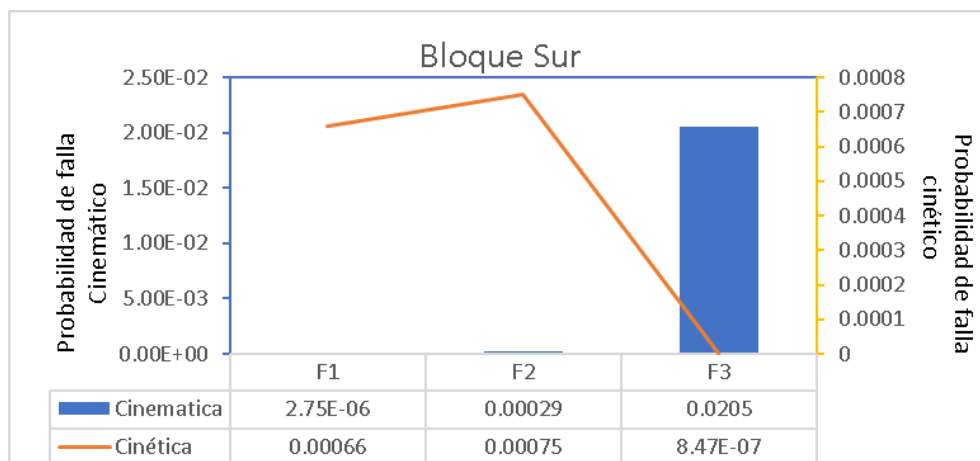


Figura 4-56. Probabilidad de falla mecanismo de falla por volteo Bloque Sur

Los resultados de las probabilidades para falla por volteo indican la estimación de la probabilidad cinemática menor a la probabilidad cinética, en la primera solo se tiene en cuenta la geometría y las características geotécnicas del material, en la segunda se implica las fuerzas actuantes como el sismo.

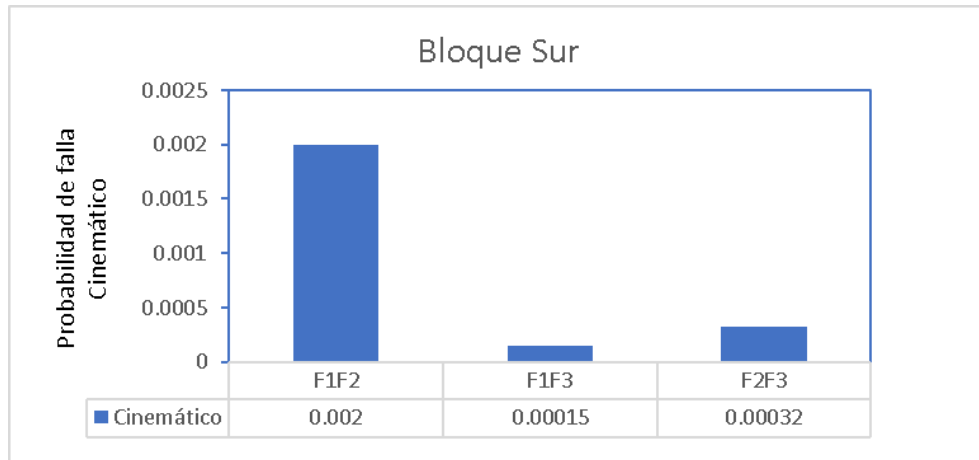


Figura 4-57. Probabilidad de falla mecanismo de falla por Cuña Bloque Sur

Los resultados de probabilidad de falla por análisis cinemático, muestra valores de probabilidad de fallade 0.002 en la cuña formada por el plano 1 y 2, siendo la orientación de esta intersección la de mejor orientación e incidencia por la orientación del talud critico de 230°.

En los análisis de probabilidad realizados, se obtuvieron probabilidades menores a 0.0205, lo cual indica valores bajos para la ocurrencia de deslizamientos o fallas en los taludes. Sin embargo, determinar si estos valores son aceptables o no depende de varios factores relacionados con la variabilidad e incertidumbre de las variables y la interacción en el cálculo de la probabilidad de falla. Además, otros aspectos como las condiciones geológicas, las características del talud, las fuerzas actuantes y los criterios de diseño también deben considerarse en el análisis global de la estabilidad del talud.

5. Conclusiones y recomendaciones

5.1 Conclusiones

El principal aporte de este trabajo es el desarrollo de una metodología para la evaluación de la estabilidad de taludes rocosos mediante probabilidad conjunta y la representación gráfica de probabilidades de red estereográfica y se complementa con una evaluación cinético de confiabilidad de forma gráfica para cada uno de los mecanismos de falla en taludes en roca.

Esta metodología se diferencia de otras metodologías reportadas en la literatura técnica porque plantea un análisis estadístico y probabilístico de las orientaciones de las discontinuidades con la función de probabilidad de Fisher que es una función bivariada, luego es representa de forma gráfica cada una de las familias mediante contornos de probabilidad para los diferentes datos en la red polar, es decir que cada punto dentro de la red tendrá asociado un valor de probabilidad con respecto a la orientación y la desviación angular entre la familia de discontinuidad y los demás datos.

Dentro de la metodología se plantea la representación del ángulo de fricción no como un solo valor sino como una zona de probabilidad o anillo de probabilidades en 2D en la red estereográfica la cual está definida por la función de densidad de probabilidad normal truncada, y finalmente se limita una zona mediante el uso de una envolvente de la orientación de un talud crítico, lo que permite obtener una probabilidad de falla respecto a la probabilidad de Fisher y las demás probabilidades calculadas para cada una las familias de discontinuidades identificadas, metodología que es aplicada a cada uno de los mecanismos de falla.

Como complemento a la metodología planteada con el análisis cinemático, se incluye el análisis cinético, en donde se presenta gráficamente la probabilidad conjunta de variables de resistencia al corte como cohesión y fricción con un análisis estadístico y probabilístico y la implementación de funciones de densidad de probabilidad definidas para cada una paramero de acuerdo con su tendencia y comportamiento; varios autores hacen referencia al uso de funciones como la función normal, normal truncada y Beta para este tipo de variables.

El desarrollo de un código de computador en lenguaje de programación Python en su aplicación PyCharm Community como herramienta para la obtención de resultados de la metodología propuesta permite de forma práctica y rápida la visualización de los cálculos de la estabilidad de fallas con control cinemático en taludes en roca mediante la probabilidad conjunta de variables típicas usadas en los análisis de estabilidad y que serán los datos de entrada en el análisis como son los datos estructurales y valores de resistencia al corte.

La aplicación de la metodología y la implementación de la herramienta creada en esta tesis en un caso de estudio real permite visualizar el comportamiento del código frente a una cantidad considerable de datos, la obtención de resultados de forma rápida, poder variar la orientación del talud según la necesidad de análisis y poder comparar resultados con lo que es percibido en campo.

Se concluye que a través de la implementación de herramientas de uso probabilístico es posible evaluar cada una de las tendencias de los parámetros geométricos y geomecánicos de los macizos rocosos y evaluar su variabilidad para la aplicación en modelos de análisis de estabilidad en taludes en rocas.

Se destaca el uso de metodologías y herramientas comúnmente implementadas en el análisis de estabilidad de taludes en roca como la red estereográfica y la graficación de probabilidades en ésta, dando la pauta para la aplicación de métodos gráficos en la interpretación de modelos probabilísticos en taludes rocosos.

Es de considerar que el desarrollo de nuevas metodologías y formas de interpretar los mecanismos de falla de taludes en roca como se presentó en este informe, conducen a la

implementación de nuevas herramientas y metodologías para el análisis de datos y resultados dentro del análisis de estabilidad de taludes de una forma gráfica con la implementación de elementos o recursos comúnmente usados en la práctica de la geología y geotecnia de taludes en roca como es el caso de la red estereográfica.

Dado que la cohesión en las rocas es una propiedad que se deriva de las fuerzas de atracción entre las partículas individuales de la roca y es esencial para la resistencia y estabilidad en los de los taludes, la cohesión en los diferentes mecanismos de falla hace referencia a una cohesión aparente, pues se refiere a la resistencia al corte que se puede observar en un talud debido a la interacción entre las superficies de discontinuidad y la geometría del talud. Esta resistencia aparente puede ser menor que la cohesión verdadera de las rocas pues las discontinuidades debilitan el macizo rocoso.

En el análisis de la estabilidad de taludes en suelos u otros medios particulados, la cohesión y el ángulo de fricción son parámetros clave para determinar el comportamiento y la resistencia del material. Sin embargo, en el caso de los planos de discontinuidad en rocas, la relación entre la cohesión y el ángulo de fricción puede ser más compleja; en el caso de las rocas y los planos de discontinuidad, la cohesión puede ser menos significativa en comparación con el ángulo de fricción ya que las discontinuidades en las rocas pueden tener una cohesión intrínseca baja o nula. En cambio, el ángulo de fricción entre las superficies de discontinuidad juega un papel importante en la resistencia al deslizamiento y la estabilidad de los taludes rocosos. La relación entre la cohesión y el ángulo de fricción en los planos de discontinuidad puede variar según las características geológicas y geotécnicas específicas de cada macizo.

La metodología planteada de las discontinuidades asociadas a parámetros de resistencia puede conducir a resultados razonables de la probabilidad de falla, sin embargo, es de aclarar que la determinación precisa de la probabilidad de falla depende de varios factores o consideraciones adicionales, como la incertidumbre de los parámetros y variables involucradas, junto con la calidad y cantidad de datos disponibles para la adecuada caracterización de las propiedades geomecánicas del macizo rocoso y las discontinuidades. Pero dichas consideraciones no implican que los resultados obtenidos en el estudio sean erróneos. Más bien, los resultados son una aproximación a la probabilidad de falla pues se tiene en cuenta los parámetros y variables relevantes dentro

del análisis de estabilidad. No obstante, como recomendación para otros estudios a futuro, se podría plantear un análisis integral y considerar otras variables y factores relevantes para obtener resultados más precisos y confiables.

Los parámetros de resistencia registrados y obtenidos en diferentes campañas de ensayos, presentar alguna dispersión que puede estar influenciada por factores como el grado de alteración de la roca, diferencias en la composición mineralógica, rugosidad de las discontinuidades o localización geográfica de puntos de ensayo. Los resultados de parámetros de resistencia fueron obtenidos de diferentes campañas de ensayos en diferentes sectores de la mina para una clasificación de material similar, y dada la condición de exposición y quizás meteorización de la roca en la superficie, rugosidad o plano de falla tenida en cuenta en el ensayo, esto puede influir en los resultados y variabilidad de los resultados. Se entiende además que la dispersión en los resultados de los parámetros de resistencia de las rocas es natural y quizás esperada debido a la heterogeneidad inherente de los materiales rocosos. Pero es importante que en futuras investigaciones realizar un muestreo más amplio y utilizar métodos de ensayo específicos para obtener datos precisos y representativos de los parámetros de resistencia.

Quizás los datos de resistencia presentados en el análisis estadístico no sean suficientes, sin embargo, gracias a los datos recolectados, se pudo llevar a cabo una aproximación del análisis de tendencia y obtención de resultados medios de cada uno de los parámetros requeridos. Es importante destacar que un tamaño de muestra más grande habría proporcionado una base más sólida para el análisis estadístico y podría haber conducido a resultados más confiables. No obstante, esto no significa que los análisis realizados con los datos disponibles carezcan de validez dentro del estudio.

Con relación a la aceptabilidad de la probabilidad de falla, la literatura reporta algunos criterios de aceptación basados en la escala del talud y las consecuencias de falla de este, este último derivado del análisis de riesgo realizado para la zona de influencia del talud, según la geometría del talud fallado. Dependiendo del tipo de escala (berma, interrampa o global) y si las consecuencias de falla son altas, medias o bajas, se establecen valores máximos de aceptación para la probabilidad de falla. Según lo establece Read & Stacey (2009). En el caso más crítico, es decir, en una escala global con altas consecuencias en caso de falla, la probabilidad de falla máxima aceptable es de 5%. Teniendo en cuenta que

en los análisis de probabilidad realizados se obtuvieron probabilidades de falla menores a 2%, se puede considerar que el talud de la mina del caso de estudio cumple con los criterios de diseño reportados en la literatura.

Cabe resaltar que este es sólo uno de los criterios que se debe tener en cuenta para determinar que la probabilidad de falla es aceptable. Es claro que no se puede considerar únicamente el valor de la probabilidad en el análisis de estabilidad de taludes en roca, ya que hay otros factores adicionales como las condiciones geológicas, las características del talud y las cargas aplicadas. Adicionalmente, cada caso de análisis de probabilidad de un talud rocoso o mina es particular y requiere de datos de resultados de probabilidades o registro de las experiencias previas en situaciones similares, que permitan asociar el comportamiento de la estabilidad de los taludes de la mina y las condiciones propias de orientaciones, características geológicas de los materiales y registros de falla recopilados a través del tiempo, para poder asociar la probabilidad de falla a valores estimados de análisis previos y evaluar si es aceptable las probabilidades en el contexto del proyecto en sí, en conjunto con los criterios de seguridad establecidos en la mina.

Ahora bien, desde un punto de vista de seguridad de la mina, un valor bajo de probabilidad de falla es un buen indicador de que el macizo rocoso es estable y por ende de que su diseño es seguro; esto tiene como consecuencia práctica que tanto el personal como la maquinaria están expuestos a un muy bajo riesgo de accidentalidad por deslizamientos del talud, con las implicaciones legales y económicas que esto supone. Sin embargo, desde una perspectiva económica, un factor de seguridad tan alto puede ser una limitante para el aprovechamiento de material en la mina, al impedir pendientes más pronunciadas que permitieran una mayor extracción y rentabilidad. Se puede intuir una oportunidad de optimización del diseño del talud en esta baja probabilidad de falla, lo cual permitiría aprovechar más el terreno de extracción, alargar su vida productiva y por tanto los beneficios económicos, sin sacrificar la seguridad del personal de campo.

En los análisis de probabilidad realizados, se obtuvieron probabilidades menores a 0.0205, lo cual indica valores bajos para la ocurrencia de deslizamientos o fallas en los taludes. Sin embargo, determinar si estos valores son aceptables o no depende de varios factores relacionados con la variabilidad e incertidumbre de las variables y la interacción en el cálculo de la probabilidad de falla. Además, otros aspectos como las condiciones

geológicas, las características del talud, las fuerzas actuantes y los criterios de diseño también deben considerarse en el análisis global de la estabilidad del talud.

5.2 Recomendaciones

Con el desarrollo de este trabajo se pueden considerar las siguientes recomendaciones para el mejoramiento de las metodologías propuestas:

Debido a la gran cantidad de datos a graficar en la red polar para los análisis de la metodología por análisis cinemático y cinético, se recomienda el uso de herramientas de programación para tener más rapidez en el procesamiento de los datos y obtención de resultados, que en este caso se pone a disposición el programa PCTR desarrollado.

Dentro de los análisis estadísticos y probabilísticos es necesario emplear una cantidad significativa de datos para tener más precisión en los resultados y así mismo una tendencia de datos más adecuada para identificación de funciones de densidad de probabilidad que se ajusten a dicha tendencia o que se pueda identificar dentro de los referentes dados con respecto a las funciones propias para cada parámetro.

En este informe se presenta el análisis de estabilidad de taludes en roca de manera probabilística con el estudio de dos o tres variables, ya sean fuerzas resistentes como es la cohesión y ángulo de fricción, fuerzas actuantes como la aceleración sísmica, y en algunos casos la inclinación de la superficie de falla, es sabido que dentro de la estabilidad de los taludes influyen muchas más variables como son las geométricas propias del talud o del macizo rocoso y otras fuerzas actuantes o resistentes adicionales que debido a su carácter probabilístico permitirían un análisis más completo dentro de la metodología, es recomendable reconocer cada uno de estos parámetros e implementarlos dentro del sistema de multivariabilidad y cálculo de probabilidad conjunta de más de dos variables.

5.3 Limitaciones

Durante el desarrollo de esta tesis, se han identificado diversas limitaciones que vale la pena destacar. Estas limitaciones abarcan desde la disponibilidad limitada de información

bibliográfica sobre el tema hasta el uso de un número reducido de datos, los cuales podrían haber influido en la interpretación y generación de los resultados. Además, se han encontrado limitaciones asociadas a la programación y desarrollo de la herramienta de computado. Sin embargo, es importante resaltar que, al identificar y reconocer estas limitaciones, se puede obtener una idea clave para orientar investigaciones futuras o el desarrollo de nuevos estudios relacionados con el tema en cuestión.

La investigación bibliografía relacionada con el análisis probabilístico de estabilidad de taludes en roca permitió identificar que la metodología utilizada en la tesis de investigación está poco desarrollada, solo hasta los últimos años que se empieza a implementar metodologías alternativas para el estudio de la estabilidad de taludes con en uso de herramientas de probabilidad en el estudio de la estabilidad de fallas.

La presente tesis explora un enfoque novedoso y poco común al utilizar la probabilidad conjunta y su representación gráfica. Aunque es cierto que hay escasas referencias específicas sobre el tema de investigación para poder comparar los resultados, esto no disminuye la validez de la tesis. Por el contrario, brinda una oportunidad para futuras investigaciones y el avance del conocimiento en el campo de la representación gráfica de la probabilidad conjunta en el ámbito de la geotecnia.

Es sabido que los análisis probabilísticos requieren datos confiables y suficientes para realizar estimaciones y cálculos estadísticos y de tendencia de resultados. Para esta investigación se usó una cantidad limitada de datos recolectados y obtenidos en varios periodos de tiempo y por diferentes profesionales que fueron alimentando una base de datos, esto puede implicar que la calidad de los datos pueda variar e introducir alguna incertidumbre en los resultados, es por eso que se recomienda el uso de una cantidad significativa de datos para poder estudiar con más precisión la variabilidad de los datos y evaluar de manera más fiable la tendencia de estos en los modelos estadísticos y probabilísticos. La tesis doctoral de Rodrigo Hernández Carrillo considera de manera significativa la disponibilidad limitada de información, así como la incertidumbre relacionada con la veracidad de los datos recopilados a lo largo del tiempo, los cuales se utilizan para alimentar la base de datos destinada al análisis de la estabilidad de los taludes. Es importante destacar que esta incertidumbre puede generar cambios en los resultados a medida que se obtiene nueva información y se actualiza la base de datos.

En el desarrollo de la tesis se asumen la independencia de las variables de resistencia implicadas en la estabilidad de los taludes, lo que significa que no se consideran las posibles interacciones o dependencias entre ellos. Es posible que, en la realidad existan correlaciones o dependencias que puedan influir en los resultados. Para el desarrollo de la probabilidad conjunta se trabaja con un modelo simplificado con variables independientes lo que implica intersección de variables y por ende multiplicación de probabilidades. Si se quisiera ser estricto en los análisis de variables dependientes se debe contemplar la probabilidad total como la unión de las probabilidades, lo que daría para el desarrollo de una investigación más compleja.

Durante el desarrollo de la metodología de análisis probabilístico con enfoque cinético en cuña, se afronta con la complejidad de las ecuaciones involucradas en el análisis de estabilidad, el cual abarcan diversas variables. La implementación de esta metodología dentro del programa se presenta para indicar valores estimados de las variables. Es importante completar el código con los cálculos correspondientes de estas variables, los cuales requieren análisis matricial más compleja, según la propuesta de Granados (2009). De esta manera, se logrará una implementación completa y precisa de la metodología, permitiendo obtener resultados confiables en el análisis de estabilidad mediante el enfoque cinético en cuña.

Una limitación en el desarrollo del programa radica en la arquitectura de este, ya que podrían haberse generado complejidades innecesarias en el código. Esto puede ser problemático, especialmente considerando que el programa maneja grandes conjuntos de datos y realiza cálculos complejos, lo cual puede afectar su eficiencia sin llegar a generar resultados incorrectos. Para abordar esta limitación, es recomendable contar con la ayuda de un especialista en la materia, quien puede optimizar los procesos y mejorar la arquitectura del programa. De esta manera, se puede lograr un mejor rendimiento y eficiencia en la manipulación de datos y cálculos complejos.

A. Anexo: Trabajos desarrollados en análisis probabilístico en taludes en roca

B. Anexo: Funciones de densidad y distribución de Fisher

C. Anexo: Datos estructurales de caso de estudio

D. Anexo: Interfaz del programa

E. Anexo: Código fuente del programa

Bibliografía

- Admassu, Y., & Shakoor, A. (2013). DIPANALYST: A computer program for quantitative kinematic analysis of rock slope failures. *Computers and Geosciences*, *54*, 196–202. <https://doi.org/10.1016/j.cageo.2012.11.018>
- Bell, F.G. (1987). Ground engineer's reference book. Oxford: Butterworth-Heinemann.
- Bolle, A. (1987). *A probabilistic approach of slope stability in fractured rock*. 301–303.
- Butler, R. (1992). *Paleomagnetism : Magnetic Domains to Geologic PALEOMAGNETISM : Magnetic Domains to Geologic Terranes Electronic Edition , September 2004 University of Portland. October.*
- Carter, B. J., & Lajtai, E. Z. (1991). Rock slope stability and distributed joint systems. *Canadian Geotechnical Journal*, *29*(1), 53–60. <https://doi.org/10.1139/t92-006>
- Duzgun, H. S. B., Yucemen, M. S., & Karpuz, C. (2002). A probabilistic model for the assessment of uncertainties in the shear strength of rock discontinuities. *International Journal of Rock Mechanics and Mining Sciences*, *39*(6), 743–754. [https://doi.org/10.1016/S1365-1609\(02\)00050-3](https://doi.org/10.1016/S1365-1609(02)00050-3)
- Einstein, H. H., & Baecher, G. B. (1983). Probabilistic and statistical methods in engineering geology - Specific methods and examples part I: Exploration. *Rock Mechanics and Rock Engineering*, *16*(1), 39–72. <https://doi.org/10.1007/BF01030217>
- Fisher, R. (n.d.). *Dispersion on a sphere*.
- Fisher, R. (1953). Dispersion on a Sphere. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *217*(1130), 295–305. <https://doi.org/10.1098/rspa.1953.0064>
- Gibson, W. (2011). Probabilistic methods for slope analysis and design. *Australian Geomechanics Journal*, *46*(3), 1–12.
- Glynn, E. F. (1979). A probabilistic approach to the stability of rock slopes. *Massachusetts Institute of Technology*.
- Gokceoglu, C., Sonmez, H., & Ercanoglu, M. (2000). Discontinuity controlled probabilistic slope failure risk maps of the Altindag (settlement) region in Turkey. In *Engineering Geology* (Vol. 55). www.elsevier.nl/locate/enggeo
- Gonzalez de Vallejo, L., Ferrer, M., Ortuño, L., & Oteo, C. (2004). *Ingenieria Geologica.pdf*.

- Goodman, R. E. (1989). *Introduction to Rock Mechanics*.
- Granados, A. (2009). Probabilidad de falla de cuñas en roca. In *IV Jornadas académicas de ingeniería de rocas. Universidad nacional de Colombia*. (p. 8).
- Griffiths, D. V., & Fenton, G. A. (2007). Probabilistic Methods in Geotechnical Engineering. *Environmental & Engineering Geoscience*, 1(1), 345.
<https://doi.org/10.2113/gseegeosci.i.1.122>
- Hammah, R. E., Yacoub, T. E., Curran, J. K., & . (2009). Probabilistic Slope Analysis with the Finite Element Method. *American Rock Mechanics Association*, 09, 1–19.
- Hernandez- Carrillo y Beltran. (2017). Reliability Assessment of Rock Slope Stability by Dempster-Shafer Theory. National University of Colombia Bogotá campus Colombia.
- Hernandez-Carrillo R, Peñuela J y Beltran G. (2019). Characterization of Rock Masses by Three-Dimensional Image Processing Technique. Universidad Nacional de Colombia sede Bogotá. Geotechnical Engineering in the XXI Century: Lessons learned and future challenges.
- Hernandez- Carrillo y Beltran. (2019). Updating the probability of failure of rock wedges. Universidad Nacional de Colombia. Earth Sciences Research Journal.
- Hoek, E. (1998a). *Technical note Reliability of Hoek-Brown Estimates of Rock Mass Properties and their Impact on Design*.
- Hoek, E. (1998b). Technical note Reliability of Hoek-Brown Estimates of Rock Mass Properties and their Impact on Design EVERT HOEK INTRODUCTION. In *J. Rock Mech. Min. Scr* (Vol. 35, Issue I). Ekvier Saence Ltd.
- Hoek, E., & Bray, J. W. (1981). *ROCK SLOPES: Design , Excavation , Stabilization* (Issue September).
- Jimenez-Rodriguez, R., Sitar, N., & Chacón, J. (2006). System reliability approach to rock slope stability. *International Journal of Rock Mechanics and Mining Sciences*, 43(6), 847–859. <https://doi.org/10.1016/j.ijrmms.2005.11.011>
- Johari, A., & Mehrabani lari, A. (2016). System reliability analysis of rock wedge stability considering correlated failure modes using sequential compounding method. *International Journal of Rock Mechanics and Mining Sciences*, 82, 61–70.
<https://doi.org/10.1016/j.ijrmms.2015.12.002>
- Johari, A., & Mehrabani Lari, A. (2017a). System probabilistic model of rock slope stability considering correlated failure modes. *Computers and Geotechnics*, 81, 26–38.
<https://doi.org/10.1016/j.compgeo.2016.07.010>

- Johari, A., & Mehrabani Lari, A. (2017b). System probabilistic model of rock slope stability considering correlated failure modes. *Computers and Geotechnics*, 81, 26–38. <https://doi.org/10.1016/j.compgeo.2016.07.010>
- Jordá-Bordeshore, L., Jordá-Bordeshore, R., & Romero-Crespo, P. L. (2016). *Kinematic Assessment of Multi-Face Round Slopes Using Hemispherical Projection Methods (HPM)*.
- Kerguelen, A. (2009). *Análisis de la estabilidad de taludes mineros por métodos numéricos*. 1–155.
- Leung, C. F., & Quek, S. T. (1995). Probabilistic stability analysis of excavations in jointed rock. *Canadian Geotechnical Journal*, 32(3), 397–407. <https://doi.org/10.1139/t95-044>
- Lisle, R. J. (2004). Calculation of the daylight envelope for plane failure of rock slopes. *Geotechnique*, 54(4), 279–280. <https://doi.org/10.1680/geot.2004.54.4.279>
- Lisle, R. J. & Leyshon, P. R. (2004). *Stereographic techniques for geologists and civil engineers*, 2nd edn. Cambridge: Cambridge University Press.
- Low, B. K. (2008). Efficient probabilistic algorithm illustrated for a rock slope. *Rock Mechanics and Rock Engineering*, 41(5), 715–734. <https://doi.org/10.1007/s00603-007-0146-7>
- Marek, J. M., & Savely, P. (1978). Probabilistic analysis of the plane shear failure mode. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 16(1), A15. [https://doi.org/10.1016/0148-9062\(79\)90903-3](https://doi.org/10.1016/0148-9062(79)90903-3)
- Mat Radhi, M. S., Mohd Pauzi, N. I., & Omar, H. (2008). Probabilistic Approach of Rock Slope Stability Analysis Using Monte Carlo Simulation. *Iccbt*, 37, 449–468. https://www.researchgate.net/profile/Nur_Pauzi/publication/263351304_Probabilistic_Approach_of_Rock_Slope_Stability_using_Monte_Carlo_Simulation/links/00b7d53aa2e50e6793000000/Probabilistic-Approach-of-Rock-Slope-Stability-using-Monte-Carlo-Simulation.pdf
- Miller, S. M. (1983). *Probabilistic Analysis of Bench Stability For Use In Designing Open Pit Mine Slope*. 621–630.
- Miller, S. M. (1984). *Probabilistic rock slope engineering*.
- Müller, S. (1982). Ingenieurgeologie und Geomechanik als Grundlagen des Felsbaues / Engineering Geology and Geomechanics as Fundamentals of Rock Engineering. In *Ingenieurgeologie und Geomechanik als Grundlagen des Felsbaues / Engineering Geology and Geomechanics as Fundamentals of Rock Engineering*. <https://doi.org/10.1007/978-3-7091-8665-7>

- Nadim, F., Einstein, H., & Roberds, W. (2005). *State of the Art Paper 3 Probabilistic stability analysis for individual slopes in soil and rock*.
- Nilsen, B. (2000a). New trends in rock slope stability analyses. *Bulletin of Engineering Geology and the Environment*, 58(3), 173–178. <https://doi.org/10.1007/s100640050072>
- Nilsen, B. (2000b). New trends in rock slope stability analyses. In *Bull Eng Geol Env* (Vol. 58). Springer-Verlag.
- Obregon, C., & Mitri, H. (2019). Probabilistic approach for open pit bench slope stability analysis – A mine case study. *International Journal of Mining Science and Technology*, 29(4), 629–640. <https://doi.org/10.1016/j.ijmst.2019.06.017>
- Park, H., & West, T. R. (2000). Development of a probabilistic approach for rock wedge failure. *Engineering Geology*, 59(3–4), 233–251. [https://doi.org/10.1016/S0013-7952\(00\)00076-4](https://doi.org/10.1016/S0013-7952(00)00076-4)
- Park, H.-J. (2000). Probabilistic Approach of Stability Analysis for Rock Wedge Failure. *Economic and Environmental Geology*, 33(4), 295–307.
- Pathak, S., Poudel, R. K., & Kansakar, B. R. (2006). Application of Probabilistic Approach in Rock Slope Stability Analysis-AN Experience from Nepal. *Universal Academy Press*, 1(Tokyo, Japan), 797–802.
- Peñuela, J.L., Beltran-Calvo G., Hernandez-Carrillo, R. (2019). Adquisición y evaluación de datos geométricos de macizos rocosos a partir de imágenes tridimensionales para su uso en análisis geotécnicos. Universidad EAFIT. Ingeniería y Ciencia.
- Read & Stacey, (2009). Guidelines for Open Pit Slope Design. CSIRO. Australia. 221-235
- Robertson, a. M. (1970). The interpretation of geological factors for use in slope theory. *Planning Open Pit Mines, Proceedings, Johannesburg*, 55–71.
- Rosada, Omar (2021). Propuesta metodológica para el ajuste de una red de fracturas discretas (DFN) a partir de fotogrametría de corto alcance. Universidad Nacional de Colombia.
- Tang, W. H., Yucemen, M. S., & Ang, A. H.-S. (1976). Probability-based short term design of soil slopes. *Canadian Geotechnical Journal*, 13(3), 201–215. <https://doi.org/10.1139/t76-024>
- Tatone, B. S. A., & Grasselli, G. (2010). ROCKTOPPLE: A spreadsheet-based program for probabilistic block-toppling analysis. *Computers and Geosciences*, 36(1), 98–114. <https://doi.org/10.1016/j.cageo.2009.04.014>

Ya-Fen Lee; Yun-Yao Chi; C. Hsein Juang, F.ASCE; & Der-Her Lee (2012). Reliability Analysis of Rock Wedge Stability: Knowledge-Based Clustered Partitioning Approach. *Journal of geotechnical and geoenvironmental engineering*.

Zheng, J., Kulatilake, P. H. S. W., Shu, B., Sherizadeh, T., & Deng, J. (2014a). Probabilistic block theory analysis for a rock slope at an open pit mine in USA. *Computers and Geotechnics*, 61, 254–265.
<https://doi.org/10.1016/j.compgeo.2014.06.002>

Zheng, J., Kulatilake, P. H. S. W., Shu, B., Sherizadeh, T., & Deng, J. (2014b). Probabilistic block theory analysis for a rock slope at an open pit mine in USA. *Computers and Geotechnics*, 61, 254–265.
<https://doi.org/10.1016/j.compgeo.2014.06.002>

Anexo A. Análisis probabilístico de taludes en roca a través de los años

La metodología convencional empleada para el análisis de estabilidad de taludes en roca se realiza mediante determinaciones estadísticas, sin embargo la experiencia que ha dejado este tipo de análisis indica que hay errores que no pueden ser cuantificados, por lo que diversos investigadores han propuesto un tratamiento diferente a dichos análisis, con los que se busca identificar y cuantificar variables influyentes que de otra manera no podrían ser incluidos en los análisis de estabilidad convencional.

A continuación, se presenta de manera resumida, la evolución a través de los años que han tenido los análisis de estabilidad de taludes en roca involucrando la probabilidad.

1.1 El inicio de los métodos de análisis probabilísticos (1965 – 1979)

Arthur Casagrande fue uno de los primeros en articular formalmente la necesidad de equilibrar un posible riesgo con el costo mientras tomaba decisiones en la ingeniería geotécnica. Su concepto de "riesgo calculado" en la ingeniería geotécnica incorporó dos elementos, (Casagrande, 1965) señala:

1. El uso de conocimiento imperfecto, guiado por el juicio y la experiencia, para estimar los rangos probables de todas las cantidades pertinentes que se implementan en la solución de un problema.
2. La decisión sobre un margen de seguridad apropiado, o grado de riesgo, tomando en consideración el factor económico y la magnitud de las pérdidas que resultarían de la falla.

Pero fue a principios de los años setenta que la teoría de la probabilidad se introdujo por primera vez en la ingeniería geotécnica y teniendo sus raíces y el apoyo en la industria minera. En ese momento, la División de Mecánica de Suelos y Fundaciones de la Sociedad

Estadounidense de Ingenieros Civiles estableció un Comité de Conceptos de Confiabilidad y Probabilidad en el diseño en Ingeniería Geotécnica. En 1977, la *National Science Foundation* patrocinó un taller en el *Instituto Politécnico Rensselaer* sobre teoría de probabilidad y análisis de confiabilidad en ingeniería geotécnica.

Con los inicios de los métodos probabilísticos en los análisis de estabilidad de taludes en roca se empezaron a realizar estudios que presentaron una alternativa en el diseño de obras en los macizos rocosos.

Varios investigadores han reconocido la promesa de los enfoques probabilísticos y han formulado varios modelos los cuales se pueden agrupar en cuatro grandes categorías:

- Técnicas gráficas
- Simulaciones de Monte Carlo
- Modelos compuestos
- Modelos estocásticos

Dentro de las técnicas gráficas, McMahon, (1971, 1974, 1975) hizo una gran cantidad de trabajos en este campo. Su método es más adecuado para taludes secos, donde el deslizamiento se produce a lo largo de una sola discontinuidad.

McMahon, (1971) desarrolló un enfoque práctico a principios de los años setenta. Su modelo original tenía un rango de aplicabilidad relativamente limitado, pero introdujo la noción de usar la probabilidad de falla en lugar del factor de seguridad en los análisis de estabilidad de taludes rocosos. Propuso los procedimientos requeridos para la estabilidad cinemática y la estabilidad cinética para los métodos de análisis probabilísticos, fue probablemente el primero en proponer formalmente análisis probabilísticos de bloque - cuña.

Las técnicas de Monte Carlo se han aplicado a numerosos problemas en mecánica de rocas. Su et al. (1970), utilizó simulaciones de Monte Carlo junto con métodos de elementos finitos para evaluar distribuciones de esfuerzo alrededor de fallas subterráneas en estratos rocosos. Pariseau (1973), amplió su trabajo para investigar la estabilidad de los techos y pilares en las minas.

Major et al. (1978) desarrollaron un modelo probabilístico que se basa exclusivamente en la simulación de Monte Carlo. El núcleo de su modelo es un algoritmo determinista que

evalúa la estabilidad de las cuñas de roca como su Factor de Seguridad contra el deslizamiento. Los cálculos de estabilidad se basan en el modelo convencional que examina el movimiento traslacional de un cuerpo rígido. Procedimientos para la simulación de las propiedades de las fallas por Métodos de Monte Carlo se basan en el muestreo aleatorio de las distribuciones de probabilidad estimada para cada una de las propiedades señalan (Marek & Savely, 1978).

(Marek & Savely, 1978) plantean un análisis probabilístico para el mecanismo de falla planar por un plano de corte bidimensional de un talud, teniendo en cuenta la función de la resistencia al corte, orientación y longitud de la fractura en la formación de la falla potencial. Utilizan análisis de bloques rígidos para calcular la probabilidad de deslizamiento y la implementación del método de Monte Carlo para la obtención de las distribuciones de los factores de seguridad. Proponen la obtención de datos de campo de la longitud de las discontinuidades, espaciamiento, rugosidad, buzamiento y posteriormente una reducción de datos mediante distribuciones estadísticas según (Call y otros, 1977). El modelo de Marek y Savely (1978) examina fallas de corte plano bidimensional.

Call & Kim (1978) extendieron el método Marek-Savely para analizar taludes con numerosas discontinuidades, definen (PI), probabilidad de inestabilidad, como la probabilidad de que al menos una discontinuidad inestable se exponga. El modelo de Call-Kim depende en gran medida del supuesto de independencia. Trata la estabilidad de las discontinuidades como una serie de ensayos de Bernoulli, es decir, experimentos mutuamente independientes con una probabilidad constante de falla.

Call & Nicholas (1978) desarrollaron un modelo estocástico inductivo para investigar las superficies de falla escalonadas bidimensionales o por volteo. Este modo de falla ocurre en taludes que contienen dos conjuntos de diaclasas aproximadamente ortogonales.

Call & Nicholas (1978) & (Glynn, 1979) proponen métodos de análisis de estabilidad 2-D para taludes que utilizan información estadística sobre la discontinuidad y que permiten análisis de falla tanto planar como en volteo. Las características de la falla del talud como la orientación, el espaciamiento, la longitud y la rugosidad del macizo rocoso son variables aleatorias que pueden ser modeladas por distribuciones estadísticas estimadas a partir de datos cartográficos según (Call y otros, 1976).

Baecher y otros, (1977), presentan un resumen sobre la estimación de la PDF (Probability Density Function) para el espaciamiento y longitud de las discontinuidades, desarrollaron un modelo estadístico deductivo para las discontinuidades en rocas. Su objetivo era formular un modelo conceptual que pudiera explicar dos observaciones empíricas como son:

1. Las longitudes de las discontinuidades están log normalmente distribuidas.
2. Los espaciamientos de las discontinuidades están distribuidos exponencialmente.

El modelo Veneziano (1978) es similar al propuesto por Baecher y otros (1977). Este último modelo trata las discontinuidades como fenómenos aislados en el que cada plano contiene una sola articulación.

(Glynn, 1979) da a conocer algunas herramientas para usar en el análisis de la estabilidad de taludes en roca desde un punto de vista probabilístico. Presenta un procedimiento que permite utilizar las PDF de la orientación de las discontinuidades para derivar a las PDF de parámetros que caracterizan la forma y la orientación de las posibles cuñas de roca delimitadas por las discontinuidades. Combina los modelos determinísticos con la simulación de Monte Carlo para que algunos de los parámetros de entrada se puedan tratar como variables aleatorias.

1.2 Avances entre los años 1980 y 1989

(Athanasίου-Grivas, 1980), propone un análisis probabilístico de la estabilidad sísmica para un talud natural mediante un modelo previamente desarrollado, examina la dependencia de parámetros sísmicos en la probabilidad de falla de los taludes. La seguridad del talud se mide en términos de su probabilidad de falla (en lugar del factor de seguridad convencional), cuyos valores numéricos se obtienen a través de una simulación Monte Carlo de falla.

En el trigésimo Coloquio de Geomecánica de la Sociedad Austríaca de Geomecánica, (Einstein & Baecher, 1983) presentan una introducción a los Métodos Probabilísticos y Estadísticos en Ingeniería Geológica en donde señalan que *la incertidumbre sobre las condiciones geológicas y sobre los parámetros geotécnicos es probablemente la característica más distintiva de la ingeniería geotécnica y la geología*. Se analizan los enfoques tradicionales para estudiar la incertidumbre y se señalan sus deficiencias. También se da una descripción muy breve de la teoría de la probabilidad y la estadística.

Herget (1982), en este mismo coloquio, presenta un artículo sobre el Diseño probabilístico de taludes para minas a cielo abierto, señalando que el mejor enfoque para dicho diseño es determinar la probabilidad de que un talud sea estable mediante la técnica de muestreo de Monte Carlo, el cual se utiliza para obtener un valor muestreado de la población de cada una de las variables. Estos valores muestreados se usan para obtener un solo factor de seguridad de la manera convencional señala Herget (1982).

(Einstein et al., 1983) indican el efecto de la persistencia de las discontinuidades en la estabilidad de los taludes de roca, pues la persistencia de la discontinuidad tiene un efecto mayor en la resistencia de la masa de roca, pero como no es posible cartografiar internamente las discontinuidades de un macizo rocoso, la persistencia es un parámetro difícil de medir. Ellos desarrollan un método para relacionar la estabilidad del macizo rocoso, la persistencia con la geometría y la variabilidad espacial de las discontinuidades.

(Miller, 1983) recomienda un análisis probabilístico de la estabilidad de taludes para el uso en el diseño de minas a cielo abierto, en donde las inestabilidades del banco suelen influir en las geometrías generales de los taludes de la mina. Los posibles modos de falla se identifican evaluando cómo interactúan las orientaciones estructurales mapeadas en un sector dado. Los datos cartografiados también se utilizan para estimar distribuciones estadísticas y correlaciones espaciales de los conjuntos de discontinuidades que comprometen los posibles modos de falla. La probabilidad de falla para varios mecanismos de falla de taludes se suele estimar mediante técnicas de simulación de Monte Carlo que se basan en el muestreo repetido de las distribuciones de probabilidad de los parámetros de entrada.

(Miller, 1984) en su publicación *Probabilistic Rock Slope Engineering*, define que el propósito de un análisis probabilístico de la estabilidad de los taludes en roca es proporcionar una estimación de la estabilidad del talud sobre la base de un tratamiento realista e incorporación de variabilidades naturales, incertidumbres de medición, y múltiples caminos potenciales de falla. Presenta un recopilado de temas relacionados con el análisis de la estabilidad de los taludes en roca desde un enfoque probabilístico, los factores que influyen en la estabilidad de los taludes, un breve resumen del surgimiento de la probabilidad en la ingeniería de taludes, procedimientos probabilísticos de la ingeniería de taludes, mapeo de las discontinuidades, análisis estadístico de los datos de las

discontinuidades, análisis de los parámetros de resistencia de la roca, análisis probabilístico de estabilidad para mecanismos de falla común y procedimientos de diseño probabilístico de taludes.

(Bolle, 1987), señala que el enfoque probabilístico permite tener en cuenta parámetros con una variación considerable, así como acciones aleatorias como terremotos. La técnica de cálculo utilizada fue el método multivariado de estimación puntual de Rosenblueth. El cual permite el uso de distintas variables aleatorias correlacionadas dadas por sus dos o tres primeros momentos estadísticos (media, desviación estándar, asimetría) con cualquier método de diseño. El enfoque probabilístico de los problemas de ingeniería civil por el método de estimación puntual de Rosenblueth da la posibilidad de analizar cualquier tipo de estructura incluyendo la variabilidad y la correlación de las variables, así como las sollicitaciones aleatorias.

1.3 Propuestas presentadas entre los años 1990 y 1999

Para este periodo, (Scavia et al., 1990), realizan un análisis de equilibrio límite en 2D de bloques de roca ubicados sobre una superficie de falla escalonada y se lleva a cabo incluyendo tanto un procedimiento de simulación de Monte Carlo como la Teoría de Cadenas de Markov.

(Genske & Walz, 1991) evalúa la estabilidad de taludes en roca de forma probabilística, señala que la mayoría de los parámetros de la función de desempeño deben considerarse estadísticamente dispersivos. Por lo tanto, antes de un análisis de confiabilidad, se debe discutir las estadísticas de los parámetros. Posteriormente, lleva a cabo estudios de sensibilidad probabilística, utilizando el enfoque de primer orden segundo momento FOSM.

(Fenton & Vanmarcke, 1991) presentan un método rápido y preciso de la ejecución de un proceso aleatorio escalar Gaussiano homogéneo en una, dos, o tres dimensiones, definido por su función media y de covarianza. El concepto detrás del enfoque de subdivisión media local (LAS) surgió del algoritmo de subdivisión estocástico descrito por Carpenter (1980) y Founder y otros (1982). En el enfoque LAS, la consistencia interna implica que ciertas regiones mantienen un promedio constante a lo largo de la subdivisión.

(Carter & Lajtai, 1991) en su publicación sobre estabilidad de taludes en roca y sistemas de discontinuidades, señalan que la probabilidad de falla se expresa como la proporción de cuñas cinemáticamente libres que tienen un factor de seguridad menor que la unidad con respecto al número total de cuñas. Introdúcen un código determinista (GEOSLIDE) y un código de microordenador probabilístico (PROSLIDE) para ayudar a realizar análisis de cuña de roca basados en el método de equilibrio límite.

(Muralha, 1991) presenta un enfoque probabilístico de la estabilidad de los taludes rocosos, señalando que las herramientas y técnicas que se utilizan comúnmente para diseñar taludes rocosos (analíticas o gráficas y de dos o tres dimensiones) generalmente no tienen en cuenta la variabilidad de los parámetros involucrados. Presenta un enfoque probabilístico del análisis de estabilidad de los taludes de las rocas basado en los datos que usualmente están disponibles para estos problemas (características geométricas y mecánicas de las discontinuidades).

Low y Einstein, (1992) presentaron un análisis más completo de las fallas en cuña, incluyendo una geometría del talud más compleja y utilizando el método de primer orden segundo momento FOSM.

(Leung & Quek, 1995), señalan que la orientación de las discontinuidades es uno de los principales parámetros de entrada en el análisis de estabilidad determinista convencional de los bloques de roca. En este trabajo se propone un enfoque probabilístico para incorporar la constante de Fisher en el análisis de estabilidad de bloques de roca.

(Stevens, 1998), presenta el desarrollo de un programa denominado RocFall, como una herramienta para el análisis probabilístico de diseño de medidas correctivas y predicción de la caída de rocas. El empleo de probabilidad y estadística en el análisis de simulaciones de caída de rocas ha demostrado ser un método eficaz y aceptable para hacer frente a estas dificultades.

(Hoek, 1998), argumentan que, en lugar de basar una decisión de diseño de ingeniería en un solo factor de seguridad calculado, una evaluación más racional de los riesgos asociados con un diseño particular se presenta al llevar a cabo un estudio de sensibilidad. Esto implica una serie de cálculos en la que cada parámetro significativo se varía sistemáticamente a lo

largo de su rango máximo creíble a fin de determinar su influencia sobre el factor de seguridad

(Sjöberg, 1999) indica que, en un método de diseño probabilístico se incluye la naturaleza estocástica de los parámetros de entrada y se calcula la probabilidad de falla resultante. En los casos en que la función de rendimiento es compleja y contiene un gran número de variables, se puede utilizar una técnica de simulación como el método de Monte Carlo. A partir de cada distribución, se muestrea aleatoriamente un valor de parámetro y se calcula el valor de la función de rendimiento para cada conjunto de muestras aleatorias.

(Nilsen, 2000) analiza los enfoques alternativos para la estabilidad incluido el principio determinista tradicional, el llamado principio de factor parcial y el enfoque probabilístico que evalúa la probabilidad de fallo y no el factor de seguridad.

1.4 Aportes entre 2000 y 2009

(H. Park & West, 2000) desarrollan un enfoque probabilístico para la falla en cuña en roca, señalando que, para la ingeniería de taludes en roca, la incertidumbre y la variabilidad son inherentes a los datos recopilados sobre la orientación y la intensidad de las discontinuidades, obteniéndose una gama de resultados. Como parte de la solución, la probabilidad de inestabilidad cinemática y la probabilidad de inestabilidad cinética se evalúan por separado para proporcionar una evaluación adecuada y combinada para la probabilidad de falla. La probabilidad de falla se define simplemente como la probabilidad de tener $FS \leq 1$ dado como un porcentaje igual a la zona bajo la función de densidad de probabilidad (PDF) para FS que sea $FS \leq 1$. Usa el método de primer orden segundo momento (FOSM) para calcular la probabilidad de falla de manera aproximada.

(Gokceoglu et al., 2000) examinaron la posibilidad de fallas de taludes, controlados por las discontinuidades de andesitas fracturadas en la región de asentamientos de Altindag (Ankara) y preparan mapas probabilísticos de riesgo de falla planar, volteo y falla por cuña usando las reglas cinemáticas y modelo de elevación digital de la zona de estudio.

(Duzgun et al., 2002), proponen un modelo probabilístico para la evaluación de las incertidumbres de la resistencia al corte por discontinuidades en roca. En este estudio se examinan minuciosamente las incertidumbres que subyacen en la resistencia al corte por

discontinuidades y se desarrolla un modelo de análisis de incertidumbre para la estimación de la resistencia al corte de la discontinuidad in situ con un énfasis especial en taludes en roca. Consideran las principales causas de discrepancias, a saber, la escala, la anisotropía y la saturación de agua. Presentan un modelo con un procedimiento muy simple para la cuantificación de los sesgos en la toma de datos y los diferentes tipos de incertidumbres implicadas en el procedimiento de estimación del valor in situ del ángulo de fricción máximo con los resultados de las pruebas de laboratorio. Factores de corrección aleatorios se utilizan para dar cuenta de cualquier fuente de incertidumbre y error, y el efecto de varios tipos de incertidumbres y errores se combinan de acuerdo con el formato de análisis de la incertidumbre de primer orden.

(Genevois & Romeo, 2003), presentan una evaluación probabilística global de las condiciones de estabilidad de los taludes de roca, basándose en el análisis de la variabilidad espacial de los aspectos estructurales del macizo rocoso, de la incertidumbre de las propiedades geomecánicas y de las probabilidades de ocurrencia de ciertos valores de fuerzas sísmicas e intensidades de lluvia.

(Nadim et al., 2005), identifican varios métodos de confiabilidad bien establecidos, tales como la aproximación de primer orden segundo momento (FOSM), el método de confiabilidad de primer orden (FORM) y simulación Monte Carlo como técnicas útiles para determinar la confiabilidad de los taludes y para la estimación de la probabilidad de falla. El análisis probabilístico planteado por los autores se basa en la recopilación de datos como la geometría de la discontinuidad, la resistencia a la discontinuidad, la geometría del talud, la presión del agua, la persistencia de la discontinuidad, etc. y las características mecánicas como ángulo de fricción, cohesión, la resistencia de la roca intacta.

(H. J. Park et al., 2005), describen un análisis probabilístico de la estabilidad de taludes de roca y propiedades aleatorias de los parámetros de discontinuidad. En este estudio un procedimiento de análisis probabilístico y los algoritmos relacionados se desarrollaron mediante la extensión de la simulación Monte Carlo. Se presenta un enfoque probabilístico el cual consta de dos partes: análisis de datos geotécnicos disponibles para obtener propiedades aleatorias de los parámetros de las discontinuidades; el análisis probabilístico de la estabilidad de taludes en función de parámetros con propiedades aleatorias y análisis por el método de la inferencia estadística.

(Jimenez-Rodriguez et al., 2006), dan a conocer una metodología cuantitativa y sistemática para el análisis de confiabilidad de la estabilidad de taludes en roca. Los métodos de estimativos puntuales y el método de primer orden segundo momento (FOSM) se aplican a los problemas de estabilidad de taludes. El método de simulación de Monte Carlo se utiliza para obtener una solución "exacta", a expensas de un costo computacional más alto, mientras que los métodos basados en aproximaciones de primer orden resultan ser computacionalmente eficientes y proporcionar información de interés para el proceso de diseño.

(Pathak et al., 2006) hacen notar que las técnicas de análisis probabilístico, y en particular la técnica de Hypercubo latino, proporcionan una metodología para evaluar la estabilidad de taludes que aporta más información útil en comparación con el valor del factor de seguridad. Este método busca utilizar la teoría de la probabilidad para proporcionar un procedimiento sistemático para tratar las incertidumbres cuantitativamente. En un enfoque probabilístico, se hace la determinación de una distribución de probabilidad apropiada de los parámetros variables de entrada que se utilizarán en el modelo.

(H. J. Park et al., 2006) deducen una evaluación de la probabilidad de falla de la cuña de roca utilizando el método de estimativos puntuales y el método de estimación de máxima verosimilitud, se propuso el método de aproximación para el análisis de confiabilidad. El método de aproximación tal como el método de primer orden segundo momento (FOSM) y el método de estimativos puntuales (PEM) sólo requiere la media y la desviación estándar de la variable aleatoria y, por lo tanto, es fácil de utilizar cuando la información es limitada. Para superar la limitación, se utiliza la ecuación simple, obtenida del método de estimación de máxima verosimilitud para la falla de cuña para calcular la probabilidad de falla.

(Hammah, Yacoub, & Curran, 2009) describen una modelación numérica de la incertidumbre de taludes debido a las juntas en los macizos rocosos, este trabajo tiene como objetivo evaluar el impacto que la aleatoriedad de la geometría que la red de discontinuidades tiene sobre la estabilidad de los taludes en roca. Las técnicas probabilísticas consideradas son el método de estimativos puntuales (PEM), la superficie de respuesta (RSM), métodos de confiabilidad como FOSM, los métodos de Monte Carlo y

el método Hypercubo latino. La intención del análisis probabilístico es calcular los momentos estadísticos de la distribución de los factores de seguridad.

(Mat Radhi et al., 2008) presenta un enfoque probabilístico de análisis de estabilidad en taludes en roca usando la simulación de Monte Carlo basado en parámetros de las discontinuidades. El análisis probabilístico se realizó mediante el análisis cinemático y cinético. El análisis cinemático se estableció en el análisis de la proyección estereográfica y el análisis cinético se basó en el análisis determinista. Los datos de entrada para del análisis cinemático son la inclinación y orientación, ángulo de fricción, cohesión, y ángulo de inclinación del talud. Los datos de salidas son del tipo de falla del talud si es planar, cuña, volteo o combinación de dos tipos de falla. Los datos de entrada para el análisis cinético son propiedades del talud, la cohesión, ángulo de fricción, y nivel freático.

(Hammah, Yacoub, Curran, et al., 2009), aplicaron el método de elementos finitos (FEM) y el análisis de Reducción de Resistencia al corte (SSR) para calcular las probabilidades de falla en taludes, utilizando dos aproximaciones probabilísticas: el Método de estimativos puntuales (PEM) y un número limitado de simulaciones de Monte Carlo. El objetivo final del análisis probabilístico de la estabilidad de taludes es obtener la distribución completa de valores del factor de seguridad dado un conjunto de variables de entrada aleatorias (inciertas) con propiedades estadísticas específicas y determinando momentos estadísticos de la distribución de salida.

1.5 Planteamientos entre los años 2010 y 2019

(Tatone & Grasselli, 2010) proponen el programa ROCKTOPPLE: programa de hoja de cálculo para análisis probabilístico de volteo de bloques, un procedimiento de simulación de Monte Carlo para el análisis probabilístico de volteo de bloques. El procedimiento de análisis considera ambas probabilidades de falla, cinemática y cinética. Estas probabilidades se evalúan por separado y se multiplicaron para dar la probabilidad total del volteo de bloques.

(Naghadehi et al., 2011), sugieren una metodología de sistemas probabilísticos para analizar la importancia de los factores que afectan la estabilidad de taludes en roca. Una metodología de codificación probabilística semi-cuantitativa (PESQ) se emplea para evaluar la importancia de los factores que afectan la estabilidad de taludes en roca dentro

de los sistemas de ingeniería de rocas (RES). Con este marco de codificación PESQ, las incertidumbres en las asignaciones de códigos se expresan utilizando probabilidades que se asignan a cada valor de codificación particular.

(Li et al., 2011), propone un método cuantitativo sistemático para la evaluación de la confiabilidad del sistema de taludes de la roca con falla planar que implique varios modos de falla correlacionados. A través de un árbol de fallas probabilístico, se pueden definir todos los modos de falla, y las correlaciones entre diferentes modos de falla que pueden tomarse en consideración de una manera más racional.

(Gibson, 2011), compara métodos probabilísticos diferentes (FOSM, PEM y MC) usados en el análisis de estabilidad y presenta las ecuaciones requeridas para usar un Método de Estimación puntual Modificado (mPEM) presentado por Harr, (1989). Los métodos se comparan usando ejemplos simples. Las probabilidades de falla recomendadas para el diseño a cielo abierto también se presentan.

(Lee et al., 2011), presentan un análisis de confiabilidad de la estabilidad de cuñas para taludes en roca. Desarrollan una técnica de partición agrupada basada en el conocimiento para determinar el índice de confiabilidad y la probabilidad de falla de una cuña en roca. Analizan el índice de confiabilidad y realizan la optimización utilizando una técnica de particionamiento en racimos (KCP) basada en el conocimiento. El índice de confiabilidad calculado con esta técnica se compara con los métodos que utilizan otros enfoques como el método basado en Excel Solver.

(Cruz, 2012) propone en su trabajo un análisis probabilístico de fallas superficiales para taludes debido a procesos de infiltración; para definir la metodología, se aplicaron los métodos probabilísticos de Primer Orden Segundo Momento, Estimativos Puntuales, Hasofer-Lind y Simulación de Monte Carlo a dos modelos de análisis superficial de taludes en un caso piloto. Primero determino las variables aleatorias que controlan la estabilidad superficial y sus modelos de probabilidad.

(Azizi et al., 2013) desarrollaron un modelo físico de prueba de estabilidad de taludes en Indonesia en el laboratorio de Geomecánica y Equipo de Minería del Departamento de Ingeniería de Minas de ITB Indonesia, con el cual se han realizado modelos físicos de

análisis de fallas de taludes utilizando material de arenisca. Se enfatiza en el efecto de la vibración, la altura y la pendiente del talud, en el análisis de la probabilidad de falla con el uso del método de Monte Carlo.

(Gravanis et al., 2014), propone una solución analítica en la evaluación probabilística de estabilidad de taludes de roca basado en campos aleatorios; tanto la cohesión y el ángulo de fricción a lo largo de una discontinuidad y se tratan como parámetros gaussianos aleatorios, campos que están completamente descritos por sus valores medios.

(Admassu & Shakoore, 2013) plantean DIPANALYST: Un programa informático para el análisis cinemático cuantitativo de fallas de taludes en roca, mediante el cual se analiza el potencial de los diversos modos de falla estructuralmente controlados (planar, cuña, y volteo) debido a la presencia de discontinuidades orientadas desfavorablemente. Usa la distribución de la orientación más común, la distribución circular de Fisher, que es el más cercano a la distribución normal (Borradaile, 2003). La probabilidad de inestabilidad cinemática es calculada como la relación entre el número de iteraciones que son cinemáticamente inestables con respecto al número total de iteraciones.

(Wang et al., 2013) presentan un retro análisis probabilístico de una falla del talud en un sitio en vías en el norte de Taiwán. Investigaciones posteriores al evento de esta falla encontraron desaciertos en los parámetros y sistemas de anclaje como unas de las causas más probables para la falla. Para dar cuenta de las incertidumbres en los parámetros de entrada, se adoptó el enfoque del análisis probabilístico, el método de Monte Carlo, la cadena de Markov se utilizó para respaldar y calcular los parámetros de resistencia geotécnicos y la fuerza de anclaje.

(Zheng et al., 2014), presentan un análisis probabilístico de la teoría de bloque para un talud de roca en una mina a cielo abierto en EE.UU., desarrollan un código de computadora (PBTAC) *Probabilistic Block Theory Analysis Code* para realizar análisis de teoría de bloques tanto determinísticos como probabilísticos. La variabilidad de la orientación de las discontinuidades y la resistencia al corte se incorpora en el análisis probabilístico de la teoría de bloques. La orientación de discontinuidades se trata como una variable aleatoria bivariada que incluye la correlación que existe entre el ángulo de inclinación y su orientación. Las distribuciones Beta y Fisher se han usado para representar la variabilidad

del ángulo de inclinación y la orientación de las discontinuidades. El ángulo de inclinación y la orientación han sido tomadas como parámetros no correlacionados.

(Johari & Mehrabani Iari, 2016) presentan un modelo probabilístico del sistema de estabilidad de taludes en roca considerando modos de falla correlacionados. El modelo involucra múltiples modos de falla correlacionadas, considerando todos los parámetros de entrada como parámetros estocásticos. Para este propósito, se emplea el Método de Composición Secuencial (SCM) como un método poderoso para reducir el tiempo computacional y la evaluación precisa para determinar los índices de confiabilidad al considerar correlaciones entre modos de falla que se calculan definiendo un margen de seguridad lineal equivalente para cada modo de falla.

(Niño, 2017) planteó en su trabajo una metodología de análisis, combinando aproximaciones basadas en elementos discretos con conceptos de confiabilidad y probabilidad. Los métodos de elementos discretos le permitieron considerar la naturaleza discontinua de un macizo rocoso modelado como un sistema de bloques; este método tiene en cuenta los cambios que sufren los contactos mientras los bloques interactúan ante alguna sollicitación. Seleccionó el análisis de deformaciones discontinuas (DDA), basado en el cálculo de desplazamientos a partir de ecuaciones de equilibrio.

Hernandez- Carrillo (2017) presenta una Evaluación de la confiabilidad de la estabilidad de taludes en roca mediante la teoría de Dampster-Shafer, tiene en cuenta la limitada información, la incertidumbre de la veracidad de los datos que se recolectan a través del tiempo que van alimentando la base de datos para el análisis de estabilidad de los taludes y que representan cambios en los resultados con la nueva información que se va obteniendo. Con el uso de la teoría de Dampster y la combina las propuestas de Yanger para no eliminar los datos contradictorios, sino que los asigne a un conjunto universal. Representa cada una de las funciones de probabilidad acumulada de la orientación y ángulo de fricción de los diferentes conjuntos de datos recopilados en diferentes años, en dichas graficas se identifica los intervalos en donde la información presenta conflicto y la que es consistente usando el concepto DS y RST o teoría de conjuntos aleatorios.

Obregon & Mitri, (2019), muestran una evaluación de estabilidad de taludes para una mina a cielo abierto en Perú usando enfoque probabilístico, donde combinan un análisis cinemático basado en técnicas de proyección estereográfica y luego un análisis cinético

mediante el método del equilibrio límite. Finalmente, estas dos probabilidades se combinan para proporcionar una medida general de la probabilidad de falla (PoF) del sistema de talud. Realizan un mapeo extenso de la superficie en 36 sitios siguiendo los procedimientos sugeridos por ISRM. Además, llevaron a cabo varios ensayos de corte directo y demostraron que, al combinar mediciones de campo, laboratorio y juicio de ingeniería, las funciones de densidad de probabilidad (PDF) de los parámetros de las discontinuidades pueden ser obtenidas. Luego utilizan simulaciones de Monte Carlo para calcular tanto la probabilidad cinemática como la probabilidad cinética de falla.

(Hernandez-Carrillo & Beltrán, 2019), presentan un análisis de confiabilidad estabilidad de cuñas en una mina de arenisca, localizada en Une, Cundinamarca, donde se ha recolectado información durante 20 años. Este análisis permitió actualizar la probabilidad de falla. Como resultado se obtuvo una probabilidad de falla en un rango entre 0.04 y 0.25. Adicional se presenta la teoría de falla de roca por mecanismo de cuña bajo la teoría de los conjuntos aleatorios. La evaluación de la confiabilidad se presenta por medio de curvas de probabilidad que a su vez permiten ser actualizadas a través de la función de densidad acumulativa acotada de cada variable y del factor de seguridad, el cual es el resultado de la metodología aplicada para calcular y actualizar la probabilidad de falla.

Hernández- Carrillo y Beltrán (2019) presentan una propuesta para el uso de la teoría de conjuntos aleatorios para el combinar diferentes fuentes de información y efectuar análisis de confiabilidad cuando la información es escasa, y a medida que se obtenga nueva información poder actualizar la evaluación de probabilidad de falla para el análisis de confiabilidad de la estabilidad de cuñas de una mina en Une, Cundinamarca.

Hernández – Carrillo, Peñuela y Beltrán (2019) presenta una alternativa a la medición de características estructurales en macizos rocosos usando imágenes estereoscópicas de la cara del talud, usando programas computaciones de referenciación con el objetivo de generar imágenes 3D con la cual se pueda medir la orientación y geometría de las discontinuidades, procesar estos datos y representar los polos en la red estereográfica, permitiendo la agrupación y definición de familias de discontinuidades. Esta propuesta representa grandes aportes a la geotecnia de rocas en la recolección masiva de datos estructurales y aplicación de metodologías probabilísticas para la definición de las familias de discontinuidades representativas del macizo rocoso y demás parámetros geométricos.

Rosada (2021), presenta en su tesis de grado una Propuesta metodológica para el ajuste de una red de fracturas discretas (DFN) a partir de fotogrametría de corto alcance, en donde analiza la susceptibilidad de un modelo DFN ante la presencia de datos que direccionalmente se clasifican como discordantes, y mediante un código de programación estudia los datos mediante criterios estadísticos.

Citaciones

- Admassu, Y., & Shakoor, A. (2013). DIPANALYST: A computer program for quantitative kinematic analysis of rock slope failures. *Computers and Geosciences*, 54, 196–202. <https://doi.org/10.1016/j.cageo.2012.11.018>
- Athanasiou-Grivas, D. (1980). Probabilistic seismic stability analysis - a case study. *Canadian Geotechnical Journal*, 17(3), 352–360. <https://doi.org/10.1139/t80-042>
- Azizi, M. A., Kramadibrata, S., Wattimena, R. K., & Sidi, I. D. (2013). Probabilistic Analysis of Physical Models Slope Failure. *Procedia Earth and Planetary Science*, 6, 411–418. <https://doi.org/10.1016/j.proeps.2013.01.054>
- Bolle, A. (1987). *A probabilistic approach of slope stability in fractured rock*. 301–303.
- Carter, B. J., & Lajtai, E. Z. (1991). Rock slope stability and distributed joint systems. *Canadian Geotechnical Journal*, 29(1), 53–60. <https://doi.org/10.1139/t92-006>
- Casagrande, A. (1965). Role of the Calculated Risk in Earthwork and Foundation Engineering. *Journal of the Soil Mechanics and Foundations Division*, 91(4), 1–40.
- Cruz, L. (2012). *Análisis probabilístico de fallas superficiales en taludes debido a procesos de infiltración*.
- Duzgun, H. S. B., Yucemen, M. S., & Karpuz, C. (2002). A probabilistic model for the assessment of uncertainties in the shear strength of rock discontinuities. *International Journal of Rock Mechanics and Mining Sciences*, 39(6), 743–754. [https://doi.org/10.1016/S1365-1609\(02\)00050-3](https://doi.org/10.1016/S1365-1609(02)00050-3)
- Einstein, H. H., & Baecher, G. B. (1983). Probabilistic and statistical methods in engineering geology - Specific methods and examples part I: Exploration. *Rock Mechanics and Rock Engineering*, 16(1), 39–72. <https://doi.org/10.1007/BF01030217>
- Einstein, H. H., Veneziano, D., Baecher, G. B., & O'Reilly, K. J. (1983). The effect of discontinuity persistence on rock slope stability. *International Journal of Rock Mechanics and Mining Sciences And*, 20(5), 227–236. [https://doi.org/10.1016/0148-9062\(83\)90003-7](https://doi.org/10.1016/0148-9062(83)90003-7)

- Fenton, G. A., & Vanmarcke, E. H. (1991). *SIMULATION OF RANDOM FIELDS VIA LOCAL AVERAGE SUBDIVISION*. 116(8), 1733–1749.
- Genevois, R., & Romeo, R. W. (2003). Probability of Failure Occurrence and Recurrence in Rock Slopes Stability Analysis. *International Journal of Geomechanics*, 3(1), 34–42. [https://doi.org/10.1061/\(asce\)1532-3641\(2003\)3:1\(34\)](https://doi.org/10.1061/(asce)1532-3641(2003)3:1(34))
- Genske, D. D., & Walz, B. (1991). Probabilistic assessment of the stability of rock slopes. *Structural Safety*, 9(3), 179–195. [https://doi.org/10.1016/0167-4730\(91\)90042-8](https://doi.org/10.1016/0167-4730(91)90042-8)
- Gibson, W. (2011). Probabilistic methods for slope analysis and design. *Australian Geomechanics Journal*, 46(3), 1–12.
- Glynn, E. F. (1979). *A probabilistic approach to the stability of rock slopes*.
- Gokceoglu, C., Sonmez, H., & Ercanoglu, M. (2000). Discontinuity controlled probabilistic slope failure risk maps of the Altindag (settlement) region in Turkey. *Engineering Geology*, 55(4), 277–296. [https://doi.org/10.1016/S0013-7952\(99\)00083-6](https://doi.org/10.1016/S0013-7952(99)00083-6)
- Gravanis, E., Pantelidis, L., & Griffiths, D. V. (2014). An analytical solution in probabilistic rock slope stability assessment based on random fields. *International Journal of Rock Mechanics and Mining Sciences*, 71, 19–24. <https://doi.org/10.1016/j.ijrmms.2014.06.018>
- Hammah, R. E., Yacoub, T. E., & Curran, J. H. (2009). Numerical modelling of slope uncertainty due to rock mass jointing. *International Conference on Rock Joints and Jointed Rock Masses*, 1–8.
- Hammah, R. E., Yacoub, T. E., Curran, J. K., & . (2009). Probabilistic Slope Analysis with the Finite Element Method. *American Rock Mechanics Association*, 09, 1–19.
- Hernandez-Carrillo, R., & Beltrán, G. (2019). Updating the probability of failure of rock wedges. *Earth Sciences Research Journal*, 23(3), 225–236. <https://doi.org/10.15446/esrj.v23n3.74779>
- Hoek, E. (1998). *Technical note Reliability of Hoek-Brown Estimates of Rock Mass Properties and their Impact on Design*.
- Jimenez-Rodriguez, R., Sitar, N., & Chacón, J. (2006). System reliability approach to rock slope stability. *International Journal of Rock Mechanics and Mining Sciences*, 43(6), 847–859. <https://doi.org/10.1016/j.ijrmms.2005.11.011>
- Johari, A., & Mehrabani Iari, A. (2016). System reliability analysis of rock wedge stability considering correlated failure modes using sequential compounding method. *International Journal of Rock Mechanics and Mining Sciences*, 82, 61–70. <https://doi.org/10.1016/j.ijrmms.2015.12.002>

- Lee, Y.-F., Chi, Y.-Y., Juang, C. H., & Lee, D.-H. (2011). Reliability Analysis of Rock Wedge Stability: Knowledge-Based Clustered Partitioning Approach. *Journal of Geotechnical and Geoenvironmental Engineering*, 138(6), 700–708. [https://doi.org/10.1061/\(asce\)gt.1943-5606.0000618](https://doi.org/10.1061/(asce)gt.1943-5606.0000618)
- Leung, C. F., & Quek, S. T. (1995). Probabilistic stability analysis of excavations in jointed rock. *Canadian Geotechnical Journal*, 32(3), 397–407. <https://doi.org/10.1139/t95-044>
- Li, D. Q., Jiang, S. H., Chen, Y. F., & Zhou, C. B. (2011). System reliability analysis of rock slope stability involving correlated failure modes. *KSCE Journal of Civil Engineering*, 15(8), 1349–1359. <https://doi.org/10.1007/s12205-011-1250-5>
- Marek, J. M., & Savely, P. (1978). Probabilistic analysis of the plane shear failure mode. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 16(1), A15. [https://doi.org/10.1016/0148-9062\(79\)90903-3](https://doi.org/10.1016/0148-9062(79)90903-3)
- Mat Radhi, M. S., Mohd Pauzi, N. I., & Omar, H. (2008). Probabilistic Approach of Rock Slope Stability Analysis Using Monte Carlo Simulation. *Iccbt*, 37, 449–468. https://www.researchgate.net/profile/Nur_Pauzi/publication/263351304_Probabilistic_Approach_of_Rock_Slope_Stability_using_Monte_Carlo_Simulation/links/00b7d53aa2e50e6793000000/Probabilistic-Approach-of-Rock-Slope-Stability-using-Monte-Carlo-Simulation.pdf
- Miller, S. M. (1983). *Probabilistic Analysis of Bench Stability For Use In Designing Open Pit Mine Slope*. 621–630.
- Miller, S. M. (1984). *Probabilistic rock slope engineering*.
- Muralha, J. (1991). Probabilistic approach to the stability of rock slopes. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, 30(3), A192. [https://doi.org/10.1016/0148-9062\(93\)93143-1](https://doi.org/10.1016/0148-9062(93)93143-1)
- Nadim, F., Einstein, H., & Roberds, W. (2005). *State of the Art Paper 3 Probabilistic stability analysis for individual slopes in soil and rock*.
- Naghadehi, M. Z., Jimenez, R., KhaloKakaie, R., & Jalali, S. M. E. (2011). A probabilistic systems methodology to analyze the importance of factors affecting the stability of rock slopes. *Engineering Geology*, 118(3–4), 82–92. <https://doi.org/10.1016/j.enggeo.2011.01.003>
- Nilsen, B. (2000). New trends in rock slope stability analyses. *Bulletin of Engineering Geology and the Environment*, 58(3), 173–178. <https://doi.org/10.1007/s100640050072>
- Niño, J. (2017). *Análisis Probabilístico para el Diseño de Taludes en Explotación Minera a Cielo Abierto*.

- Park, H. J., Jeong, U. J., Han, B., Ro, R., Shing, K., & Kim, J. (2006). The evaluation of the probability of rock wedge failure using the point estimate method and maximum likelihood estimation method. *The 10th IAEG International Congress*, 485, 1–7.
- Park, H. J., West, T. R., & Woo, I. (2005). Probabilistic analysis of rock slope stability and random properties of discontinuity parameters, Interstate Highway 40, Western North Carolina, USA. *Engineering Geology*, 79(3–4), 230–250. <https://doi.org/10.1016/j.enggeo.2005.02.001>
- Park, H., & West, T. R. (2000). Development of a probabilistic approach for rock wedge failure. *Engineering Geology*, 59(3–4), 233–251. [https://doi.org/10.1016/S0013-7952\(00\)00076-4](https://doi.org/10.1016/S0013-7952(00)00076-4)
- Pathak, S., Poudel, R. K., & Kansakar, B. R. (2006). Application of Probabilistic Approach in Rock Slope Stability Analysis-AN Experience from Nepal. *Universal Academy Press*, 1(Tokyo, Japan), 797–802.
- Scavia, C., Barla, G., & Bernaudo, V. (1990). Probabilistic stability analysis of block toppling failure in rock slopes. *International Journal of Rock Mechanics and Mining Sciences And*, 27(6), 465–478. [https://doi.org/10.1016/0148-9062\(90\)90999-1](https://doi.org/10.1016/0148-9062(90)90999-1)
- Sjöberg, J. (1999). *Analysis of Large Scale Rock Slope*. 1, 647. [https://doi.org/ISSN: 1402-1544; ISRN; LTU-DT--99/1--SE](https://doi.org/ISSN:1402-1544;ISRN;LTU-DT--99/1--SE)
- Stevens, W. D. (1998). *Rocfall: A TOOL FOR PROBABILISTIC ANALYSIS, DESIGN OF REMEDIAL MEASURES AND PREDICTION OF ROCKFALLS*.
- Tatone, B. S. A., & Grasselli, G. (2010). ROCKTOPPLE: A spreadsheet-based program for probabilistic block-toppling analysis. *Computers and Geosciences*, 36(1), 98–114. <https://doi.org/10.1016/j.cageo.2009.04.014>
- Wang, L., Hwang, J. H., Luo, Z., Juang, C. H., & Xiao, J. (2013). Probabilistic back analysis of slope failure - A case study in Taiwan. *Computers and Geotechnics*, 51(3), 12–23. <https://doi.org/10.1016/j.compgeo.2013.01.008>
- Zheng, J., Kulatilake, P. H. S. W., Shu, B., Sherizadeh, T., & Deng, J. (2014). Probabilistic block theory analysis for a rock slope at an open pit mine in USA. *Computers and Geotechnics*, 61, 254–265. <https://doi.org/10.1016/j.compgeo.2014.06.002>

Anexo B. Funciones de densidad de probabilidad

El estudio de fenómenos físicos de interés conlleva a realizar suposiciones o simplificaciones que permitan acoplar dichos fenómenos a los modelos de análisis representativos de cada caso particular sin perder la esencia de su comportamiento. La implementación de funciones de densidad de probabilidad es el primer paso para el análisis del comportamiento de los datos de cada una de las variables.

En muchas situaciones no se puede sugerir el uso de una función de densidad particular, sin embargo, comúnmente para la elección se realiza la comparación entre la forma del histograma de datos y la forma de una PDF (Función de densidad de probabilidad). La elección de dicha función de densidad debe basarse en la comprensión del fenómeno y su ajuste en el modelo. Su elección es de especial importancia debido a que a partir de este se realizarán los análisis y conclusiones del fenómeno estudiado. En Tabla 1 se presentan las funciones de probabilidad más usadas y sus tendencias centrales que brindan los mecanismos para la correcta elección de estas.

1.1 Función de densidad de probabilidad de funciones aleatorias continuas

Dentro de las funciones de densidad de probabilidad presentadas en la Tabla 1, se presentan funciones continuas y discretas. Para los casos en geotecnia, muchas de las variables dentro de su análisis estadístico se manejan como funciones de distribución continuas y algunas otras como la sismicidad con variables discretas. De las distribuciones de probabilidad continua se dice que la probabilidad que un valor X tome un valor entre ciertos límites a y b es igual al área bajo la función de densidad, y a partir de una integral se tiene que:

$$P(a < X < b) = \int_a^b f(x)dx \quad (1)$$

Donde la función $f(x)$ es una *función de densidad de probabilidad* (PDF) para la variable aleatoria continua X , definida en el conjunto de números reales, cumpliéndose las siguientes condiciones:

1. $f(x) \geq 0$, para toda $x \in \mathbb{R}$.
2. $\int_{-\infty}^{\infty} f(x)dx = 1$
3. $P(a < X < b) = \int_a^b f(x)dx$

Tabla 1. Funciones de densidad de probabilidad

Distribución	PDF	Media	Varianza	Explicación
Uniforme	$f(x; a, b) = \frac{1}{b-a}$ $a \leq x \leq b$	$\frac{a+b}{2}$	$\left(\frac{b-a}{12}\right)^2$	
Normal	$f(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $-\infty < x < \infty$	μ	σ^2	$x \in R$ $\mu \in R$ $\sigma > 0$
Normal truncada	$f(x; \bar{\mu}, \bar{\sigma}, a, b) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{\phi(\bar{\mu}, \bar{\sigma}; x)}{\Phi(\bar{\mu}, \bar{\sigma}; b) - \Phi(\bar{\mu}, \bar{\sigma}; a)} & \text{si } a < x < b \\ 0 & \text{si } b \geq x \end{cases}$	$\alpha = \frac{a-\bar{\mu}}{\bar{\sigma}}; \beta = \frac{b-\bar{\mu}}{\bar{\sigma}}$ $\mu = \bar{\mu} - \bar{\sigma} * \frac{\Phi(0,1; \beta) - \Phi(0,1; \alpha)}{\Phi(0,1; \beta) - \Phi(0,1; \alpha)}$	$\sigma^2 = \bar{\sigma}^2 * \left(1 - \frac{\beta\Phi(0,1; \beta) - \alpha\Phi(0,1; \alpha)}{\Phi(0,1; \beta) - \Phi(0,1; \alpha)} - \frac{(\Phi(0,1; \beta) - \Phi(0,1; \alpha))^2}{(\Phi(0,1; \beta) - \Phi(0,1; \alpha))^2}\right)$	$\bar{\mu}$ y $\bar{\sigma}$ son el valor medio y varianza de la distribución normal (a,b) rango de truncamiento - $-\infty = a; b = +\infty$ - $-\infty < a; b = +\infty$ - $-\infty = a; b < +\infty$ - $-\infty < a; b < +\infty$ Φ función de distribución ϕ función de densidad
Log-normal	$f(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $0 \leq x < \infty$	$e^{\mu + \frac{\sigma^2}{2}}$	$e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$	$x > 0$ $\mu > 0$ $\sigma > 0$
Exponencial o exponencial negativa	$f(x; \alpha) = \begin{cases} \frac{1}{\alpha} e^{-\frac{x}{\alpha}}, & x \geq 0 \\ 0, & x < 0 \end{cases}$	α	α^2	$x \geq 0$ $\alpha > 0$
Exponencial truncada	$f(x; \lambda, b) = \frac{\frac{1}{\lambda} e^{-x/\lambda}}{1 - e^{-b/\lambda}}$ $0 < x \leq b, \lambda > 0$	$\lambda \left[\frac{1 - (k+1)e^{-k}}{1 - e^{-k}} \right]$ $b = k\lambda, k > 0$	$2\lambda^2 \left[\frac{1 - \frac{1}{2}(k^2 + 2k + 2)e^{-k}}{1 - e^{-k}} \right]$	
Beta	$f(x; p, q) = \frac{1}{B(p, q)} x^{p-1}(1-x)^{q-1}$ $0 \leq x \leq 1$	$\frac{p}{p+q}$	$\frac{pq}{(p+q)^2(p+q+1)}$	$p > 0$ $q > 0$ $B(p, q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$
Poisson	$p(r; \mu) = \frac{\mu^r e^{-\mu}}{r!}$ $r \geq 0$	μ	μ	$\mu > 0$

Y la *función de distribución acumulativa* $F(x)$, de una variable aleatoria continua X con función de densidad $f(x)$, es:

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t)dt \quad \text{Para } -\infty < x < \infty \quad (2)$$

Entonces las ecuaciones serán las siguientes:

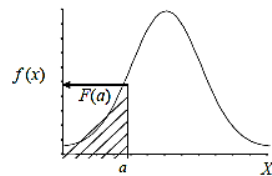
$$f(x) = \frac{dF(x)}{dx} \quad (3)$$

$$P(a < X < b) = F(b) - F(a)$$

Y de forma gráfica se podría identificar en la Figura 1, en donde la probabilidad es el área de la zona sombreada bajo la función de densidad de probabilidad.

Figura 1. Situaciones más comunes en distribuciones continuas

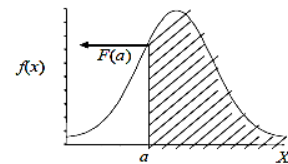
$P(X \leq a)$:



$$P(X \leq a) = \int_{-\infty}^a f(x) dx$$

$$P(X \leq a) = F(a)$$

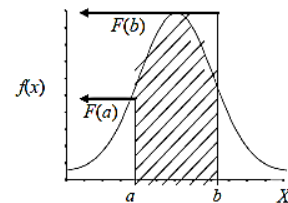
$P(X \geq a)$:



$$P(X \geq a) = \int_a^{+\infty} f(x) dx$$

$$P(X \geq a) = 1 - F(a)$$

$P(a \leq X \leq b)$:



$$P(a \leq X \leq b) = \int_a^b f(x) dx$$

$$P(a \leq X \leq b) = F(b) - F(a) \quad \text{si } b > a$$

1.2 Variables aleatorias independientes

En muchas situaciones, la información acerca del valor observado de una de las dos variables X y Y proporciona información acerca del valor de la otra variable. A y B son eventos independientes si y solo si la probabilidad de que ocurran ambos ($A \cap B$) es el producto de cada una de las probabilidades (DeVore, 2005).

$$P(A \cap B) = P(A|B) * P(B) = P(A) * P(B)$$

$$P(A/B) = P(A) \quad (4)$$

$$P(B/A) = P(B)$$

y en consecuencia:

$$P(A \cap B) = P(A) * P(B) \quad (5)$$

Se dice que dos variables aleatorias X y Y son independientes si para cada valor de x y y le corresponde uno en cada función,

$$p(x, y) = p_x(x) \cdot p_y(y) \text{ cuando } X \text{ y } Y \text{ son discretas} \quad (6)$$

$$f(x, y) = f_x(x) \cdot f_y(y) \text{ cuando } X \text{ y } Y \text{ son continuas} \quad (7)$$

$$F(x, y) = F_X(x) \cdot F_Y(y) \quad (8)$$

1.3 Variables aleatorias múltiples

El concepto de variables aleatorias y la distribución de probabilidad no solo se extiende para una o dos variables sino para múltiples variables aleatorias y su respectiva distribución de probabilidad conjunta. Con el propósito de identificar eventos del resultado de procesos físicos que se han convertido en términos numéricos, dichos eventos podrían ser mapeados entre dos o más dimensiones en el espacio real. Es así que:

$$(X = x \cap Y = y) \text{ y } (X \leq x, Y \leq y) \quad (9)$$

Como eventos simultáneos se tiene que:

$$(X = x \cap Y = y) \text{ y } (X \leq x \cap Y \leq y) \quad (10)$$

Definido por variables aleatorias en el espacio (x, y). Este concepto puede ser extendido a múltiples variables aleatorias. (Ang & Tang, 2007).

No solo se restringe el estudio de las distribuciones de probabilidad de espacios muestrales unidimensionales o experimentos con valores que tome una sola variable aleatoria, sino que a menudo se presentaran situaciones en las que se registren resultados simultáneos de diversas variables. Para el caso se puede tratar con funciones de distribución conjuntas para variables continuas (Walpole, Myers, Myers, & Ye, 2012).

1.4 Función de densidad de probabilidad conjunta para dos Variables aleatorias continuas.

Una función de distribución conjunta de dos variables aleatorias en donde se tiene a X y Y como dos variables aleatorias definidas en el mismo espacio de probabilidad se puede mostrar como la de función de densidad de variables aleatorias conjuntas.

$$F_{XY}(x, y) = P(X \leq x, Y \leq y) \quad (11)$$

$$P(x \leq a \cap y \leq b) = P(x \leq a) \cdot P(y \leq b) = Fx(a) \cdot Fy(b) \quad (12)$$

$$F(x, y) = P(x \leq a \cap y \leq b) \rightarrow \text{Funcion de distribucion conjunta de } x \text{ y } y \quad (13)$$

$$F(x, y) = P(X \leq x, Y \leq y) = P((X \leq x) \cap (Y \leq y)) \quad (14)$$

Donde la función F perteneciente al espacio maestral $R^2 - R$

Ahora, la probabilidad de que el valor observado de una variable aleatoria continua X este en un conjunto unidimensional A se obtiene integrando f(x) sobre el conjunto A. De manera similar, la probabilidad de que un par (X,Y) de variables aleatoria continuas se situé en un conjunto A bidimensional se obtiene integrando una función llamada función de densidad conjunta (DeVore, 2005).

Sean X y Y variables aleatorias continuas. Entonces f(x,y) es una función de densidad de probabilidad conjunta para X y Y si para cualquier conjunto bidimensional A

$$P((X, Y) \in A) = \int_A \int f(x, y) dx dy \quad (15)$$

En particular si A es un rectángulo bidimensional $(x, y): a \leq x \leq b, c \leq y \leq d)$ entonces

$$P([(X, Y) \in A] = P(a \leq X \leq b, c \leq Y \leq d) = \int_a^b \int_c^d f(x, y) dy dx \quad (16)$$

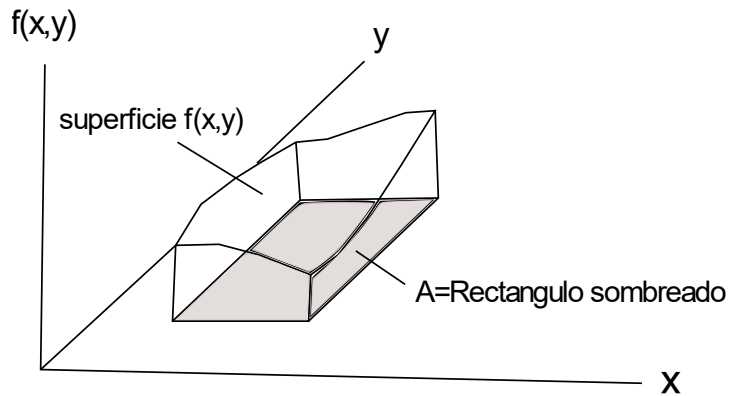
Siendo f(x, y) la función de densidad conjunta que cumple:

- 1) $f(x, y) \geq 0 \quad \forall x, y$
- 2) $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy = 1$

Se puede considerar que f(x, y) especifica una superficie a la altura f(x, y) arriba del punto (x, y) en un sistema coordenado tridimensional. Entonces $P[(x, y) \in A]$ es el volumen debajo

de esta superficie y arriba de la región A, análogo al área bajo la curva en el caso de una dimensión. Esto se ilustra en la siguiente figura:

Figura 2. $P[(x,y) \in A] = \text{volumen de la superficie de densidad arriba de A.}$ (DeVore, 2005)



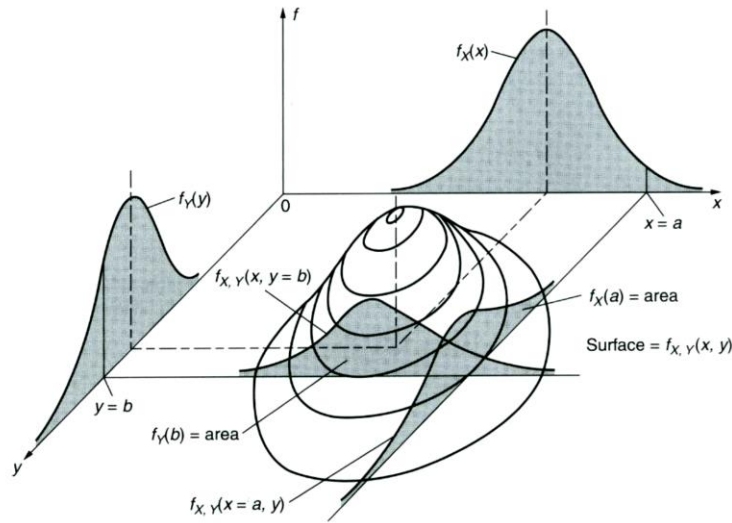
A partir de la PDF de X y Y, se puede calcular cada una de las dos funciones de densidad marginal. Las funciones de densidad de probabilidad marginal de X y Y, dadas por $f_x(x)$ y $f_y(y)$, respectivamente, están dadas por:

$$f_x(x) = \int_{-\infty}^{\infty} f(x,y) dy \text{ para } -\infty < x < \infty \quad (17)$$

$$f_y(y) = \int_{-\infty}^{\infty} f(x,y) dx \text{ para } -\infty < y < \infty \quad (18)$$

Las características de la función conjunta de variables aleatorias, x y y la asociada PDF marginal, es mostrada gráficamente a continuación:

Figura 3. Función de densidad conjunta y marginal de dos variables aleatorias continuas. (Ang & Tang, 2007)



1.5 Cálculo del volumen de funciones de densidad conjunta

Para calcular el volumen debajo de la superficie de la matriz se usa un método integral numérico, que para una superficie continua dada por: $z = f(x, y) \in R$ el volumen puede ser calculado como:

$$\iint_R f(x, y) dx dy \tag{19}$$

Y usando un método numérico, se puede escribir como:

$$\iint_R f(x, y) dx dy = \lim_{\Delta x \rightarrow 0} \lim_{\Delta y \rightarrow 0} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} f(x_i, y_j) \Delta x \Delta y \approx \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} f(x_i, y_j) \Delta x \Delta y \tag{20}$$

Donde m y n es el número de filas y numero de columnas de la matriz, respectivamente. En el proceso de computación, $(f(x_i, y_j) + f(x_i, y_{j+1}) + f(x_{i+1}, y_j) + f(x_{i+1}, y_{j+1}))/4$ es usado en lugar de $f(x_i, y_i)$.

Distribución de Fisher

La distribución Fisher fue desarrollada por el estadístico británico R.A Fisher, y plantea que a cada dirección se le asigna un peso unitario y se representa por un punto en una esfera de radio unitario (Fisher, 1953). La función de distribución de Fisher $f(\theta)$ suministra la

probabilidad por unidad de área angular de encontrar una dirección dentro de un área angular centrada en un ángulo θ , que es un ángulo entre orientación medida y orientación verdadera.

$$f(\theta) = \frac{k e^{k \cos \theta}}{4\pi \sinh k} \quad (21)$$

Donde θ es la desviación angular de la orientación media en un rango entre 0 y $\pi/2$, k es la constante de Fisher la cual es la medida de la concentración de la distribución sobre la verdadera dirección media; k es 0 para una distribución de direcciones que es uniforme sobre la esfera, y se aproxima a ∞ para direcciones concentradas en un punto.

Luego la probabilidad de encontrar una dirección específica dentro de un área dA será:

$$p = f(\theta) * dA \quad (22)$$

$$p = \int_{dA} \frac{k e^{k \cos \theta}}{4\pi \sinh k} dA \quad (23)$$

$$dA = \sin \theta d\theta d\varepsilon \quad (24)$$

$$p = \int_{\varepsilon_0}^{\varepsilon_1} \int_{\theta_0}^{\theta_1} \frac{k e^{k \cos \theta}}{4\pi \sinh k} \sin \theta d\theta d\varepsilon \quad (25)$$

Donde $0 \leq \varepsilon \leq 2\pi$ y $0 \leq \theta \leq \pi$

Para ε de 0 a 2π

$$p = 2\pi \int_{\theta_0}^{\theta_1} \frac{k e^{k \cos \theta}}{4\pi \sinh k} \sin \theta d\theta \quad (26)$$

$$p = \int_{\theta_0}^{\theta_1} \frac{k e^{k \cos \theta}}{2 \sinh k} \sin \theta d\theta \quad (27)$$

Y la probabilidad acumulada se calcularía de la siguiente manera:

$$P(\theta) = \int_0^{\theta} \frac{k e^{k \cos \theta}}{2 \sinh k} \sin \theta d\theta \quad (28)$$

$$P(\theta) = \frac{e^k (e^{k \cos \theta} - e^k)}{1 - e^{2k}} \quad (29)$$

$$P(\theta) = \frac{e^{k \cos \theta + k} - e^{2k}}{1 - e^{2k}} \quad \text{para } \theta \text{ de } 0 \text{ a } \pi \quad (30)$$

$$P(\pi) = 1 \quad (31)$$

Y la probabilidad truncada en $0 \leq \theta \leq \pi/2$

$$P_T(\theta) = P(\theta) * \lambda \quad (32)$$

$$\frac{1}{\lambda} = P\left(\frac{\pi}{2}\right) = \frac{e^k - e^{2k}}{1 - e^{2k}} \quad (33)$$

$$\frac{1}{\lambda} = \frac{e^k (1 - e^k)}{1 - e^{2k}} \rightarrow \lambda = \frac{e^{-k} (1 - e^{2k})}{1 - e^k} \quad (34)$$

Remplazando en la ecuación (32) los resultados de las ecuaciones (30) y (34)

$$P_T(\theta) = \left(\frac{e^{k \cos \theta + k} - e^{2k}}{1 - e^{2k}} \right) \left(\frac{e^{-k} (1 - e^{2k})}{1 - e^k} \right) = \frac{e^k (1 - e^{k(\cos \theta - 1)})}{e^k - 1} \quad (35)$$

Si $k \geq 3 \rightarrow e^k - 1 \approx e^k$ se simplifica la ecuación anterior siguiendo una expresión aproximada para la distribución acumulada. (H. Park & West, 2000).

$$P(\theta) \approx 1 - e^{k(\cos \theta - 1)} \quad (36)$$

Anexo C: Datos estructurales caso de estudio

Datos estructurales Bloque Norte

No	Año	Planos		Polos		Cosenos directores		
		DD	D	α	β	l	m	n
1	2011	36	76	216	14	0.196	0.142	0.970
2	2016	47	79	227	11	0.130	0.140	0.982
3	2016	47	66	227	24	0.277	0.297	0.914
4	2016	48	68	228	22	0.251	0.278	0.927
5	2016	49	71	229	19	0.214	0.246	0.946
6	2011	54	68	234	22	0.220	0.303	0.927
7	2016	54	66	234	24	0.239	0.329	0.914
8	2016	59	90	239	0	0.000	0.000	1.000
9	2016	62	79	242	11	0.090	0.168	0.982
10	2016	64	87	244	3	0.023	0.047	0.999
11	2016	64	83	244	7	0.053	0.110	0.993
12	2016	71	76	251	14	0.079	0.229	0.970
13	2016	74	66	254	24	0.112	0.391	0.914
14	2016	75	87	255	3	0.014	0.051	0.999
15	2016	75	88	255	2	0.009	0.034	0.999
16	2016	75	67	255	23	0.101	0.377	0.921
17	2016	78	81	258	9	0.033	0.153	0.988
18	2016	79	77	259	13	0.043	0.221	0.974
19	2016	80	59	260	31	0.089	0.507	0.857
20	2016	82	79	262	11	0.027	0.189	0.982
21	2016	83	62	263	28	0.057	0.466	0.883
22	2016	84	77	264	13	0.024	0.224	0.974
23	2016	89	77	269	13	0.004	0.225	0.974
24	2016	93	64	273	26	-0.023	0.438	0.899
25	2016	94	81	274	9	-0.011	0.156	0.988
26	2016	129	69	309	21	-0.226	0.279	0.934
27	2016	134	47	314	43	-0.474	0.491	0.731
28	2016	138	73	318	17	-0.217	0.196	0.956
29	2016	139	67	319	23	-0.295	0.256	0.921
30	2016	141	53	321	37	-0.468	0.379	0.799
31	2016	146	74	326	16	-0.229	0.154	0.961
32	2016	147	74	327	16	-0.231	0.150	0.961
33	2016	149	80	329	10	-0.149	0.089	0.985
34	2016	152	70	332	20	-0.302	0.161	0.940
35	2016	154	82	334	8	-0.125	0.061	0.990
36	2011	156	66	336	24	-0.372	0.165	0.914
37	2016	158	68	338	22	-0.347	0.140	0.927
38	2016	166	81	346	9	-0.152	0.038	0.988
39	2016	166	78	346	12	-0.202	0.050	0.978
40	2016	170	89	350	1	-0.017	0.003	1.000
41	2016	172	85	352	5	-0.086	0.012	0.996
42	2016	174	84	354	6	-0.104	0.011	0.995
43	2016	176	89	356	1	-0.017	0.001	1.000
44	2011	182	62	2	28	-0.469	-0.016	0.883
45	2016	224	16	44	74	-0.691	-0.668	0.276
46	2016	244	11	64	79	-0.430	-0.882	0.191
47	2016	253	16	73	74	-0.281	-0.919	0.276
48	2016	254	19	74	71	-0.261	-0.909	0.326
49	2016	273	29	93	61	0.046	-0.873	0.485
50	2016	278	19	98	71	0.132	-0.936	0.326
51	2016	289	11	109	79	0.320	-0.928	0.191
52	2016	296	21	116	69	0.409	-0.839	0.358
53	2016	296	24	116	66	0.400	-0.821	0.407
54	2016	299	22	119	68	0.450	-0.811	0.375
55	2016	305	11	125	79	0.563	-0.804	0.191
56	2016	306	39	126	51	0.457	-0.629	0.629
57	2016	308	23	128	67	0.567	-0.725	0.391
58	2016	313	34	133	56	0.565	-0.606	0.559
59	2016	315	20	135	70	0.664	-0.664	0.342
60	2016	316	10	136	80	0.708	-0.684	0.174
61	2016	318	28	138	62	0.656	-0.591	0.469
62	2016	318	34	138	56	0.616	-0.555	0.559
63	2016	319	19	139	71	0.714	-0.620	0.326
64	2016	324	29	144	61	0.708	-0.514	0.485
65	2016	347	59	167	31	0.502	-0.116	0.857
66	2016	349	90	169	0	0.000	0.000	1.000
67	2016	353	86	173	4	0.069	-0.009	0.998
68	2016	356	90	176	0	0.000	0.000	1.000
69	2016	356	60	176	30	0.499	-0.035	0.866
70	2016	357	89	177	1	0.017	-0.001	1.000
71	2016	358	86	178	4	0.070	-0.002	0.998

\sum	2.250	5.720	23.873
l^2, m^2, n^2	5.060	32.722	569.925
R		24.652	
IR, mR, nR	0.091	0.232	0.968
Buzamiento β	76		
Azímuto α	69		
Polos β	14		
α	249		
N		25	
KF		71.780	
$\bar{\psi}$		0.118	
P		5.0%	
cos Ψ		0.99929	
Ψ		2.166174	

\sum	-4.481	2.620	17.858
l^2, m^2, n^2	20.081	6.864	318.894
R		18.597	
IR, mR, nR	-0.241	0.141	0.960
Buzamiento β	74		
Azímuto α	150		
Polos β	16		
α	330		
N		19	
KF		47.117	
$\bar{\psi}$		0.146	
P		5.0%	
cos Ψ		0.99891	
Ψ		2.673746	

\sum	6.311	-14.980	7.333
l^2, m^2, n^2	39.829	224.407	53.777
R		17.833	
IR, mR, nR	0.354	-0.840	0.411
Buzamiento β	24		
Azímuto α	293		
Polos β	66		
α	113		
N		20	
KF		9.229	
$\bar{\psi}$		0.329	
P		5.0%	
cos Ψ		0.99444	
Ψ		6.043557	

\sum	1.157	-0.163	6.718
l^2, m^2, n^2	1.339	0.026	45.134
R		6.819	
IR, mR, nR	0.170	-0.024	0.985
Buzamiento β	80		
Azímuto α	352		
Polos β	10		
α	172		
N		7	
KF		38.676	
$\bar{\psi}$		0.161	
P		5.0%	
cos Ψ		0.99867	
Ψ		2.951187	

Datos estructurales Bloque Sur

No	Año	Plano		Polos		Cosenos directores		
		DD	D	α	β	l	m	n
1	2016	5	81	185	9	0.156	0.014	0.988
2	2016	6	74	186	16	0.274	0.029	0.961
3	2016	12	70	192	20	0.335	0.071	0.940
4	2016	13	79	193	11	0.186	0.043	0.982
5	2016	15	72	195	18	0.298	0.080	0.951
6	2016	17	70	197	20	0.327	0.100	0.940
7	2016	18	89	198	1	0.017	0.005	1.000
8	2011	20	77	200	13	0.211	0.077	0.974
9	2016	25	84	205	6	0.095	0.044	0.995
10	2011	34	73	214	17	0.242	0.163	0.956
11	2016	34	64	214	26	0.363	0.245	0.899
12	2016	35	80	215	10	0.142	0.100	0.985
13	2016	37	82	217	8	0.111	0.084	0.990
14	2016	37	88	217	2	0.028	0.021	0.999
15	2016	37	82	217	8	0.111	0.084	0.990
16	2016	39	78	219	12	0.162	0.131	0.978
17	2011	40	85	220	5	0.067	0.056	0.996
18	2011	40	86	220	4	0.053	0.045	0.998
19	2011	40	76	220	14	0.185	0.156	0.970
20	2016	42	84	222	6	0.078	0.070	0.995
21	2016	43	86	223	4	0.051	0.048	0.998
22	2011	44	68	224	22	0.269	0.260	0.927
23	2011	45	76	225	14	0.171	0.171	0.970
24	2016	45	89	225	1	0.012	0.012	1.000
25	2011	48	82	228	8	0.093	0.103	0.990
26	2016	48	76	228	14	0.162	0.180	0.970
27	2011	50	66	230	24	0.261	0.312	0.914
28	2011	50	82	230	8	0.089	0.107	0.990
29	2011	50	82	230	8	0.089	0.107	0.990
30	2016	50	86	230	4	0.045	0.053	0.998
31	2011	52	77	232	13	0.138	0.177	0.974
32	2011	53	68	233	22	0.225	0.299	0.927
33	2016	53	69	233	21	0.216	0.286	0.934
34	2011	54	80	234	10	0.102	0.140	0.985
35	2011	54	76	234	14	0.142	0.196	0.970
36	2011	55	80	235	10	0.100	0.142	0.985
37	2011	56	76	236	14	0.135	0.201	0.970
38	2011	56	78	236	12	0.116	0.172	0.978
39	2011	58	86	238	4	0.037	0.059	0.998
40	2011	58	82	238	8	0.074	0.118	0.990
41	2011	60	71	240	19	0.163	0.282	0.946
42	2011	60	86	240	4	0.035	0.060	0.998
43	2011	60	62	240	28	0.235	0.407	0.883
44	2016	60	85	240	5	0.044	0.075	0.996
45	2016	61	85	241	5	0.042	0.076	0.996
46	2011	62	60	242	30	0.235	0.441	0.866
47	2016	63	69	243	21	0.163	0.319	0.934
48	2011	64	70	244	20	0.150	0.307	0.940
49	2016	64	66	244	24	0.178	0.366	0.914
50	2011	65	80	245	10	0.073	0.157	0.985
51	2011	66	80	246	10	0.071	0.159	0.985
52	2016	66	74	246	16	0.112	0.252	0.961
53	2016	69	83	249	7	0.044	0.114	0.993
54	2011	70	76	250	14	0.083	0.227	0.970
55	2011	70	76	250	14	0.083	0.227	0.970
56	2011	70	84	250	6	0.036	0.098	0.995
57	2011	71	76	251	14	0.079	0.229	0.970
58	2011	72	88	252	2	0.011	0.033	0.999
59	2016	73	89	253	1	0.005	0.017	1.000
60	2011	74	68	254	22	0.103	0.360	0.927
61	2011	74	60	254	30	0.138	0.481	0.866
62	2011	75	77	255	13	0.058	0.217	0.974
63	2011	75	75	255	15	0.067	0.250	0.966
64	2016	75	79	255	11	0.049	0.184	0.982
65	2011	76	59	256	31	0.125	0.500	0.857
66	2011	76	74	256	16	0.067	0.267	0.961
67	2011	76	77	256	13	0.054	0.218	0.974
68	2011	76	74	256	16	0.067	0.267	0.961
69	2016	76	81	256	9	0.038	0.152	0.988
70	2011	80	69	260	21	0.062	0.353	0.934
71	2011	86	79	266	11	0.013	0.190	0.982
72	2011	90	78	270	12	0.000	0.208	0.978
73	2011	90	78	270	12	0.000	0.208	0.978
74	2016	100	74	280	16	-0.048	0.271	0.961
75	2016	102	72	282	18	-0.064	0.302	0.951

Σ	8.490	13.204	73.470
l^2, m^2, n^2	72.075	174.348	5397.780
R	0.113	0.176	0.978
lR, mR, nR			
Buzamiento β	78		
Azimuth α	57		
Polos β	12		
α	237		
N		76	
KF		87.147	
ψ		0.107	
P		5.0%	
$\cos \Psi$		0.99941	
Ψ		1.965909	

76	2016	107	80	287	10	-0.051	0.166	0.985
77	2011	118	78	298	12	-0.098	0.184	0.978
78	2016	118	73	298	17	-0.137	0.258	0.956
79	2016	120	80	300	10	-0.087	0.150	0.985
80	2016	125	73	305	17	-0.168	0.239	0.956
81	2016	127	89	307	1	-0.011	0.014	1.000
82	2011	130	77	310	13	-0.145	0.172	0.974
83	2011	130	72	310	18	-0.199	0.237	0.951
84	2011	130	78	310	12	-0.134	0.159	0.978
85	2016	130	85	310	5	-0.056	0.067	0.996
86	2016	130	72	310	18	-0.199	0.237	0.951
87	2016	131	68	311	22	-0.246	0.283	0.927
88	2016	132	74	312	16	-0.184	0.205	0.961
89	2011	134	84	314	6	-0.073	0.075	0.995
90	2011	135	76	315	14	-0.171	0.171	0.970
91	2016	135	72	315	18	-0.219	0.219	0.951
92	2016	135	77	315	13	-0.159	0.159	0.974
93	2011	136	72	316	18	-0.222	0.215	0.951
94	2011	138	80	318	10	-0.129	0.116	0.985
95	2011	140	86	320	4	-0.053	0.045	0.998
96	2011	144	76	324	14	-0.196	0.142	0.970
97	2011	144	74	324	16	-0.223	0.162	0.961
98	2011	146	86	326	4	-0.058	0.039	0.998
99	2011	148	86	328	4	-0.059	0.037	0.998
100	2011	148	82	328	8	-0.118	0.074	0.990
101	2011	148	86	328	4	-0.059	0.037	0.998
102	2011	150	88	330	2	-0.030	0.017	0.999
103	2011	150	80	330	10	-0.150	0.087	0.985
104	2011	150	87	330	3	-0.045	0.026	0.999
105	2011	151	70	331	20	-0.299	0.166	0.940
106	2011	152	81	332	9	-0.138	0.073	0.988
107	2011	152	76	332	14	-0.214	0.114	0.970
108	2011	154	86	334	4	-0.063	0.031	0.998
109	2011	155	80	335	10	-0.157	0.073	0.985
110	2011	156	82	336	8	-0.127	0.057	0.990
111	2011	159	82	339	8	-0.130	0.050	0.990
112	2011	160	79	340	11	-0.179	0.065	0.982
113	2011	160	90	340	0	0.000	0.000	1.000
114	2011	160	82	340	8	-0.131	0.048	0.990
115	2011	162	75	342	15	-0.246	0.080	0.966
116	2011	162	81	342	9	-0.149	0.048	0.988
117	2011	162	75	342	15	-0.246	0.080	0.966
118	2016	164	86	344	4	-0.067	0.019	0.998
119	2011	166	80	346	10	-0.168	0.042	0.985
120	2011	166	66	346	24	-0.395	0.098	0.914
121	2016	166	84	346	6	-0.101	0.025	0.995
122	2011	170	81	350	9	-0.154	0.027	0.988
123	2011	170	82	350	8	-0.137	0.024	0.990
124	2011	171	83	351	7	-0.120	0.019	0.993
125	2011	171	88	351	2	-0.034	0.005	0.999
126	2011	175	88	355	2	-0.035	0.003	0.999
127	2011	176	80	356	10	-0.173	0.012	0.985
128	2011	178	62	358	28	-0.469	0.016	0.883
129	2011	179	76	359	14	-0.242	0.004	0.970
130	2011	180	84	360	6	-0.105	0.000	0.995
131	2016	180	84	360	6	-0.105	0.000	0.995
132	2016	185	82	360	5	-0.139	-0.012	0.990
133	2016	190	89	360	10	-0.217	-0.003	0.990
134	2016	194	89	360	14	-0.311	-0.004	0.990
135	2016	202	83	360	22	-0.443	-0.046	0.993
136	2016	204	80	360	24	-0.480	-0.071	0.985
137	2011	205	89	360	1	-0.016	-0.007	1.000
138	2016	208	78	360	12	-0.184	-0.098	0.978
139	2016	203	81	360	9	-0.144	-0.061	0.988
140	2011	286	23	106	67	0.254	-0.885	0.391
141	2011	294	26	114	64	0.366	-0.821	0.438
142	2011	294	22	114	68	0.377	-0.847	0.375
143	2011	294	15	114	75	0.393	-0.882	0.259
144	2011	295	16	115	74	0.406	-0.871	0.276
145	2011	296	14	116	76	0.425	-0.872	0.242
146	2011	298	15	118	75	0.453	-0.853	0.259
147	2016	299	22	119	68	0.450	-0.811	0.375
148	2011	300	16	120	74	0.481	-0.832	0.276
149	2011	305	12	125	78	0.561	-0.801	0.208
150	2011	305	22	125	68	0.532	-0.760	0.375
151	2011	305	17	125	73	0.549	-0.783	0.292
152	2011	306	36	126	54	0.476	-0.655	0.588
153	2011	310	17	130	73	0.615	-0.733	0.292
154	2011	310	14	130	76	0.624	-0.743	0.242
155	2011	312	28	132	62	0.591	-0.656	0.469
156	2011	312	18	132	72	0.636	-0.707	0.309
157	2011	312	12	132	78	0.655	-0.727	0.208
158	2011	316	20	136	70	0.676	-0.653	0.342
159	2011	316	20	136	70	0.676	-0.653	0.342
160	2011	318	12	138	78	0.727	-0.655	0.208

\sum	-8.799	4.705	61.708
l^2, m^2, n^2	77.424	22.137	3807.821
R		62.509	
lR, mR, nR	-0.141	0.075	0.987
Buzamiento β	81		
Azimuth α	152		
Polos β	9		
α	332		
N		63	
KF		128.324	
$\bar{\psi}$		0.088	
P		5.0%	
cos Ψ		0.99960	
Ψ		1.620053	

\sum	10.920	-16.199	6.764
l^2, m^2, n^2	119.253	262.417	45.758
R	20.674		
lR, mR, nR	0.528	-0.784	0.327
Buzamiento β	19		
Azimuth α	304		
Polos β	71		
α	124		
N		21	
KF		64.487	
$\bar{\psi}$		0.125	
P		5.0%	
cos Ψ		0.99920	
Ψ		2.285398	

Anexo D: Desarrollo de metodologías y presentación de resultados por la herramienta propuesta.

Explicación del uso del programa desarrollado y ejecución con datos del presente documento

En el desarrollo del presente documento se da a conocer una herramienta de cálculo en el lenguaje de programación Python, en este programa denominado Análisis probabilístico de estabilidad de taludes en Roca se exponen y se obtienen los resultados de las diferentes metodologías propuestas, con éste se genera las diferentes graficas mostradas en el documento y quedara a disposición del lector para su aplicación.

Para el manejo de la herramienta es necesario generar ciertos archivos que requerirá el programa para ejecutarse entre estos esta: rumbos y buzamientos de las discontinuidades, ángulo de fricción y cohesión del material.

Los datos para importar se deberán presentar en archivos independientes en formato con extensión .csv que para el caso se usó Excel.csv de la siguiente manera:

Figura 1. Ejemplo de formato de datos de orientación para importar en formato csv

	A	B	C	D	E
1	rumbo_ang	rumbo_dir	buzamiento_ar	buzamiento_dir	
2		40 SE		35 NE	
3		42 SE		35 NE	
4		40 SE		39 NE	
5		30 SW		60 NW	
6		35 SW		61 NW	
7		41 SE		34 NE	
8		32 SW		59 NW	
9		35 SW		62 NW	
10		38 SE		37 NE	
11		40 SE		37 NE	

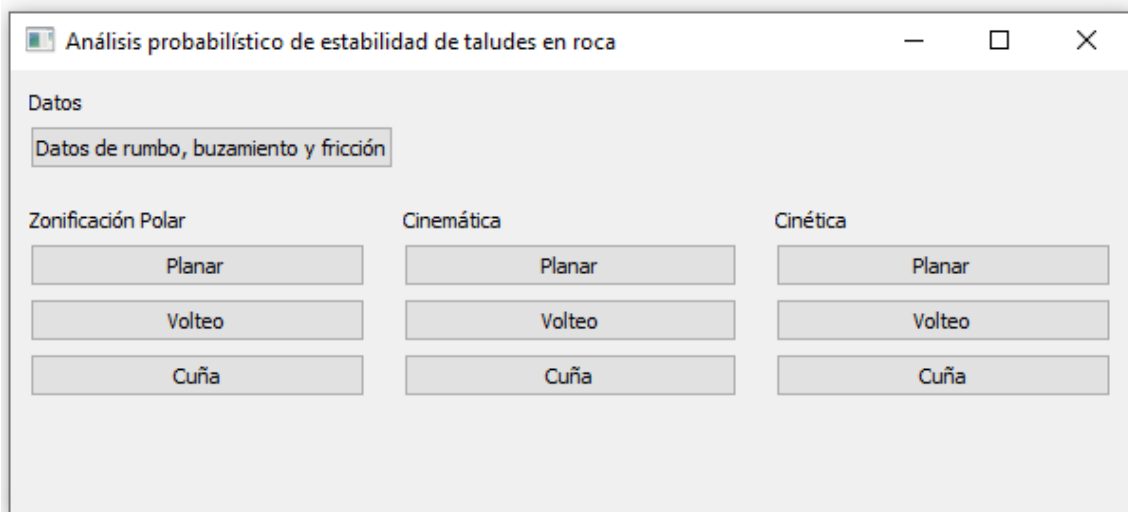
Figura 2. Ejemplo de formato de datos fricción y cohesión para importar en formato csv

	A		A
1	phi	1	Cr2
2	50	2	15.6
3	49	3	15.5
4	49	4	15.1
5	47	5	15
6	46	6	14.8
7	46	7	14.7
8	45	8	14.6
9	45	9	14.3

Luego teniendo los respectivos datos organizados en archivos independientes se procederá a ejecutar el programa.

Al ejecutar el programa saldrá esta primera ventana de inicio, en donde aparecerá las pestañas con todos los análisis a ejecutar.

Figura 3. Interfaz de Ventana de inicio del programa



En esta sección se ingresará a la opción de Datos – Datos de rumbo, buzamiento y fricción y se remitirá a la sección de Sectorización cinemática allí se importarán los datos que requiere el programa en la pestaña - Leer datos desde archivo-, cargados todos los datos se seleccionará el mecanismo de falla a analizar Planar, Volteo o cuña y se le asignará el valor del ángulo de fricción para el cono de fricción ver Figura 4.

Sección sectorización cinemática

Una vez en la sección de Sectorización cinemática se indicará los valores requeridos como ángulo de fricción para el mecanismo a analizar.

Figura 4. Interfaz de sección de Sectorización cinemática

	rumbo_eng	rumbo_dir	buzamiento_eng	buzamiento_dir
1	40	SE	35	NE
2	42	SE	35	NE
3	40	SE	39	NE
4	30	SW	60	NW
5	35	SW	61	NW
6	41	SE	34	NE
7	32	SW	59	NW
8	35	SW	62	NW

Luego para comenzar con el análisis se volverá a la ventana de inicio y se seleccionará el mecanismo a analizar. Por ejemplo, Zonificación polar - Planar.

Datos

Datos de rumbo, buzamiento y fricción

Zonificación Polar

Cinemática

Cinética

Planar

Volteo

Cuña

El resultado serán la gráfica de sectorización para falla planar, cuña o volteo de las familias encontradas con los datos del documento se presentan en las siguientes figuras.

Figura 5. Gráfica de sectorización para falla planar

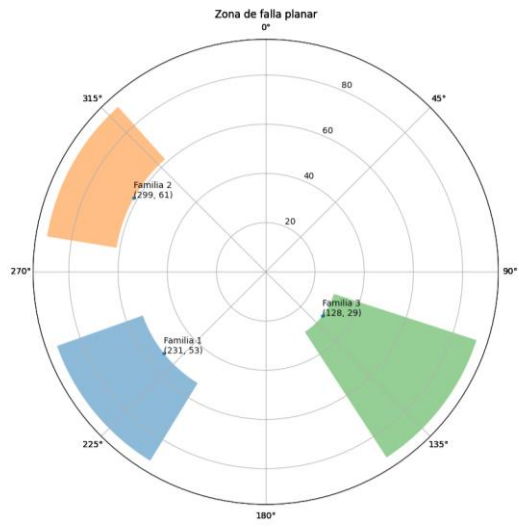


Figura 6. Gráfica de sectorización para falla en cuña

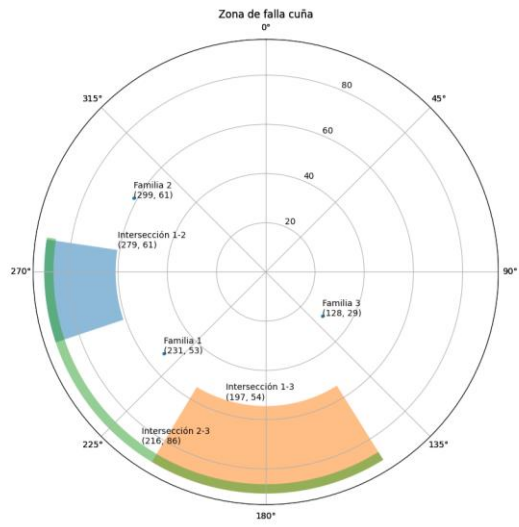
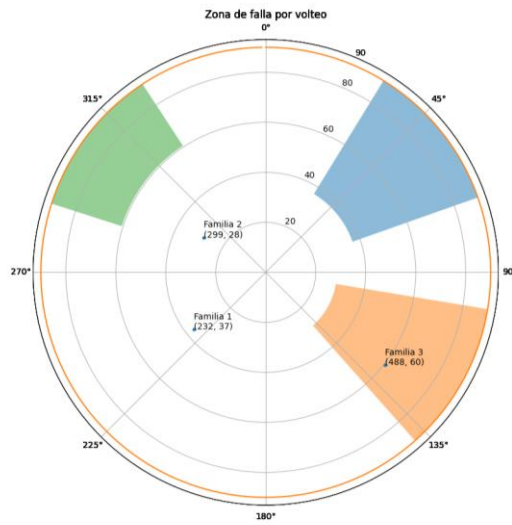


Figura 7. Gráfica de sectorización para falla por volteo



Sección Probabilidad – Análisis cinemático

Para cada mecanismo se asignará el valor de la orientación de la envolvente o lo que es la orientación del talud a analizar.

Para el ejemplo se asignó 231° y se volverá a la pantalla de inicio y se seleccionará el mecanismo a analizar en este caso planar.

Falla planar

Figura 8. Interfaz para Sección Probabilidad – Análisis cinemático – Planar F1

	rumbo_ang	rumbo_dir	buzamiento_ang	buzamiento_dir
1	40	SE	35	NE
2	42	SE	35	NE
3	40	SE	39	NE
4	30	SW	60	NW
5	35	SW	61	NW
6	41	SE	34	NE
7	32	SW	59	NW
8	35	SW	62	NW

Se genera las gráficas de la zona de falla crítica para cada falla planar con límite de la envolvente con orientación dada y finalmente la probabilidad conjunta entre la probabilidad cinemática calculada, la probabilidad por ángulo de fricción limitada por la envolvente. Ver Figura 12 y Figura 13.

Figura 9. Zona de falla critica falla planar F1

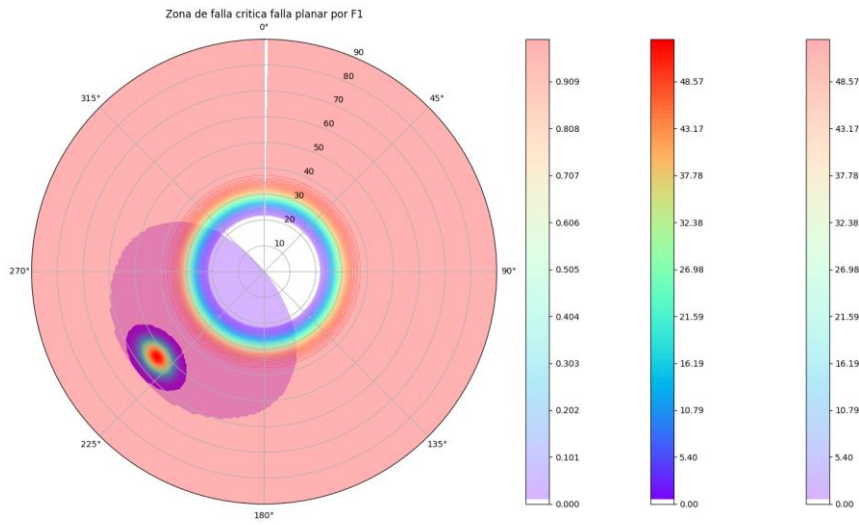
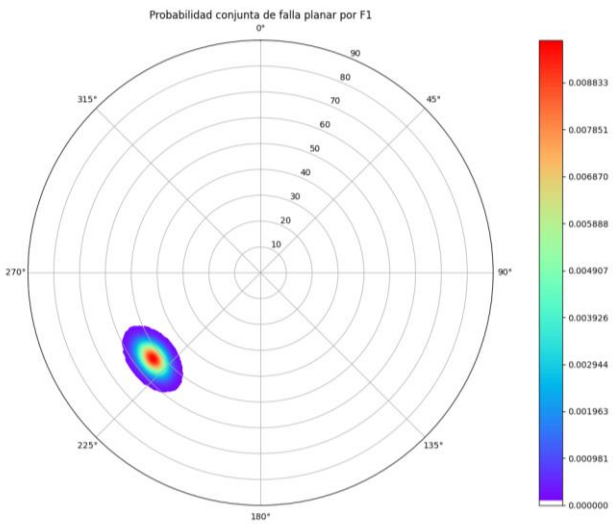


Figura 10. Probabilidad conjunta falla planar F1



Probabilidad de falla para F1: 0.9790409943621898

Y para la Familia 2

Figura 11. Interfaz para Sección Probabilidad – Análisis cinemático – Planar F2

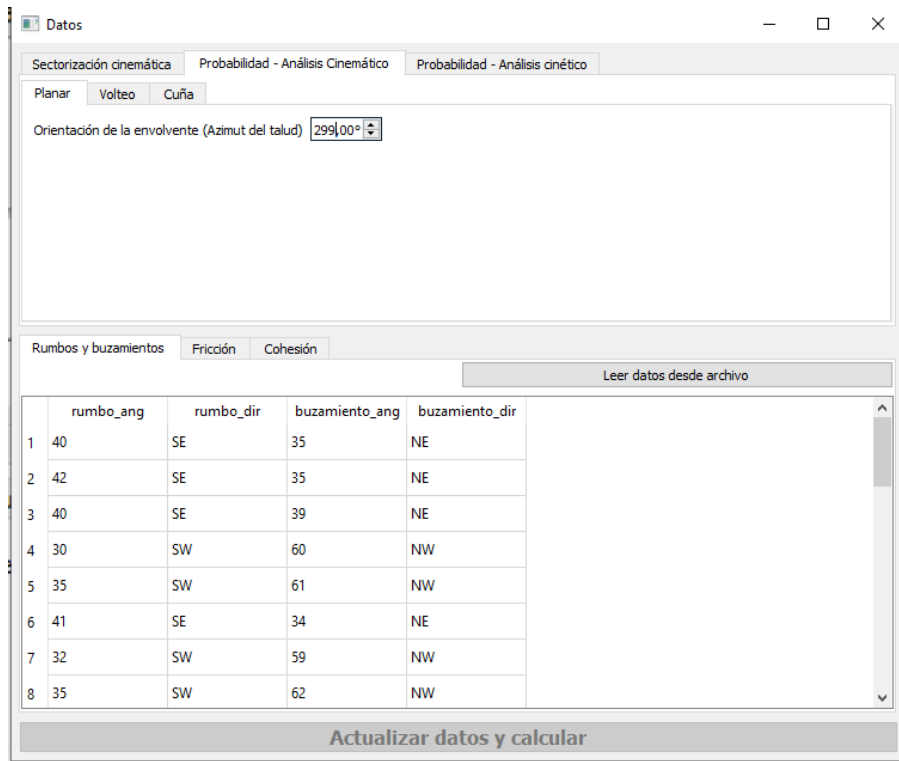


Figura 12. Zona de falla crítica falla planar F2

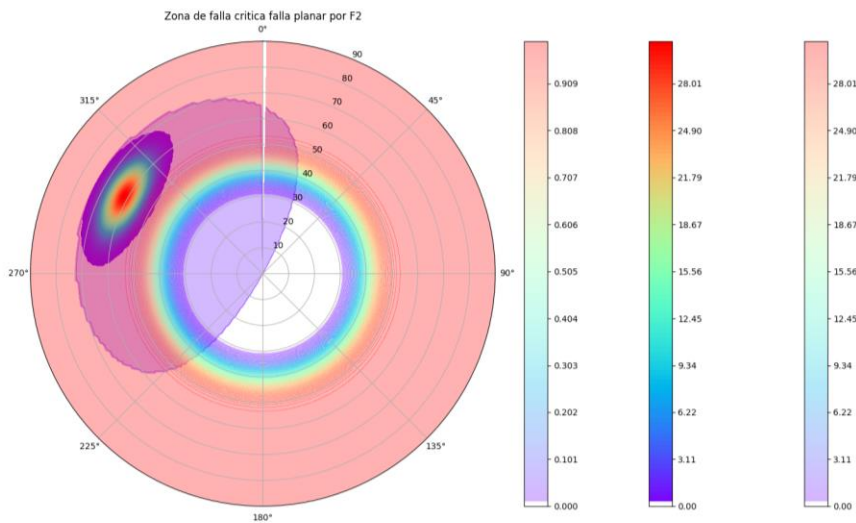
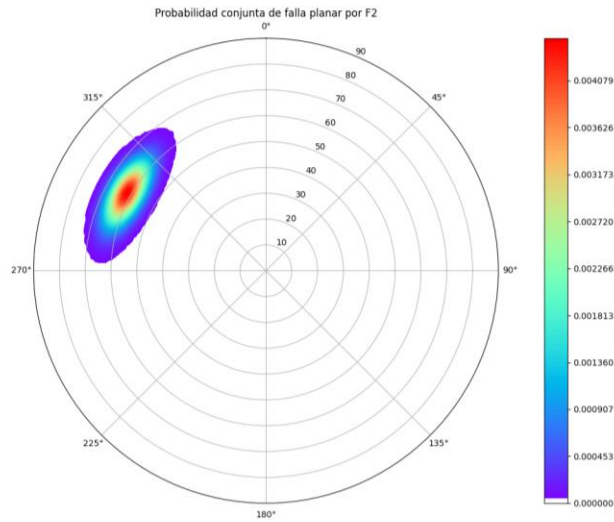


Figura 13. Probabilidad conjunta falla planar F2



Probabilidad de falla para F2: 0.9992706038095928

Falla en cuña

Luego para falla en cuña se procederá de la misma forma que con la planar y los resultados obtenidos se muestran a continuación.

Figura 14. Interfaz para Sección Probabilidad – Análisis cinemático – Cuña F1-F2

Datos

Sectorización cinemática Probabilidad - Análisis Cinemático Probabilidad - Análisis cinético

Planar Volteo **Cuña**

Orientación de la envolvente (Azimut del talud)

Rumbos y buzamientos Fricción Cohesión

Leer datos desde archivo

	rumbo_ang	rumbo_dir	buzamiento_ang	buzamiento_dir
1	40	SE	35	NE
2	42	SE	35	NE
3	40	SE	39	NE
4	30	SW	60	NW
5	35	SW	61	NW
6	41	SE	34	NE
7	32	SW	59	NW
8	35	SW	62	NW

Actualizar datos y calcular

Figura 15. Zona de falla critica falla Cuña F1-F2

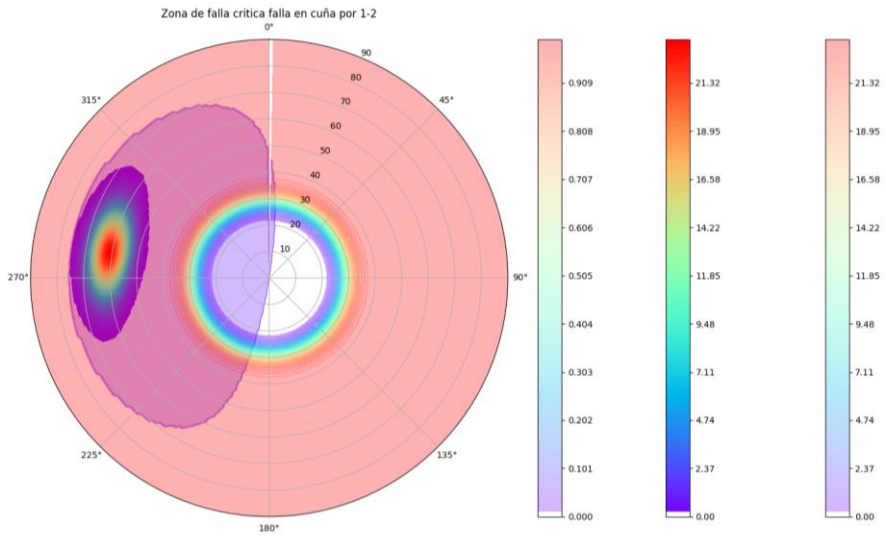


Figura 16. Probabilidad conjunta falla Cuña F1-F2

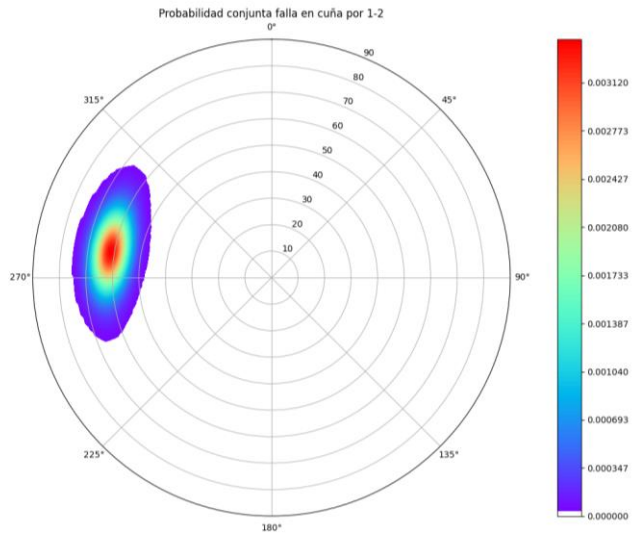


Figura 17. Interfaz para Sección Probabilidad – Análisis cinemático – Cuña F1-F3

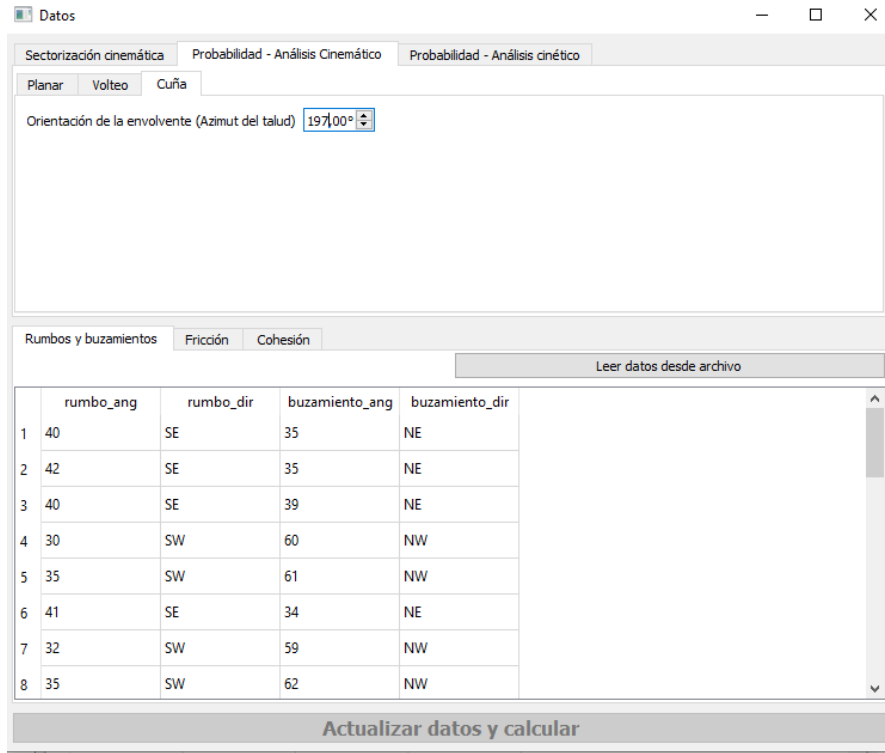


Figura 18. Zona de falla crítica falla Cuña F1-F3

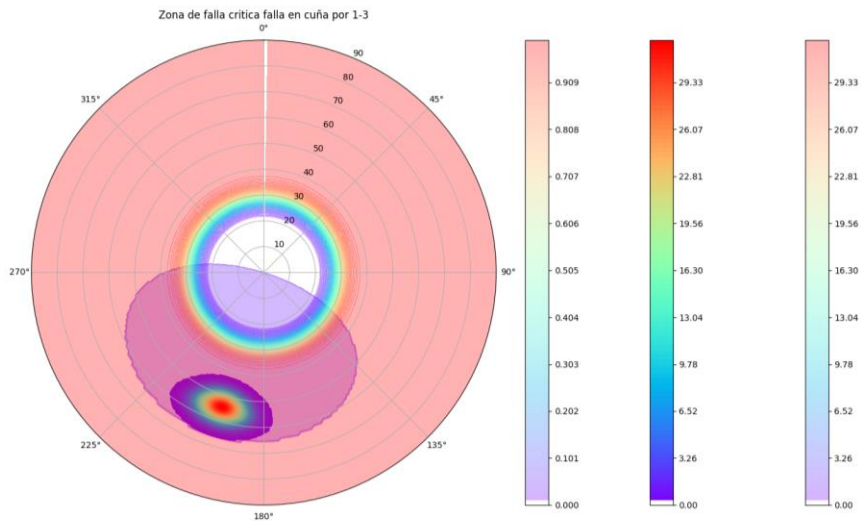
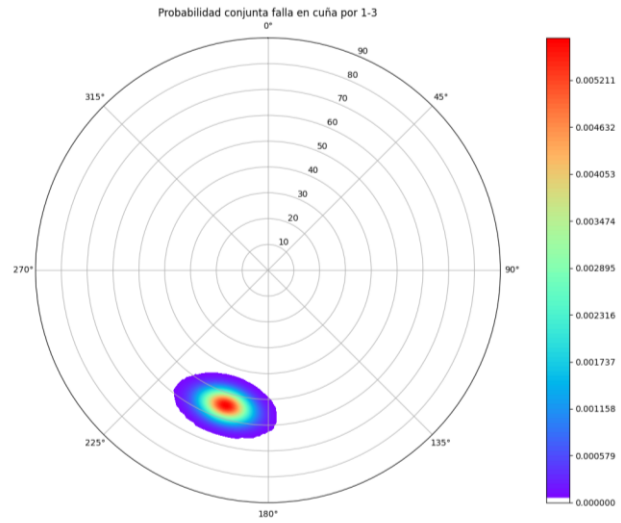


Figura 19. Probabilidad conjunta falla Cuña F1-F3



Probabilidad de falla para F1-3: 0.9838278533091511

Falla por volteo

Para falla por volteo se realizará el mismo procedimiento que se viene haciendo.

Figura 20. Interfaz para Sección Probabilidad – Análisis cinemático – Volteo F1

Datos

Sectorización cinemática Probabilidad - Análisis Cinemático Probabilidad - Análisis cinético

Planar Volteo Cuña

Orientación de la envolvente (Azimut del talud)

Rumbos y buzamientos Fricción Cohesión

Leer datos desde archivo

	rumbo_ang	rumbo_dir	buzamiento_ang	buzamiento_dir
1	40	SE	35	NE
2	42	SE	35	NE
3	40	SE	39	NE
4	30	SW	60	NW
5	35	SW	61	NW
6	41	SE	34	NE
7	32	SW	59	NW
8	35	SW	62	NW

Actualizar datos y calcular

Figura 21. Zona de falla crítica falla por volteo F1

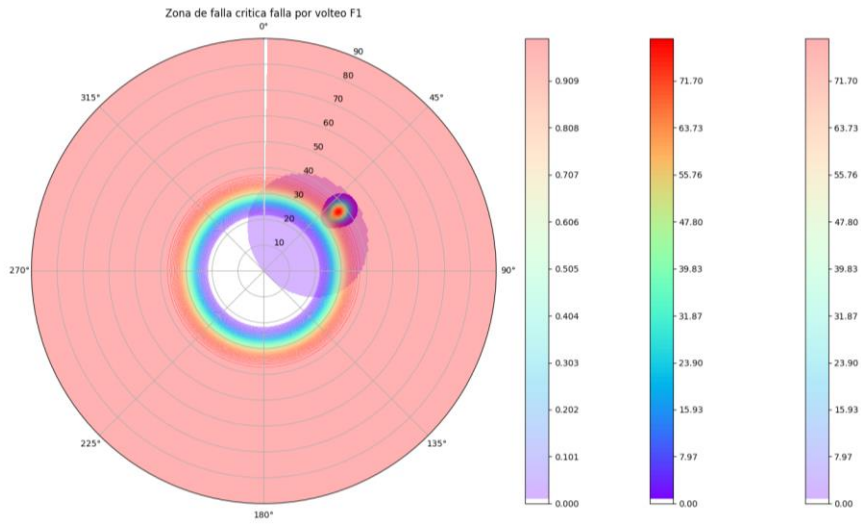
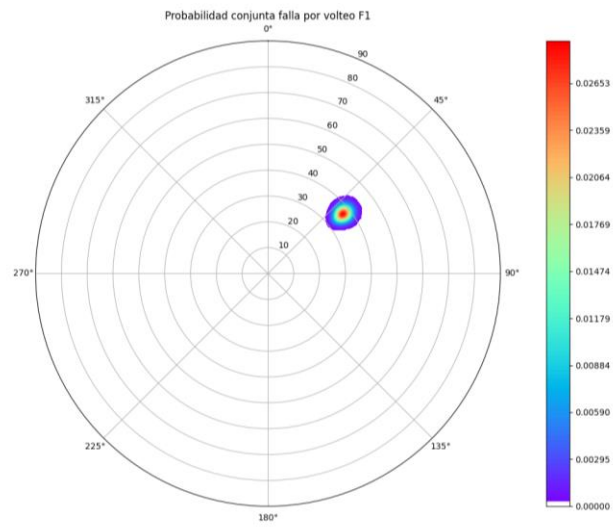


Figura 22. Probabilidad conjunta falla por volteo F1



Probabilidad de falla para F1: 1.4690031161024037

Sección Probabilidad – Análisis cinético

Para el análisis de la potabilidad por análisis cinemático se procederá a completar la información solicitada para cada una de las pestañas y en cada uno de los mecanismos de falla.

Figura 23. Interfaz para Sección Probabilidad – Análisis cinético – Planar

Datos

Sectorización cinemática Probabilidad - Análisis Cinemático Probabilidad - Análisis cinético

Planar Volteo Cuña

Parámetros del plano deslizante

Inclinación - Theta: 45,00

Parámetros de la distribución Beta

Alfa: 3,00 Beta: 3,00

Amenaza sísmica

Tiempo de retorno: 475,00 años Periodo: 50,00 años

Parámetros del bloque deslizante

Altura - h: 2,00 Peso unitario - Gama: 2,20

Rumbos y buzamientos Fricción Cohesión

Leer datos desde archivo

	rumbo_ang	rumbo_dir	buzamiento_ang	buzamiento_dir
1	40	SE	35	NE
2	42	SE	35	NE
3	40	SE	39	NE
4	30	SW	60	NW
5	35	SW	61	NW
6	41	SE	34	NE
7	32	SW	59	NW
8	35	SW	62	NW

Actualizar datos y calcular

Figura 24. Volumen de probabilidad conjunta con funciones de densidad Beta

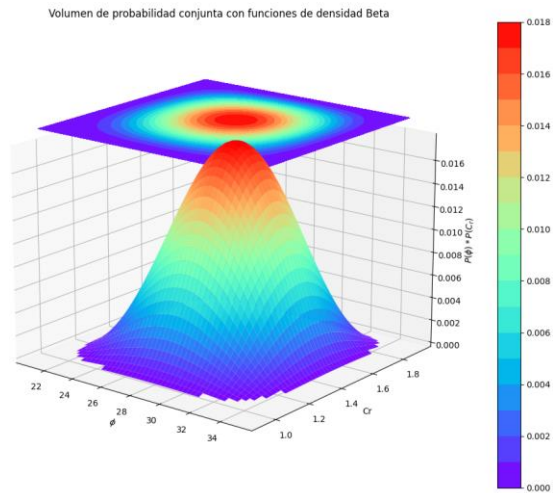
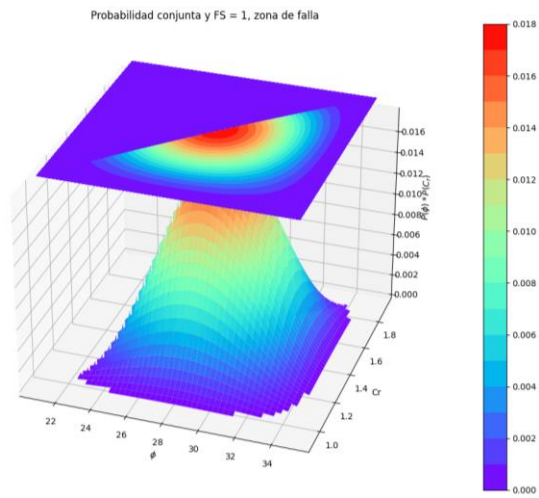


Figura 25. Probabilidad conjunta y FS= 1, zona de falla



Probabilidad de falla: 0.04722957603494041

Figura 26. Interfaz para Sección Probabilidad – Análisis cinético – Volteo

Datos

Sectorización cinemática Probabilidad - Análisis Cinemático Probabilidad - Análisis cinético

Planar Volteo Cuña

Parámetros del talud

Inclinación - theta: 60,00°

Parámetros de la distribución Beta

Alfa: 3,00 Beta: 3,00

Amenaza sísmica

Tiempo de retorno: 475,00 años Periodo: 50,00 años

Rumbos y buzamientos Fricción Cohesión

Leer datos desde archivo

	rumbo_ang	rumbo_dir	buzamiento_ang	buzamiento_dir
1	40	SE	35	NE
2	42	SE	35	NE
3	40	SE	39	NE
4	30	SW	60	NW
5	35	SW	61	NW
6	41	SE	34	NE
7	32	SW	59	NW
8	35	SW	62	NW

Actualizar datos y calcular

Figura 27. Volumen de probabilidad conjunta con funciones de densidad Beta Volteo

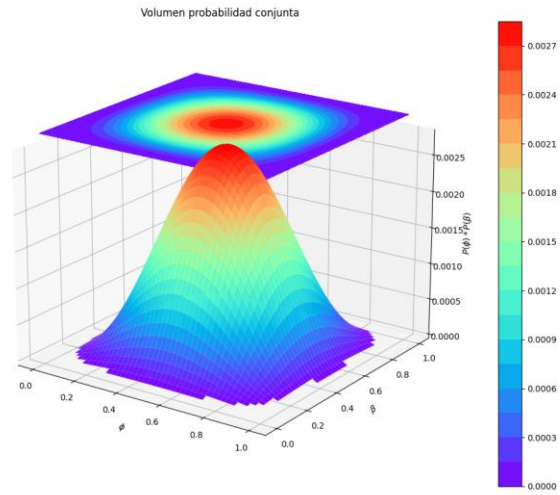


Figura 28. Probabilidad conjunta, inclinación de superficie de falla 60°

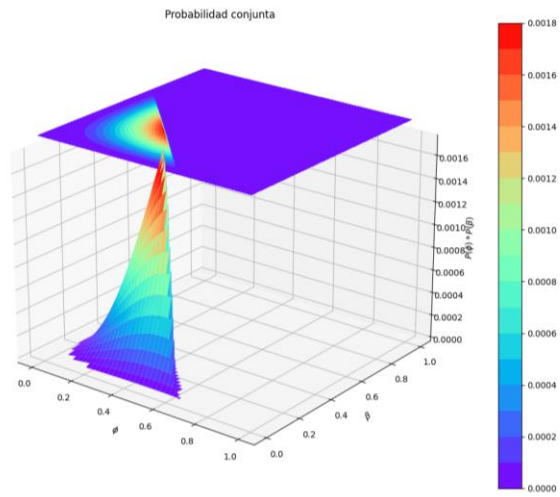


Figura 29. Figura 30. Interfaz para Sección Probabilidad – Análisis cinético – Cuña

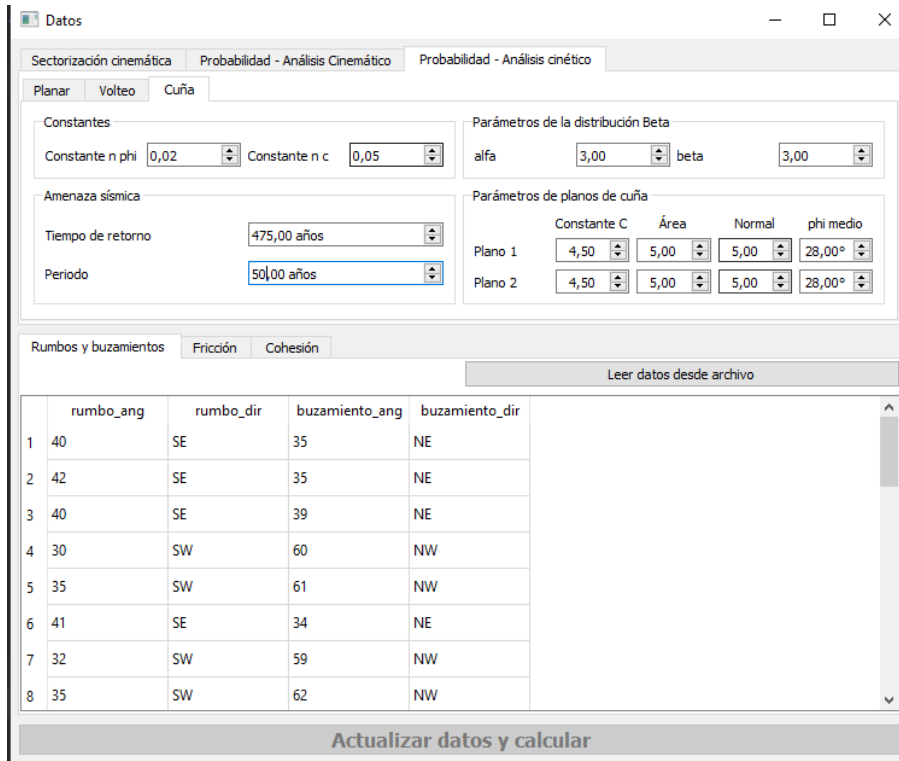


Figura 31. Volumen de probabilidad final para fricción

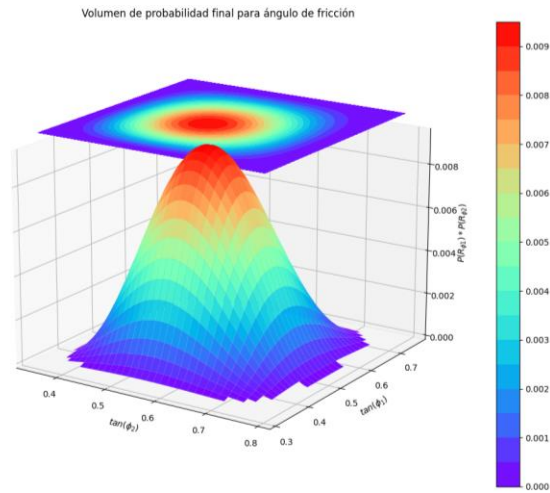
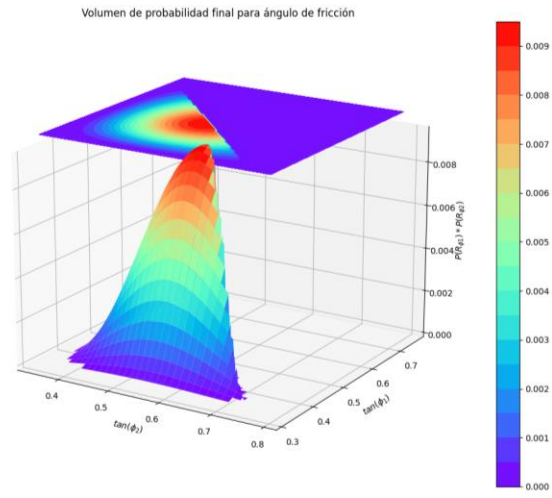


Figura 32. Zona de falla sobre volumen de probabilidad final para cohesión



Probabilidad de falla: 0.02540328312873719

Figura 33. Volumen de probabilidad final para cohesión

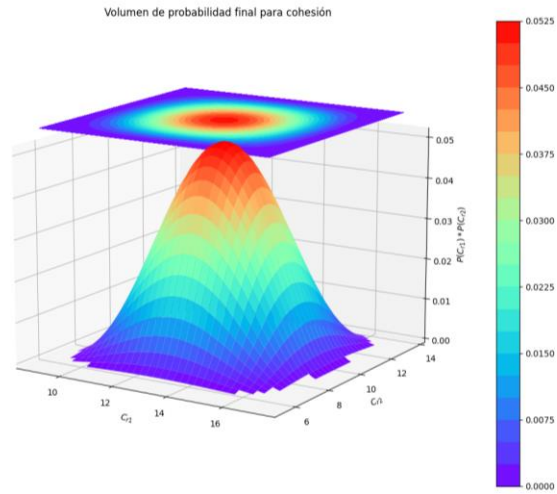
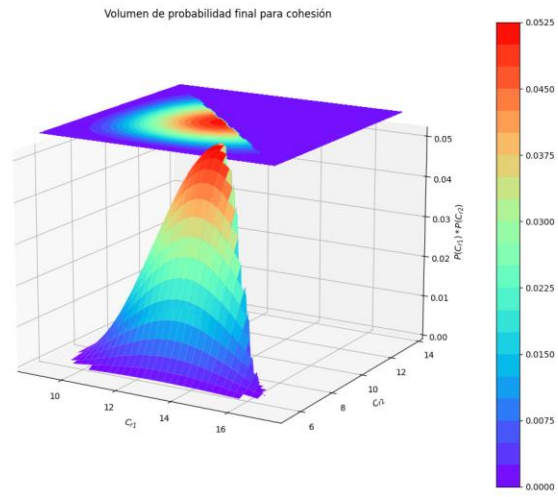


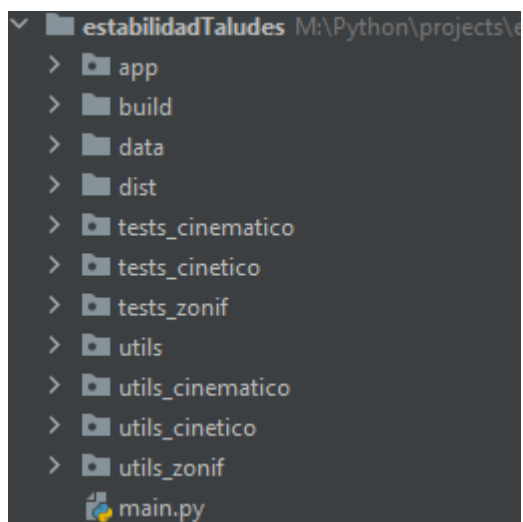
Figura 34. Zona de falla sobre volumen de probabilidad final para cohesión



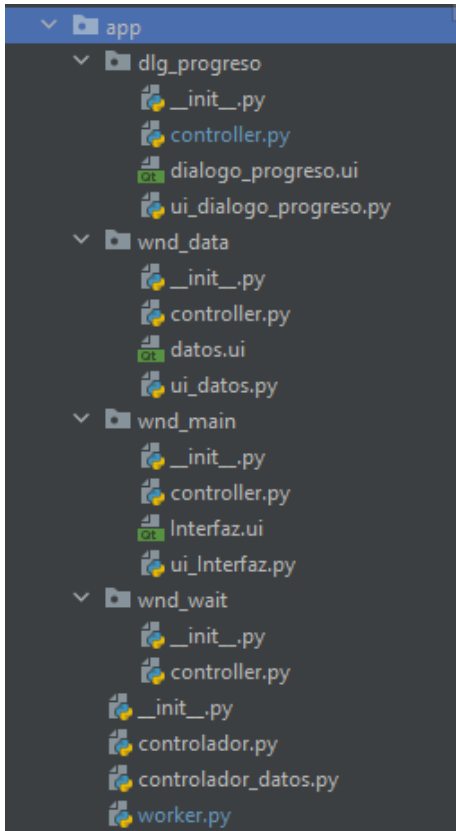
Probabilidad de falla: 0.02540328312873719

Anexo E: Código fuente del programa.

1. ESTRUCTURA DE MÓDULOS DE LA APLICACIÓN ESTABILIDAD TALUDES EN ROCA (PROBABILIDAD) - PCTR



2. MÓDULO APP



app/controlador.py

```
import matplotlib.pyplot as plt
from PySide2 import QtCore as Qtc
from PySide2 import QtWidgets as Qtw

from app.controlador_datos import DataCtl, DataOut
from app.dlg_progreso.controller import ProgressWindow
from app.wnd_data.controller import DataWindowCtrl
from app.wnd_main.controller import MainWindowCtr
from app.worker import Worker
import utils as du
import utils.graficas_utils as gu

class Controller(Qtw.QApplication):
    # Señales del controlador al controlador bkg
    sgn_iniciar_tarea = Qtc.Signal(int, arguments='metodo_ejecutado')

    def __init__(self, argv):
        """
        Controlador de la aplicación.
        :param argv:
        """
        super().__init__(argv)
```

```

self.datos = DataCtl()
self.datos_out = DataOut()

# Crear threadpool
self.threadpool = Qtc.QThreadPool()

# Figuras
self.figuras = {
    Worker.MTDS.cinematica_planar: {},
    Worker.MTDS.cinematica_volteo: {},
    Worker.MTDS.cinematica_cunia: {},
    Worker.MTDS.cinetica_deslizante: {},
    Worker.MTDS.cinetica_volteo: {},
    str(Worker.MTDS.cinetica_cunia) + 'friccion': {},
    str(Worker.MTDS.cinetica_cunia) + 'cohesion': {},
    Worker.MTDS.zonif_planar: None,
    Worker.MTDS.zonif_volteo: None,
    Worker.MTDS.zonif_cunnia: None,
}

# Crear ventanas
self.ventana_principal = MainWindowCtr()
self.ventana_datos = DataWindowCtrl(self.datos)
self.ventana_espera = ProgressWindow(self.ventana_principal, 0, 100)

# Mostrar ventana principal
self.ventana_principal.show()

# Enlazar acciones de botones de ventana principal
self.ventana_principal.sgn_btn.connect(self.abrir_ventana)
self.ventana_principal.destroyed.connect(self.closeAllWindows)

# Conectar señales de ventana de datos
self.ventana_datos.sgn_dlg_result.connect(self.leer_datos_csv)

self.ventana_datos.sgn_enable_buttons.connect(self.ventana_principal.enable_buttons)

def abrir_ventana(self, ventana: int):
    """
    Método conectado a los botones de la ventana principal
    :param ventana: Botón presionado en la ventana principal
    :return: None
    """

    if ventana == self.ventana_principal.BTNS.datos:
        self.cargar_datos()
    else:
        # Comprueba si se han cargado los datos de entrada
        if self.datos.datos_rumbos is None or self.datos.datos_friccion is
None or self.datos.datos_cohesion \
            is None:
            dialogo = Qtw.QMessageBox(self.ventana_principal)

```

```

        dialogo.setWindowTitle('Error...')
        dialogo.setText('Sin datos de análisis')
        dialogo.setInformativeText('Cargue los datos de análisis usando
el botón "Datos de rumbo, '
                                'buzamiento y dirección"')
        dialogo.setIcon(Qtw.QMessageBox.Icon.Critical)
        dialogo.exec_()
        return

    if ventana == self.ventana_principal.BTNS.cinematica_deslizamiento:
        self.cinematica_deslizamiento()
    elif ventana == self.ventana_principal.BTNS.cinematica_volteo:
        self.cinematica_volteo()
    elif ventana == self.ventana_principal.BTNS.cinematica_cunia:
        self.cinematica_cunia()
    if ventana == self.ventana_principal.BTNS.cinetica_deslizamiento:
        self.cinetica_deslizamiento()
    elif ventana == self.ventana_principal.BTNS.cinetica_volteo:
        self.cinetica_volteo()
    elif ventana == self.ventana_principal.BTNS.cinetica_cunia:
        self.cinetica_cunia()
    elif ventana == self.ventana_principal.BTNS.zonif_planar:
        self.zonificacion_planar()
    elif ventana == self.ventana_principal.BTNS.zonif_volteo:
        self.zonificacion_volteo()
    elif ventana == self.ventana_principal.BTNS.zonif_cunnia:
        self.zonificacion_cunnia()

    def cargar_datos(self):
        self.mensaje('Abrir ventana de datos')
        self.ventana_datos.show()

    def cinematica_deslizamiento(self):
        self.iniciar_bkg_task(Worker.MTDS.cinematica_planar, 'Cinémática
planar')

    def cinematica_volteo(self):
        self.iniciar_bkg_task(Worker.MTDS.cinematica_volteo, 'Cinémática
volteo')

    def cinematica_cunia(self):
        self.iniciar_bkg_task(Worker.MTDS.cinematica_cunia, 'Cinémática cuña')

    def cinetica_deslizamiento(self):
        self.iniciar_bkg_task(Worker.MTDS.cinetica_deslizante, 'Cinética
planar')

    def cinetica_volteo(self):
        self.iniciar_bkg_task(Worker.MTDS.cinetica_volteo, 'Cinética volteo')

    def cinetica_cunia(self):
        self.iniciar_bkg_task(Worker.MTDS.cinetica_cunia, 'Cinética cuña')

    def zonificacion_planar(self):

```

```

        self.iniciar_bkg_task(Worker.MTDS.zonif_planar, 'Zonificación planar')

def zonificacion_volteo(self):
    self.iniciar_bkg_task(Worker.MTDS.zonif_volteo, 'Zonificación volteo')

def zonificacion_cunnia(self):
    self.iniciar_bkg_task(Worker.MTDS.zonif_cunnia, 'Zonificación cuña')

def leer_datos_csv(self, dialogo: int, ruta: str):
    self.mensaje('Cargando datos')
    if dialogo == self.ventana_datos.BTNS.rumbos:
        self.datos.datos_rumbos = du.leer_desde_csv(ruta, ';')
        self.ventana_datos.cargar_tabla_datos(self.datos.datos_rumbos)
    elif dialogo == self.ventana_datos.BTNS.friccion:
        self.datos.datos_friccion = du.leer_desde_csv(ruta, ';')
        self.ventana_datos.cargar_tabla_friccion(self.datos.datos_friccion)
    elif dialogo == self.ventana_datos.BTNS.cohesion:
        self.datos.datos_cohesion = du.leer_desde_csv(ruta, ';')
        self.ventana_datos.cargar_tabla_cohesion(self.datos.datos_cohesion)

def detener_wait_window(self):
    self.ventana_espera.close()

def iniciar_bkg_task(self, metodo, mensaje):
    self.ventana_espera.update_progress(10, 'Calculando ' + mensaje)
    self.ventana_espera.show()
    self.mensaje(mensaje)
    trabajo_extra = Worker(self.datos, self.datos_out, metodo)
    trabajo_extra.signals.error.connect(self.tarea_error)
    trabajo_extra.signals.finished.connect(self.tarea_finished)
    self.threadpool.start(trabajo_extra)

def tarea_error(self, params: tuple):
    print(f'[App Controller] error de tarea {params}')

def tarea_finished(self, metodo: int):
    print(f'[App Controller] Tarea terminada {metodo}')

    self.ventana_espera.update_progress(50, 'Graficando...')

    # Crear figura si no existe
    if metodo not in self.figuras:
        self.figuras[metodo] = plt.figure()
        figura = self.figuras[metodo]

    if metodo == Worker.MTDS.cinematica_planar:
        self.graficar_cinematico(self.datos_out.cinematica_planar,
self.figuras[metodo])
    elif metodo == Worker.MTDS.cinematica_volteo:
        self.graficar_cinematico(self.datos_out.cinematica_volteo,
self.figuras[metodo])
    elif metodo == Worker.MTDS.cinematica_cunia:
        self.graficar_cinematico(self.datos_out.cinematica_cunia,
self.figuras[metodo])

```

```

        elif metodo == Worker.MTDS.cinetica_deslizante:
            self.graficar_cinetico(self.datos_out.cinetica_desliz,
self.figuras[metodo])
        elif metodo == Worker.MTDS.cinetica_volteo:
            self.graficar_cinetico(self.datos_out.cinetica_volteo,
self.figuras[metodo])
        elif metodo == Worker.MTDS.cinetica_cunia:
            self.graficar_cinetico(self.datos_out.cinetica_cunia_friccion,
self.figuras[str(metodo) + 'friccion'])
            self.graficar_cinetico(self.datos_out.cinetica_cunia_cohesion,
self.figuras[str(metodo) + 'cohesion'])
        elif metodo == Worker.MTDS.zonif_planar:
            self.graficar_zonificacion_planar(self.figuras, metodo)
        elif metodo == Worker.MTDS.zonif_volteo:
            self.graficar_zonificacion_volteo(self.figuras, metodo)
        elif metodo == Worker.MTDS.zonif_cunnia:
            self.graficar_zonificacion_cunnia(self.figuras, metodo)

self.ventana_espera.update_progress(100, 'Terminado')
self.ventana_espera.close()

def graficar_zonificacion_cunnia(self, figuras, metodo):
    if figuras[metodo] is None:
        figuras[metodo] = plt.figure()

    gu.grafica_zonif_cunnia(self.datos_out,
self.datos.sector_cunnia.ang_friccion, figuras[metodo])
    figuras[metodo].show()

def graficar_zonificacion_planar(self, figuras, metodo):
    if figuras[metodo] is None:
        figuras[metodo] = plt.figure()
    gu.grafica_zonif_planar(self.datos_out.zonif_planar.repr_df,
self.datos.sector_planar.ang_friccion,
        figuras[metodo])
    figuras[metodo].show()

def graficar_zonificacion_volteo(self, figuras, metodo):
    if figuras[metodo] is None:
        figuras[metodo] = plt.figure()

    gu.grafica_zonif_volteo(self.datos_out.zonif_volteo.repr_df,
self.datos.sector_volteo.ang_friccion,
        figuras[metodo])
    figuras[metodo].show()

def graficar_cinematico(self, datos, graficas: dict):
    for datos_familia in datos.calculos_familia:
        if datos_familia.idx in graficas:
            plt.close(graficas[datos_familia.idx][0])
            plt.close(graficas[datos_familia.idx][1])

        graficas[datos_familia.idx] =
(plt.subplots(subplot_kw={'projection': 'polar'})[0],

```



```

plt.subplots(subplot_kw={'projection': 'polar'})[0])

        gu.grafica_contorno(datos_familia.env_df, 'env',
datos_familia.max_level, 0.3,
                                graficas[datos_familia.idx][0],
titulo=datos_familia.labels.titulo)
        gu.grafica_contorno(datos_familia.paleom_df, 'fp',
datos_familia.max_level, 1.,
                                graficas[datos_familia.idx][0])
        gu.grafica_contorno(datos.friccion_df, 'pdf',
datos.friccion_df['pdf'].max(), 0.3,
                                graficas[datos_familia.idx][0])

        if datos_familia.probabilidad > 0:
            gu.grafica_contorno(datos_familia.paleom_df, 'conj',
datos_familia.paleom_df['conj'].max(), 1.0,
                                graficas[datos_familia.idx][1],
titulo=datos_familia.labels.titulo_2,
                                anotation=f'Probabilidad de falla para
F{datos_familia.idx}: '
                                                f'{datos_familia.probabilidad}')

            graficas[datos_familia.idx][0].show()
            graficas[datos_familia.idx][1].show()

def graficar_cinetico(self, datos, graficas: dict):
    # Graficar en 3D
    if 'fig1' in graficas:
        plt.close(graficas['fig1'])
        plt.close(graficas['fig2'])

    graficas['fig1'] = plt.figure()
    graficas['fig2'] = plt.figure()

    gu.grafica_3d_xyz(datos.axes_df[datos.eje1], datos.axes2_df[datos.eje2],
datos.red_df, datos.eje1,
                                datos.eje2, datos.eje_z1, titulo=datos.labels.titulo,
x_label=datos.labels.x_label,
                                y_label=datos.labels.y_label,
z_label=datos.labels.z_label, figure=graficas['fig1'])
    gu.grafica_3d_xyz(datos.axes_df[datos.eje1], datos.axes2_df[datos.eje2],
datos.red_df, datos.eje1,
                                datos.eje2, datos.eje_z2,
titulo=datos.labels.titulo_2, x_label=datos.labels.x_label,
                                y_label=datos.labels.y_label,
z_label=datos.labels.z_label, figure=graficas['fig2'],
                                anotation=f'Probabilidad de falla:
{datos.probabilidad}')

    graficas['fig1'].show()
    graficas['fig2'].show()

@staticmethod

```

```
def mensaje(mensaje: str):
    print('[App Controller]', mensaje)
```

app/controlador_datos.py

```
from typing import List
```

```
from dataclasses import dataclass, field
```

```
import pandas as pd
```

```
import utils.data_utils as du
```

```
@dataclass
class Plano:
    c: float
    A: float
    N: float
    phi: float
```

```
@dataclass
class Zonificacion:
    ang_friccion: float = 20.
```

```
@dataclass
class Cinematica:
    ang_talud: float = 315
```

```
@dataclass
class CineticaPlanar:
    theta: float = 90 # Ángulo de inclinación del plano deslizando
    alfa: float = 3 # Parámetro de la distribución beta
    beta: float = 3 # Parámetro de la distribución beta
    t_retorno: float = 475 # Tiempo de retorno de la actividad sísmica en años
    periodo: float = 1 # Periodo de la actividad sísmica en años
    h: float = 2 # Altura del bloque deslizando
    gamma: float = 2.2 # Peso unitario del bloque deslizando
```

```
@dataclass
class CineticaVolteo:
    theta: float = 90 # Ángulo de inclinación del plano deslizando
    alfa: float = 3 # Parámetro de la distribución beta
    beta: float = 3 # Parámetro de la distribución beta
    t_retorno: float = 475 # Tiempo de retorno de la actividad sísmica en años
    periodo: float = 1 # Periodo de la actividad sísmica en años
```

```
@dataclass
class CineticaCunnia:
    alfa: float = 3 # Parámetro de la distribución beta
```

```

beta: float = 3 # Parámetro de la distribución beta
t_retorno: float = 475 # Tiempo de retorno de la actividad sísmica en años
periodo: float = 1 # Periodo de la actividad sísmica en años
n_phi: float = 0.02
n_c: float = 0.05
plano1: Plano = Plano(4.5, 5, 5, 28)
plano2: Plano = Plano(4.5, 5, 5, 28)

```

```
@dataclass
```

```
class DataCtl:
```

```

    red_polar: pd.DataFrame = du.create_red_polar()
    datos_rumbos: pd.DataFrame = None
    datos_friccion: pd.DataFrame = None
    datos_cohesion: pd.DataFrame = None
    cinemat_planar: Cinematica = Cinematica()
    cinemat_volteo: Cinematica = Cinematica()
    cinemat_cunnia: Cinematica = Cinematica()
    cinetic_planar: CineticaPlanar = CineticaPlanar()
    cinetic_volteo: CineticaVolteo = CineticaVolteo()
    cinetic_cunnia: CineticaCunnia = CineticaCunnia()
    sector_planar: Zonificacion = Zonificacion()
    sector_volteo: Zonificacion = Zonificacion()
    sector_cunnia: Zonificacion = Zonificacion()

    # alfa_talud: float = 315 # Azimut del talud
    # beta_a: float = 3 # Parámetro a de la distribución Beta
    # beta_b: float = 3 # Parámetro b de la distribución Beta
    # h: float = 2 #
    # theta: float = 90 # Inclinación (Buzamiento) del talud
    # gamma: float = 2.2 #
    # t_retorno: float = 475 # Tiempo de retorno en años
    # periodo: float = 1 # Periodo en años
    # n_phi: float = 0.02
    # n_c: float = 0.05
    # plano_1: Plano = Plano(4.5, 5, 5, 28)
    # plano_2: Plano = Plano(4.5, 5, 5, 28)

```

```
@dataclass
```

```
class Labels:
```

```

    titulo: str = ''
    titulo_2: str = ''
    x_label: str = ''
    y_label: str = ''
    z_label: str = ''

```

```
@dataclass
```

```
class DataCinematicaFamilias:
```

```

    """
    - probabilidad: float\n
    - max_level: float\n
    - env_df: pd.DataFrame\n

```

```
- paleom_df: pd.DataFrame\n- idx: int\n""\nprobabilidad: float = 0.0\nmax_level: float = 0.0\nenv_df: pd.DataFrame = None\npaleom_df: pd.DataFrame = None\nidx: int = 0\nlabels: Labels = Labels()
```

```
@dataclass\nclass DataCinematica:\n    friccion_df: pd.DataFrame = None\n    calculos_familia: List[DataCinematicaFamilias] = None
```

```
@dataclass\nclass DataCinética:\n    axes_df: pd.DataFrame = None\n    axes2_df: pd.DataFrame = None\n    red_df: pd.DataFrame = None\n    probabilidad: float = 0.0\n    eje1: str = ''\n    eje2: str = ''\n    eje_z1: str = ''\n    eje_z2: str = ''\n    labels: Labels = Labels()
```

```
@dataclass\nclass ZonifPlanar:\n    repr_df: pd.DataFrame = None
```

```
@dataclass\nclass ZonifCunnia:\n    familias: pd.DataFrame = None\n    intersec: pd.DataFrame = None
```

```
@dataclass\nclass DataOut:\n    cinematica_planar: DataCinematica = DataCinematica()\n    cinematica_volteo: DataCinematica = DataCinematica()\n    cinematica_cunia: DataCinematica = DataCinematica()\n    cinetica_desliz: DataCinética = DataCinética()\n    cinetica_volteo: DataCinética = DataCinética()\n    cinetica_cunia_friccion: DataCinética = DataCinética()\n    cinetica_cunia_cohesion: DataCinética = DataCinética()\n    zonif_planar: ZonifPlanar = ZonifPlanar()\n    zonif_volteo: ZonifPlanar = ZonifPlanar()\n    zonif_cunnia: ZonifCunnia = ZonifCunnia()
```

```

app/worker.py
from dataclasses import dataclass
import sys
import traceback

from PySide2 import QtCore as Qtc

import utils_cinematico as cinematica
import utils_cinetico as cinetica
import utils_zonif as zonifica
from app.controlador_datos import DataCtl, DataOut

class WorkerSignals(Qtc.QObject):
    finished = Qtc.Signal(int)
    error = Qtc.Signal(tuple)
    progress = Qtc.Signal(int)

class Worker(Qtc.QRunnable):
    @dataclass(frozen=True)
    class MTDS:
        cinematica_planar = 0
        cinematica_volteo = 1
        cinematica_cunia = 2
        cinetica_deslizante = 3
        cinetica_volteo = 4
        cinetica_cunia = 5
        zonif_planar = 6
        zonif_volteo = 7
        zonif_cunnia = 8

    def __init__(self, datos: DataCtl, datos_out: DataOut, metodo: int):
        super(Worker, self).__init__()
        self._datos = datos
        self._datos_out = datos_out
        self._metodo = metodo
        self.signals = WorkerSignals()

    @Qtc.Slot()
    def run(self):
        try:
            if self._metodo == self.MTDS.cinematica_planar:
                cinematica.cinematica_planar(self._datos,
                self._datos_out.cinematica_planar)
            elif self._metodo == self.MTDS.cinematica_volteo:
                cinematica.cinematica_volteo(self._datos,
                self._datos_out.cinematica_volteo)
            elif self._metodo == self.MTDS.cinematica_cunia:
                cinematica.cinematica_cunia(self._datos,
                self._datos_out.cinematica_cunia)
            elif self._metodo == self.MTDS.cinetica_deslizante:
                cinetica.cinetica_planar(self._datos,
                self._datos_out.cinetica_desliz)

```

```

        elif self._metodo == self.MTDS.cinetica_volteo:
            cinetica.cinetica_volteo(self._datos,
self._datos_out.cinetica_volteo)
        elif self._metodo == self.MTDS.cinetica_cunia:
            cinetica.cinetica_cunia(self._datos, self._datos_out)
        elif self._metodo == self.MTDS.zonif_planar:
            zonifica.zonif_planar(self._datos, self._datos_out)
        elif self._metodo == self.MTDS.zonif_volteo:
            zonifica.zonif_volteo(self._datos, self._datos_out)
        elif self._metodo == self.MTDS.zonif_cunnia:
            zonifica.zonif_cunnia(self._datos, self._datos_out)
    except:
        traceback.print_exc()
        exctype, value = sys.exc_info()[:2]
        self.signals.error.emit((exctype, value, traceback.format_exc()))
    finally:
        self.signals.finished.emit(self._metodo)
        print('[Worker] Tarea terminada')

```

app/dlg_progreso/controller.py

```
from PySide2 import QtWidgets as Qtw
```

```
from .ui_dialogo_progreso import Ui_Dialog
```

```
class ProgressWindow(Qtw.QDialog):
```

```

    def __init__(self, parent: Qtw.QWidget, minimum, maximum, message=''):
        super(ProgressWindow, self).__init__(parent)
        self.ui = Ui_Dialog()
        self.ui.setupUi(self)
        self.setFixedSize(self.size())
        self.ui.progressBar.setMinimum(minimum)
        self.ui.progressBar.setMaximum(maximum)
        self.update_progress(10, message)

```

```

    def update_progress(self, value, message):
        self.ui.progressBar.setValue(value)
        self.ui.label.setText(message)

```

```

    def close(self) -> bool:
        return super(ProgressWindow, self).close()

```

app/dlg_progreso/ui_dialogo_progreso.py

```
# -*- coding: utf-8 -*-
```

```

#####
## Form generated from reading UI file 'dialogo_progresoeycIhS.ui'
##
## Created by: Qt User Interface Compiler version 5.15.2
##
## WARNING! All changes made in this file will be lost when recompiling UI file!
#####

```

```
from PySide2.QtCore import *
```

```

from PySide2.QtGui import *
from PySide2.QtWidgets import *

class Ui_Dialog(object):
    def setupUi(self, Dialog):
        if not Dialog.setObjectName():
            Dialog.setObjectName(u"Dialog")
        Dialog.setWindowModality(Qt.WindowModal)
        Dialog.resize(455, 70)
        sizePolicy = QSizePolicy(QSizePolicy.Fixed, QSizePolicy.Fixed)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(Dialog.sizePolicy().hasHeightForWidth())
        Dialog.setSizePolicy(sizePolicy)
        Dialog.setBaseSize(QSize(1000, 200))
        Dialog.setSizeGripEnabled(False)
        Dialog.setModal(True)
        self.verticalLayout = QVBoxLayout(Dialog)
        self.verticalLayout.setObjectName(u"verticalLayout")
        self.verticalLayout.setContentsMargins(9, 9, 9, 9)
        self.progressBar = QProgressBar(Dialog)
        self.progressBar.setObjectName(u"progressBar")
        self.progressBar.setEnabled(True)
        sizePolicy1 = QSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)
        sizePolicy1.setHorizontalStretch(0)
        sizePolicy1.setVerticalStretch(0)

        sizePolicy1.setHeightForWidth(self.progressBar.sizePolicy().hasHeightForWidth())
        self.progressBar.setSizePolicy(sizePolicy1)
        self.progressBar.setValue(0)
        self.progressBar.setAlignment(Qt.AlignCenter)

        self.verticalLayout.addWidget(self.progressBar)

        self.label = QLabel(Dialog)
        self.label.setObjectName(u"label")
        self.label.setAlignment(Qt.AlignCenter)

        self.verticalLayout.addWidget(self.label)

        self.retranslateUi(Dialog)

        QMetaObject.connectSlotsByName(Dialog)
    # setupUi

    def retranslateUi(self, Dialog):
        Dialog.setWindowTitle(QCoreApplication.translate("Dialog",
u"Procesando...", None))
        self.label.setText(QCoreApplication.translate("Dialog", u"TextLabel",
None))
    # retranslateUi

```

```

app/wnd_data/controller.py
from dataclasses import dataclass

import pandas as pd
from PySide2 import QtCore as Qtc
from PySide2 import QtWidgets as Qtw

from .ui_datos import Ui_MainWindow
from app.controlador_datos import DataCtl

class DataWindowCtrl(Qtw.QMainWindow):
    @dataclass(frozen=True)
    class BTNS:
        rumbos: int = 0
        friccion: int = 1
        recalcular: int = 2
        cohesion: int = 3

    @dataclass
    class Banderas:
        datos: bool = False
        friccion: bool = False
        cohesion: bool = False

    def datos_ok(self):
        return self.datos and self.friccion and self.cohesion

# Señales que salen del controlador
sgn_btn_pressed = Qtc.Signal(int, arguments='btn_pressed')
sgn_dlg_result = Qtc.Signal(int, str, arguments='dialogo, path')
sgn_var_changed = Qtc.Signal(str)
sgn_enable_buttons = Qtc.Signal()

def __init__(self, datos_iniciales: DataCtl):
    super(DataWindowCtrl, self).__init__()
    self.ui = Ui_MainWindow()
    self.ui.setupUi(self)
    self.ruta_datos = 'data/datos.csv'
    self.ruta_friccion = 'data/phi.csv'
    self.ruta_cohesion = 'data/cohesion.csv'
    self.datos = datos_iniciales
    self.banderas = self.Banderas()

    self.ui.push_button_rumbos.clicked.connect(self.cargar_rumbos)
    self.ui.push_button_friccion.clicked.connect(self.cargar_friccion)
    self.ui.push_button_cohesion.clicked.connect(self.cargar_cohesion)
    self.ui.push_button_recalcular.clicked.connect(self.recalcular)

# Cargar valores iniciales de spinners
self.spinners = (
    (self.ui.spin_box_sector_planar_fric, 'sector_planar',
    'ang_friccion'),
    (self.ui.spin_box_sector_volt_fric, 'sector_volteo',

```



```

'ang_friccion'),
    (self.ui.spin_box_sector_cun_fric, 'sector_cunnia', 'ang_friccion'),

    (self.ui.spin_box_cinem_planar_orientacion, 'cinemat_planar',
'ang_talud'),
    (self.ui.spin_box_cinem_volt_orientacion, 'cinemat_volteo',
'ang_talud'),
    (self.ui.spin_box_cinem_cun_orientacion, 'cinemat_cunnia',
'ang_talud'),

    (self.ui.spin_box_cine_plan_inclinacion, 'cinetic_planar', 'theta'),
    (self.ui.spin_box_cine_plan_alfa, 'cinetic_planar', 'alfa'),
    (self.ui.spin_box_cine_plan_beta, 'cinetic_planar', 'beta'),
    (self.ui.spin_box_cine_plan_retorno, 'cinetic_planar', 't_retorno'),
    (self.ui.spin_box_cine_plan_periodo, 'cinetic_planar', 'periodo'),
    (self.ui.spin_box_cine_plan_h, 'cinetic_planar', 'h'),
    (self.ui.spin_box_cine_plan_gamma, 'cinetic_planar', 'gamma'),

    (self.ui.spin_box_cine_volt_inclinacion, 'cinetic_volteo', 'theta'),
    (self.ui.spin_box_cine_volt_alfa, 'cinetic_volteo', 'alfa'),
    (self.ui.spin_box_cine_volt_beta, 'cinetic_volteo', 'beta'),
    (self.ui.spin_box_cine_volt_retorno, 'cinetic_volteo', 't_retorno'),
    (self.ui.spin_box_cine_volt_periodo, 'cinetic_volteo', 'periodo'),

    (self.ui.spin_box_cine_cun_nphi, 'cinetic_cunnia', 'n_phi'),
    (self.ui.spin_box_cine_cun_nc, 'cinetic_cunnia', 'n_c'),
    (self.ui.spin_box_cine_cun_alfa, 'cinetic_cunnia', 'alfa'),
    (self.ui.spin_box_cine_cun_beta, 'cinetic_cunnia', 'beta'),
    (self.ui.spin_box_cine_cun_retorno, 'cinetic_cunnia', 't_retorno'),
    (self.ui.spin_box_cine_cun_periodo, 'cinetic_cunnia', 'periodo'),
    (self.ui.spin_box_cine_cun_p1c, 'cinetic_cunnia', 'plano1', 'c'),
    (self.ui.spin_box_cine_cun_p1A, 'cinetic_cunnia', 'plano1', 'A'),
    (self.ui.spin_box_cine_cun_p1N, 'cinetic_cunnia', 'plano1', 'N'),
    (self.ui.spin_box_cine_cun_p1phi, 'cinetic_cunnia', 'plano1',
'phi'),

    (self.ui.spin_box_cine_cun_p2c, 'cinetic_cunnia', 'plano2', 'c'),
    (self.ui.spin_box_cine_cun_p2A, 'cinetic_cunnia', 'plano2', 'A'),
    (self.ui.spin_box_cine_cun_p2N, 'cinetic_cunnia', 'plano2', 'N'),
    (self.ui.spin_box_cine_cun_p2phi, 'cinetic_cunnia', 'plano2',
'phi'),
)

for i in self.spinners:
    self.asegnar_valor_inicial(i)
    self.conectar_evento(i)

def conectar_evento(self, par: tuple):
    par[0].editingFinished.connect(lambda: self.cambiado(par))

def asignar_valor_inicial(self, par: tuple):
    test_obj = getattr(self.datos, par[1])
    if len(par) == 3:
        par[0].setValue(getattr(test_obj, par[2]))
    elif len(par) == 4:

```

```

        plano = getattr(test_obj, par[2])
        par[0].setValue(getattr(plano, par[3]))

def cambiado(self, par: tuple):
    valor_actual = par[0].value()
    test_obj = getattr(self.datos, par[1])
    plano = None
    if len(par) == 3:
        valor = getattr(test_obj, par[2])
    else:
        plano = getattr(test_obj, par[2])
        valor = getattr(plano, par[3])

    if valor_actual != valor:
        print(f'Valor de {par[0].objectName()} cambiado de {valor} a
{valor_actual}')
        if plano is None:
            setattr(test_obj, par[2], valor_actual)
        else:
            setattr(plano, par[3], valor_actual)

def recalcular(self):
    self.sgn_btn_pressed.emit(self.BTNS.recalcular)
    self.ui.push_button_recalcular.setEnabled(False)

def data_changed(self):
    self.ui.push_button_recalcular.setEnabled(True)

def cargar_rumbos(self):
    self.mensaje('Cargando rumbos...')
    dialogo = Qtw.QFileDialog(self, 'Cargar datos de rumbo y buzamiento',
self.ruta_datos, '*.csv')
    dialogo.setFileMode(Qtw.QFileDialog.ExistingFile)
    if dialogo.exec():
        self.banderas.datos = True
        self.ruta_datos = dialogo.selectedFiles()[0]
        self.sgn_dlg_result.emit(self.BTNS.rumbos, self.ruta_datos)
        self.habilitar_botones()

def cargar_friccion(self):
    self.mensaje('Cargando datos fricción...')
    dialogo = Qtw.QFileDialog(self, 'Cargar datos de fricción',
self.ruta_friccion, '*.csv')
    dialogo.setFileMode(Qtw.QFileDialog.ExistingFile)
    if dialogo.exec():
        self.banderas.friccion = True
        self.ruta_friccion = dialogo.selectedFiles()[0]
        self.sgn_dlg_result.emit(self.BTNS.friccion, self.ruta_friccion)
        self.habilitar_botones()

def cargar_cohesion(self):
    self.mensaje('Cargando datos de cohesión...')
    dialogo = Qtw.QFileDialog(self, 'Cargar datos de cohesión',
self.ruta_cohesion, '*.csv')

```

```

dialogo.setFileMode(QtWidgets.QFileDialog.ExistingFile)
if dialogo.exec():
    self.banderas.cohesion = True
    self.ruta_cohesion = dialogo.selectedFiles()[0]
    self.sgn_dlg_result.emit(self.BTNS.cohesion, self.ruta_cohesion)
    self.habilitar_botones()

def cargar_tabla_datos(self, datos: pd.DataFrame):
    columnas = list(datos.keys())
    self.cargar_items(self.ui.table_widget_rumbos, columnas, datos)

def cargar_tabla_friccion(self, datos):
    columnas = list(datos.keys())
    self.cargar_items(self.ui.table_widget_friccion, columnas, datos)

def cargar_tabla_cohesion(self, datos):
    columnas = list(datos.keys())
    self.cargar_items(self.ui.table_widget_cohesion, columnas, datos)

def cargar_items(self, tabla: QtWidgets.QTableWidget, columnas_names: list, datos:
pd.DataFrame):
    tabla.setColumnCount(len(columnas_names))
    tabla.setRowCount(len(datos))
    tabla.setHorizontalHeaderLabels(columnas_names)

    for idx, row in datos.iterrows():
        column = 0
        for key in columnas_names:
            tabla.setItem(idx, column, QtWidgets.QTableWidgetItem(str(row[key])))
            column = column + 1

def habilitar_botones(self):
    if self.banderas.datos_ok():
        self.sgn_enable_buttons.emit()

@staticmethod
def mensaje(mensaje: str):
    print('[Data window]', mensaje)

```

app/wnd_data/ui_datos.py

```
# -*- coding: utf-8 -*-
```

```

#####
#####
## Form generated from reading UI file 'datosiguHsV.ui'
##
## Created by: Qt User Interface Compiler version 5.15.2
##
## WARNING! All changes made in this file will be lost when recompiling
UI file!
#####
#####

```

```

from PySide2.QtCore import *
from PySide2.QtGui import *

```

```

from PySide2.QtWidgets import *

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        if not MainWindow.setObjectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(748, 600)
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.verticalLayout_3 = QVBoxLayout(self.centralwidget)
        self.verticalLayout_3.setObjectName(u"verticalLayout_3")
        self.verticalLayout_3.setContentsMargins(6, 6, 6, 6)
        self.tabWidget_2 = QTabWidget(self.centralwidget)
        self.tabWidget_2.setObjectName(u"tabWidget_2")
        sizePolicy = QSizePolicy(QSizePolicy.Expanding,
QSizePolicy.Preferred)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)

        sizePolicy.setHeightForWidth(self.tabWidget_2.sizePolicy().hasHeightForWidth())

        self.tabWidget_2.setSizePolicy(sizePolicy)
        self.tab = QWidget()
        self.tab.setObjectName(u"tab")
        self.gridLayout_7 = QGridLayout(self.tab)
        self.gridLayout_7.setObjectName(u"gridLayout_7")
        self.gridLayout_7.setContentsMargins(0, 3, 0, 0)
        self.tabWidget_3 = QTabWidget(self.tab)
        self.tabWidget_3.setObjectName(u"tabWidget_3")
        self.tab_6 = QWidget()
        self.tab_6.setObjectName(u"tab_6")
        self.gridLayout_8 = QGridLayout(self.tab_6)
        self.gridLayout_8.setObjectName(u"gridLayout_8")
        self.label_22 = QLabel(self.tab_6)
        self.label_22.setObjectName(u"label_22")

        self.gridLayout_8.addWidget(self.label_22, 0, 0, 1, 1)

        self.spin_box_sector_planar_fric = QDoubleSpinBox(self.tab_6)

self.spin_box_sector_planar_fric.setObjectName(u"spin_box_sector_planar_fric")
        self.spin_box_sector_planar_fric.setMaximum(90.00000000000000)

        self.gridLayout_8.addWidget(self.spin_box_sector_planar_fric, 0,
1, 1, 1)

        self.horizontalSpacer_8 = QSpacerItem(469, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum)

        self.gridLayout_8.addItem(self.horizontalSpacer_8, 0, 2, 1, 1)

        self.verticalSpacer_7 = QSpacerItem(20, 127, QSizePolicy.Minimum,
QSizePolicy.Expanding)

```

```

self.gridLayout_8.addItem(self.verticalSpacer_7, 1, 0, 1, 1)

self.tabWidget_3.addTab(self.tab_6, "")
self.tab_4 = QWidget()
self.tab_4.setObjectName(u"tab_4")
self.gridLayout_9 = QGridLayout(self.tab_4)
self.gridLayout_9.setObjectName(u"gridLayout_9")
self.label_23 = QLabel(self.tab_4)
self.label_23.setObjectName(u"label_23")

self.gridLayout_9.addWidget(self.label_23, 0, 0, 1, 1)

self.spin_box_sector_volt_fric = QDoubleSpinBox(self.tab_4)

self.spin_box_sector_volt_fric.setObjectName(u"spin_box_sector_volt_fric"
)
self.spin_box_sector_volt_fric.setMaximum(90.00000000000000)

self.gridLayout_9.addWidget(self.spin_box_sector_volt_fric, 0, 1,
1, 1)

self.horizontalSpacer_9 = QSpacerItem(469, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum)

self.gridLayout_9.addItem(self.horizontalSpacer_9, 0, 2, 1, 1)

self.verticalSpacer_8 = QSpacerItem(20, 127, QSizePolicy.Minimum,
QSizePolicy.Expanding)

self.gridLayout_9.addItem(self.verticalSpacer_8, 1, 0, 1, 1)

self.tabWidget_3.addTab(self.tab_4, "")
self.tab_5 = QWidget()
self.tab_5.setObjectName(u"tab_5")
self.gridLayout_10 = QGridLayout(self.tab_5)
self.gridLayout_10.setObjectName(u"gridLayout_10")
self.label_24 = QLabel(self.tab_5)
self.label_24.setObjectName(u"label_24")

self.gridLayout_10.addWidget(self.label_24, 0, 0, 1, 1)

self.spin_box_sector_cun_fric = QDoubleSpinBox(self.tab_5)

self.spin_box_sector_cun_fric.setObjectName(u"spin_box_sector_cun_fric")
self.spin_box_sector_cun_fric.setMaximum(90.00000000000000)

self.gridLayout_10.addWidget(self.spin_box_sector_cun_fric, 0, 1,
1, 1)

self.horizontalSpacer_10 = QSpacerItem(469, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum)

self.gridLayout_10.addItem(self.horizontalSpacer_10, 0, 2, 1, 1)

self.verticalSpacer_9 = QSpacerItem(20, 127, QSizePolicy.Minimum,
QSizePolicy.Expanding)

```

```

self.gridLayout_10.addItem(self.verticalSpacer_9, 1, 0, 1, 1)

self.tabWidget_3.addTab(self.tab_5, "")

self.gridLayout_7.addWidget(self.tabWidget_3, 1, 0, 1, 1)

self.tabWidget_2.addTab(self.tab, "")
self.tab_3 = QWidget()
self.tab_3.setObjectName(u"tab_3")
self.verticalLayout_2 = QVBoxLayout(self.tab_3)
self.verticalLayout_2.setObjectName(u"verticalLayout_2")
self.verticalLayout_2.setContentsMargins(0, 3, 0, 0)
self.tabWidget_4 = QTabWidget(self.tab_3)
self.tabWidget_4.setObjectName(u"tabWidget_4")
self.tab_7 = QWidget()
self.tab_7.setObjectName(u"tab_7")
self.gridLayout_12 = QGridLayout(self.tab_7)
self.gridLayout_12.setObjectName(u"gridLayout_12")
self.label_20 = QLabel(self.tab_7)
self.label_20.setObjectName(u"label_20")

self.gridLayout_12.addWidget(self.label_20, 0, 0, 1, 1)

self.spin_box_cinem_planar_orientacion =
QDoubleSpinBox(self.tab_7)

self.spin_box_cinem_planar_orientacion.setObjectName(u"spin_box_cinem_pla
nar_orientacion")

self.spin_box_cinem_planar_orientacion.setMaximum(359.00000000000000)

self.gridLayout_12.addWidget(self.spin_box_cinem_planar_orientacion, 0,
1, 1, 1)

self.verticalSpacer_4 = QSpacerItem(20, 40, QSizePolicy.Minimum,
QSizePolicy.Expanding)

self.gridLayout_12.addItem(self.verticalSpacer_4, 1, 0, 1, 1)

self.horizontalSpacer_5 = QSpacerItem(40, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum)

self.gridLayout_12.addItem(self.horizontalSpacer_5, 0, 2, 1, 1)

self.tabWidget_4.addTab(self.tab_7, "")
self.tab_8 = QWidget()
self.tab_8.setObjectName(u"tab_8")
self.gridLayout_13 = QGridLayout(self.tab_8)
self.gridLayout_13.setObjectName(u"gridLayout_13")
self.label_3 = QLabel(self.tab_8)
self.label_3.setObjectName(u"label_3")

self.gridLayout_13.addWidget(self.label_3, 2, 0, 1, 1)

```

```

        self.spin_box_cinem_volt_orientacion = QDoubleSpinBox(self.tab_8)

self.spin_box_cinem_volt_orientacion.setObjectName(u"spin_box_cinem_volt_
orientacion")

self.spin_box_cinem_volt_orientacion.setMaximum(359.0000000000000000)

self.gridLayout_13.addWidget(self.spin_box_cinem_volt_orientacion, 2, 1,
1, 1)

        self.verticalSpacer_5 = QSpacerItem(20, 40, QSizePolicy.Minimum,
QSizePolicy.Expanding)

        self.gridLayout_13.addItem(self.verticalSpacer_5, 3, 0, 1, 1)

        self.horizontalSpacer_6 = QSpacerItem(40, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum)

        self.gridLayout_13.addItem(self.horizontalSpacer_6, 2, 2, 1, 1)

self.tabWidget_4.addTab(self.tab_8, "")
self.tab_11 = QWidget()
self.tab_11.setObjectName(u"tab_11")
self.gridLayout_14 = QGridLayout(self.tab_11)
self.gridLayout_14.setObjectName(u"gridLayout_14")
self.label_21 = QLabel(self.tab_11)
self.label_21.setObjectName(u"label_21")

self.gridLayout_14.addWidget(self.label_21, 0, 0, 1, 1)

        self.spin_box_cinem_cun_orientacion = QDoubleSpinBox(self.tab_11)

self.spin_box_cinem_cun_orientacion.setObjectName(u"spin_box_cinem_cun_or
ientacion")

self.spin_box_cinem_cun_orientacion.setMaximum(359.0000000000000000)

        self.gridLayout_14.addWidget(self.spin_box_cinem_cun_orientacion,
0, 1, 1, 1)

        self.horizontalSpacer_7 = QSpacerItem(40, 20,
QSizePolicy.Expanding, QSizePolicy.Minimum)

        self.gridLayout_14.addItem(self.horizontalSpacer_7, 0, 2, 1, 1)

        self.verticalSpacer_6 = QSpacerItem(20, 40, QSizePolicy.Minimum,
QSizePolicy.Expanding)

        self.gridLayout_14.addItem(self.verticalSpacer_6, 1, 0, 1, 1)

self.tabWidget_4.addTab(self.tab_11, "")

self.verticalLayout_2.addWidget(self.tabWidget_4)

self.tabWidget_2.addTab(self.tab_3, "")

```

```

self.tab_2 = QWidget()
self.tab_2.setObjectName(u"tab_2")
self.verticalLayout = QVBoxLayout(self.tab_2)
self.verticalLayout.setSpacing(3)
self.verticalLayout.setObjectName(u"verticalLayout")
self.verticalLayout.setContentsMargins(0, 3, 0, 0)
self.tabWidget_5 = QTabWidget(self.tab_2)
self.tabWidget_5.setObjectName(u"tabWidget_5")
self.tab_9 = QWidget()
self.tab_9.setObjectName(u"tab_9")
self.gridLayout_11 = QGridLayout(self.tab_9)
self.gridLayout_11.setObjectName(u"gridLayout_11")
self.gridLayout_11.setContentsMargins(9, 9, 9, 9)
self.groupBox_5 = QGroupBox(self.tab_9)
self.groupBox_5.setObjectName(u"groupBox_5")
self.horizontalLayout = QHBoxLayout(self.groupBox_5)
self.horizontalLayout.setObjectName(u"horizontalLayout")
self.label = QLabel(self.groupBox_5)
self.label.setObjectName(u"label")

self.horizontalLayout.addWidget(self.label)

self.spin_box_cine_plan_inclinacion =
QDoubleSpinBox(self.groupBox_5)

self.spin_box_cine_plan_inclinacion.setObjectName(u"spin_box_cine_plan_in
clinacion")

self.spin_box_cine_plan_inclinacion.setMaximum(90.000000000000000)

self.horizontalLayout.addWidget(self.spin_box_cine_plan_inclinacion)

self.gridLayout_11.addWidget(self.groupBox_5, 0, 0, 1, 1)

self.groupBox_4 = QGroupBox(self.tab_9)
self.groupBox_4.setObjectName(u"groupBox_4")
self.horizontalLayout_3 = QHBoxLayout(self.groupBox_4)
self.horizontalLayout_3.setObjectName(u"horizontalLayout_3")
self.horizontalLayout_3.setContentsMargins(-1, 6, 6, 6)
self.label_9 = QLabel(self.groupBox_4)
self.label_9.setObjectName(u"label_9")

self.horizontalLayout_3.addWidget(self.label_9)

self.spin_box_cine_plan_retorno = QDoubleSpinBox(self.groupBox_4)

self.spin_box_cine_plan_retorno.setObjectName(u"spin_box_cine_plan_retorn
o")

self.spin_box_cine_plan_retorno.setMaximum(1000000.000000000000000)

self.horizontalLayout_3.addWidget(self.spin_box_cine_plan_retorno)

```



```

self.label_10 = QLabel(self.groupBox_4)
self.label_10.setObjectName(u"label_10")

self.horizontalLayout_3.addWidget(self.label_10)

self.spin_box_cine_plan_periodo = QDoubleSpinBox(self.groupBox_4)
self.spin_box_cine_plan_periodo.setObjectName(u"spin_box_cine_plan_peri
o")

self.spin_box_cine_plan_periodo.setMaximum(100000.0000000000000000)

self.horizontalLayout_3.addWidget(self.spin_box_cine_plan_periodo)

self.gridLayout_11.addWidget(self.groupBox_4, 2, 0, 1, 1)

self.groupBox_2 = QGroupBox(self.tab_9)
self.groupBox_2.setObjectName(u"groupBox_2")
self.horizontalLayout_4 = QHBoxLayout(self.groupBox_2)
self.horizontalLayout_4.setObjectName(u"horizontalLayout_4")
self.horizontalLayout_4.setContentsMargins(9, 9, 9, 9)
self.label_5 = QLabel(self.groupBox_2)
self.label_5.setObjectName(u"label_5")

self.horizontalLayout_4.addWidget(self.label_5)

self.spin_box_cine_plan_alfa = QDoubleSpinBox(self.groupBox_2)
self.spin_box_cine_plan_alfa.setObjectName(u"spin_box_cine_plan_alfa")
self.spin_box_cine_plan_alfa.setMaximum(100.0000000000000000)

self.horizontalLayout_4.addWidget(self.spin_box_cine_plan_alfa)

self.label_6 = QLabel(self.groupBox_2)
self.label_6.setObjectName(u"label_6")

self.horizontalLayout_4.addWidget(self.label_6)

self.spin_box_cine_plan_beta = QDoubleSpinBox(self.groupBox_2)
self.spin_box_cine_plan_beta.setObjectName(u"spin_box_cine_plan_beta")
self.spin_box_cine_plan_beta.setMaximum(100.0000000000000000)

self.horizontalLayout_4.addWidget(self.spin_box_cine_plan_beta)

self.gridLayout_11.addWidget(self.groupBox_2, 0, 1, 1, 1)

self.groupBox_3 = QGroupBox(self.tab_9)
self.groupBox_3.setObjectName(u"groupBox_3")
self.horizontalLayout_2 = QHBoxLayout(self.groupBox_3)
self.horizontalLayout_2.setObjectName(u"horizontalLayout_2")
self.horizontalLayout_2.setContentsMargins(9, 9, 9, 9)
self.label_7 = QLabel(self.groupBox_3)

```

```

self.label_7.setObjectName(u"label_7")

self.horizontalLayout_2.addWidget(self.label_7)

self.spin_box_cine_plan_h = QDoubleSpinBox(self.groupBox_3)
self.spin_box_cine_plan_h.setObjectName(u"spin_box_cine_plan_h")
self.spin_box_cine_plan_h.setMaximum(1000.0000000000000000)

self.horizontalLayout_2.addWidget(self.spin_box_cine_plan_h)

self.label_18 = QLabel(self.groupBox_3)
self.label_18.setObjectName(u"label_18")

self.horizontalLayout_2.addWidget(self.label_18)

self.spin_box_cine_plan_gamma = QDoubleSpinBox(self.groupBox_3)

self.spin_box_cine_plan_gamma.setObjectName(u"spin_box_cine_plan_gamma")
self.spin_box_cine_plan_gamma.setMaximum(1000.0000000000000000)

self.horizontalLayout_2.addWidget(self.spin_box_cine_plan_gamma)

self.gridLayout_11.addWidget(self.groupBox_3, 2, 1, 1, 1)

self.tabWidget_5.addTab(self.tab_9, "")
self.tab_10 = QWidget()
self.tab_10.setObjectName(u"tab_10")
self.gridLayout_15 = QGridLayout(self.tab_10)
self.gridLayout_15.setObjectName(u"gridLayout_15")
self.groupBox_7 = QGroupBox(self.tab_10)
self.groupBox_7.setObjectName(u"groupBox_7")
self.horizontalLayout_6 = QHBoxLayout(self.groupBox_7)
self.horizontalLayout_6.setObjectName(u"horizontalLayout_6")
self.label_2 = QLabel(self.groupBox_7)
self.label_2.setObjectName(u"label_2")

self.horizontalLayout_6.addWidget(self.label_2)

self.spin_box_cine_volt_inclinacion =
QDoubleSpinBox(self.groupBox_7)

self.spin_box_cine_volt_inclinacion.setObjectName(u"spin_box_cine_volt_in
clinacion")

self.spin_box_cine_volt_inclinacion.setMaximum(90.0000000000000000)

self.horizontalLayout_6.addWidget(self.spin_box_cine_volt_inclinacion)

self.gridLayout_15.addWidget(self.groupBox_7, 0, 0, 1, 1)

self.groupBox = QGroupBox(self.tab_10)
self.groupBox.setObjectName(u"groupBox")
self.gridLayout = QGridLayout(self.groupBox)

```

```

        self.gridLayout.setObjectName(u"gridLayout")
        self.spin_box_cine_volt_alfa = QDoubleSpinBox(self.groupBox)

self.spin_box_cine_volt_alfa.setObjectName(u"spin_box_cine_volt_alfa")
self.spin_box_cine_volt_alfa.setMaximum(100.00000000000000)

1)

        self.gridLayout.addWidget(self.spin_box_cine_volt_alfa, 0, 1, 1,

self.label_26 = QLabel(self.groupBox)
self.label_26.setObjectName(u"label_26")

self.gridLayout.addWidget(self.label_26, 0, 0, 1, 1)

self.label_27 = QLabel(self.groupBox)
self.label_27.setObjectName(u"label_27")

self.gridLayout.addWidget(self.label_27, 0, 2, 1, 1)

self.spin_box_cine_volt_beta = QDoubleSpinBox(self.groupBox)

self.spin_box_cine_volt_beta.setObjectName(u"spin_box_cine_volt_beta")
self.spin_box_cine_volt_beta.setMaximum(100.00000000000000)

1)

        self.gridLayout.addWidget(self.spin_box_cine_volt_beta, 0, 3, 1,

self.gridLayout_15.addWidget(self.groupBox, 0, 1, 1, 1)

self.groupBox_6 = QGroupBox(self.tab_10)
self.groupBox_6.setObjectName(u"groupBox_6")
self.horizontalLayout_5 = QHBoxLayout(self.groupBox_6)
self.horizontalLayout_5.setObjectName(u"horizontalLayout_5")
self.label_19 = QLabel(self.groupBox_6)
self.label_19.setObjectName(u"label_19")

self.horizontalLayout_5.addWidget(self.label_19)

self.spin_box_cine_volt_retorno = QDoubleSpinBox(self.groupBox_6)

self.spin_box_cine_volt_retorno.setObjectName(u"spin_box_cine_volt_return
o")

self.spin_box_cine_volt_retorno.setMaximum(1000000.0000000000000000)

self.horizontalLayout_5.addWidget(self.spin_box_cine_volt_retorno)

self.label_25 = QLabel(self.groupBox_6)
self.label_25.setObjectName(u"label_25")

self.horizontalLayout_5.addWidget(self.label_25)

self.spin_box_cine_volt_periodo = QDoubleSpinBox(self.groupBox_6)

```

```

self.spin_box_cine_volt_periodo.setObjectName(u"spin_box_cine_volt_periodo")

self.spin_box_cine_volt_periodo.setMaximum(100000.0000000000000000)

self.horizontalLayout_5.addWidget(self.spin_box_cine_volt_periodo)

self.gridLayout_15.addWidget(self.groupBox_6, 2, 0, 1, 1)

self.tabWidget_5.addTab(self.tab_10, "")
self.tab_12 = QWidget()
self.tab_12.setObjectName(u"tab_12")
self.gridLayout_16 = QGridLayout(self.tab_12)
self.gridLayout_16.setObjectName(u"gridLayout_16")
self.groupBox_8 = QGroupBox(self.tab_12)
self.groupBox_8.setObjectName(u"groupBox_8")
self.horizontalLayout_7 = QHBoxLayout(self.groupBox_8)
self.horizontalLayout_7.setObjectName(u"horizontalLayout_7")
self.label_28 = QLabel(self.groupBox_8)
self.label_28.setObjectName(u"label_28")

self.horizontalLayout_7.addWidget(self.label_28)

self.spin_box_cine_cun_alfa = QDoubleSpinBox(self.groupBox_8)

self.spin_box_cine_cun_alfa.setObjectName(u"spin_box_cine_cun_alfa")
self.spin_box_cine_cun_alfa.setMaximum(100.0000000000000000)

self.horizontalLayout_7.addWidget(self.spin_box_cine_cun_alfa)

self.label_29 = QLabel(self.groupBox_8)
self.label_29.setObjectName(u"label_29")

self.horizontalLayout_7.addWidget(self.label_29)

self.spin_box_cine_cun_beta = QDoubleSpinBox(self.groupBox_8)

self.spin_box_cine_cun_beta.setObjectName(u"spin_box_cine_cun_beta")
self.spin_box_cine_cun_beta.setMaximum(100.0000000000000000)

self.horizontalLayout_7.addWidget(self.spin_box_cine_cun_beta)

self.gridLayout_16.addWidget(self.groupBox_8, 0, 2, 1, 1)

self.groupBox_9 = QGroupBox(self.tab_12)
self.groupBox_9.setObjectName(u"groupBox_9")
self.horizontalLayout_8 = QHBoxLayout(self.groupBox_9)
self.horizontalLayout_8.setObjectName(u"horizontalLayout_8")
self.label_11 = QLabel(self.groupBox_9)
self.label_11.setObjectName(u"label_11")

self.horizontalLayout_8.addWidget(self.label_11)

```

```

self.spin_box_cine_cun_nphi = QDoubleSpinBox(self.groupBox_9)
self.spin_box_cine_cun_nphi.setObjectName(u"spin_box_cine_cun_nphi")
self.spin_box_cine_cun_nphi.setMaximum(100.00000000000000)

self.horizontalLayout_8.addWidget(self.spin_box_cine_cun_nphi)

self.label_12 = QLabel(self.groupBox_9)
self.label_12.setObjectName(u"label_12")

self.horizontalLayout_8.addWidget(self.label_12)

self.spin_box_cine_cun_nc = QDoubleSpinBox(self.groupBox_9)
self.spin_box_cine_cun_nc.setObjectName(u"spin_box_cine_cun_nc")
self.spin_box_cine_cun_nc.setMaximum(100.00000000000000)

self.horizontalLayout_8.addWidget(self.spin_box_cine_cun_nc)

self.gridLayout_16.addWidget(self.groupBox_9, 0, 1, 1, 1)

self.groupBox_10 = QGroupBox(self.tab_12)
self.groupBox_10.setObjectName(u"groupBox_10")
self.gridLayout_3 = QGridLayout(self.groupBox_10)
self.gridLayout_3.setObjectName(u"gridLayout_3")
self.label_16 = QLabel(self.groupBox_10)
self.label_16.setObjectName(u"label_16")
sizePolicy1 = QSizePolicy(QSizePolicy.Minimum,
QSizePolicy.Preferred)
sizePolicy1.setHorizontalStretch(0)
sizePolicy1.setVerticalStretch(0)

sizePolicy1.setHeightForWidth(self.label_16.sizePolicy().hasHeightForWidth())
self.label_16.setSizePolicy(sizePolicy1)

self.gridLayout_3.addWidget(self.label_16, 1, 1, 1, 1)

self.spin_box_cine_cun_p1A = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p1A.setObjectName(u"spin_box_cine_cun_p1A")
self.spin_box_cine_cun_p1A.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p1A.setMaximum(1000.00000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p1A, 1, 3, 1,
1)

self.label_17 = QLabel(self.groupBox_10)
self.label_17.setObjectName(u"label_17")

sizePolicy1.setHeightForWidth(self.label_17.sizePolicy().hasHeightForWidth())
self.label_17.setSizePolicy(sizePolicy1)

self.gridLayout_3.addWidget(self.label_17, 2, 1, 1, 1)

```

```

self.label_8 = QLabel(self.groupBox_10)
self.label_8.setObjectName(u"label_8")
self.label_8.setAlignment(Qt.AlignCenter)

self.gridLayout_3.addWidget(self.label_8, 0, 2, 1, 1)

self.label_14 = QLabel(self.groupBox_10)
self.label_14.setObjectName(u"label_14")
self.label_14.setAlignment(Qt.AlignCenter)

self.gridLayout_3.addWidget(self.label_14, 0, 4, 1, 1)

self.spin_box_cine_cun_p2A = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p2A.setObjectName(u"spin_box_cine_cun_p2A")
self.spin_box_cine_cun_p2A.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p2A.setMaximum(100.00000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p2A, 2, 3, 1,
1)

self.spin_box_cine_cun_p2c = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p2c.setObjectName(u"spin_box_cine_cun_p2c")
self.spin_box_cine_cun_p2c.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p2c.setMaximum(100.00000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p2c, 2, 2, 1,
1)

self.label_13 = QLabel(self.groupBox_10)
self.label_13.setObjectName(u"label_13")
self.label_13.setAlignment(Qt.AlignCenter)

self.gridLayout_3.addWidget(self.label_13, 0, 3, 1, 1)

self.spin_box_cine_cun_p1N = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p1N.setObjectName(u"spin_box_cine_cun_p1N")
self.spin_box_cine_cun_p1N.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p1N.setMaximum(1000.00000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p1N, 1, 4, 1,
1)

self.spin_box_cine_cun_p1c = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p1c.setObjectName(u"spin_box_cine_cun_p1c")
self.spin_box_cine_cun_p1c.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p1c.setMaximum(100.00000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p1c, 1, 2, 1,
1)

self.spin_box_cine_cun_p2N = QDoubleSpinBox(self.groupBox_10)

```

```

self.spin_box_cine_cun_p2N.setObjectName(u"spin_box_cine_cun_p2N")
self.spin_box_cine_cun_p2N.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p2N.setMaximum(1000.0000000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p2N, 2, 4, 1,
1)

self.label_15 = QLabel(self.groupBox_10)
self.label_15.setObjectName(u"label_15")
self.label_15.setAlignment(Qt.AlignCenter)

self.gridLayout_3.addWidget(self.label_15, 0, 5, 1, 1)

self.spin_box_cine_cun_p1phi = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p1phi.setObjectName(u"spin_box_cine_cun_p1phi")
self.spin_box_cine_cun_p1phi.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p1phi.setMaximum(360.0000000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p1phi, 1, 5,
1, 1)

self.spin_box_cine_cun_p2phi = QDoubleSpinBox(self.groupBox_10)

self.spin_box_cine_cun_p2phi.setObjectName(u"spin_box_cine_cun_p2phi")
self.spin_box_cine_cun_p2phi.setAlignment(Qt.AlignCenter)
self.spin_box_cine_cun_p2phi.setMaximum(360.0000000000000000)

self.gridLayout_3.addWidget(self.spin_box_cine_cun_p2phi, 2, 5,
1, 1)

self.gridLayout_16.addWidget(self.groupBox_10, 1, 2, 1, 1)

self.groupBox_11 = QGroupBox(self.tab_12)
self.groupBox_11.setObjectName(u"groupBox_11")
self.gridLayout_2 = QGridLayout(self.groupBox_11)
self.gridLayout_2.setObjectName(u"gridLayout_2")
self.label_30 = QLabel(self.groupBox_11)
self.label_30.setObjectName(u"label_30")

self.gridLayout_2.addWidget(self.label_30, 0, 0, 1, 1)

self.spin_box_cine_cun_retorno = QDoubleSpinBox(self.groupBox_11)

self.spin_box_cine_cun_retorno.setObjectName(u"spin_box_cine_cun_retorno"
)

self.spin_box_cine_cun_retorno.setMaximum(1000000.0000000000000000)

self.gridLayout_2.addWidget(self.spin_box_cine_cun_retorno, 0, 1,
1, 1)

self.label_31 = QLabel(self.groupBox_11)
self.label_31.setObjectName(u"label_31")

```

```

self.gridLayout_2.addWidget(self.label_31, 1, 0, 1, 1)

self.spin_box_cine_cun_periodo = QDoubleSpinBox(self.groupBox_11)
self.spin_box_cine_cun_periodo.setObjectName(u"spin_box_cine_cun_periodo"
)
self.spin_box_cine_cun_periodo.setMaximum(100000.0000000000000000)
self.gridLayout_2.addWidget(self.spin_box_cine_cun_periodo, 1, 1,
1, 1)

self.gridLayout_16.addWidget(self.groupBox_11, 1, 1, 1, 1)

self.tabWidget_5.addTab(self.tab_12, "")

self.verticalLayout.addWidget(self.tabWidget_5)

self.tabWidget_2.addTab(self.tab_2, "")

self.verticalLayout_3.addWidget(self.tabWidget_2)

self.tabWidget = QTabWidget(self.centralwidget)
self.tabWidget.setObjectName(u"tabWidget")
self.tab_rumbos = QWidget()
self.tab_rumbos.setObjectName(u"tab_rumbos")
self.gridLayout_4 = QGridLayout(self.tab_rumbos)
self.gridLayout_4.setObjectName(u"gridLayout_4")
self.gridLayout_4.setContentsMargins(0, 0, 0, 0)
self.push_button_rumbos = QPushButton(self.tab_rumbos)
self.push_button_rumbos.setObjectName(u"push_button_rumbos")

self.gridLayout_4.addWidget(self.push_button_rumbos, 0, 1, 1, 1)

self.table_widget_rumbos = QTableWidgetItem(self.tab_rumbos)
if (self.table_widget_rumbos.columnCount() < 4):
    self.table_widget_rumbos.setColumnCount(4)
__qtablewidgetitem = QTableWidgetItem()
self.table_widget_rumbos.setHorizontalHeaderItem(0,
__qtablewidgetitem)
__qtablewidgetitem1 = QTableWidgetItem()
self.table_widget_rumbos.setHorizontalHeaderItem(1,
__qtablewidgetitem1)
__qtablewidgetitem2 = QTableWidgetItem()
self.table_widget_rumbos.setHorizontalHeaderItem(2,
__qtablewidgetitem2)
__qtablewidgetitem3 = QTableWidgetItem()
self.table_widget_rumbos.setHorizontalHeaderItem(3,
__qtablewidgetitem3)
self.table_widget_rumbos.setObjectName(u"table_widget_rumbos")
self.table_widget_rumbos.setEnabled(True)

self.table_widget_rumbos.setEditTriggers(QAbstractItemView.NoEditTriggers
)

self.table_widget_rumbos.setSelectionMode(QAbstractItemView.NoSelection)

```



```

self.table_widget_rumbos.setCornerButtonEnabled(False)

self.gridLayout_4.addWidget(self.table_widget_rumbos, 1, 0, 1, 2)

self.tabWidget.addTab(self.tab_rumbos, "")
self.tab_friccion = QWidget()
self.tab_friccion.setObjectName(u"tab_friccion")
self.gridLayout_5 = QGridLayout(self.tab_friccion)
self.gridLayout_5.setObjectName(u"gridLayout_5")
self.gridLayout_5.setContentsMargins(0, 0, 0, 0)
self.push_button_friccion = QPushButton(self.tab_friccion)
self.push_button_friccion.setObjectName(u"push_button_friccion")

self.gridLayout_5.addWidget(self.push_button_friccion, 0, 1, 1,
1)

self.table_widget_friccion = QTableWidgetItem(self.tab_friccion)
if (self.table_widget_friccion.columnCount() < 1):
    self.table_widget_friccion.setColumnCount(1)
    __qtablewidgetitem4 = QTableWidgetItem()
    self.table_widget_friccion.setHorizontalHeaderItem(0,
__qtablewidgetitem4)

self.table_widget_friccion.setObjectName(u"table_widget_friccion")
self.table_widget_friccion.setEnabled(True)

self.table_widget_friccion.setEditTriggers(QAbstractItemView.NoEditTrigge
rs)

self.table_widget_friccion.setSelectionMode(QAbstractItemView.NoSelection
)

self.table_widget_friccion.setCornerButtonEnabled(False)

self.gridLayout_5.addWidget(self.table_widget_friccion, 1, 0, 1,
2)

self.tabWidget.addTab(self.tab_friccion, "")
self.tab_cohesion = QWidget()
self.tab_cohesion.setObjectName(u"tab_cohesion")
self.gridLayout_6 = QGridLayout(self.tab_cohesion)
self.gridLayout_6.setObjectName(u"gridLayout_6")
self.gridLayout_6.setContentsMargins(0, 0, 0, 0)
self.push_button_cohesion = QPushButton(self.tab_cohesion)
self.push_button_cohesion.setObjectName(u"push_button_cohesion")

self.gridLayout_6.addWidget(self.push_button_cohesion, 0, 1, 1,
1)

self.table_widget_cohesion = QTableWidgetItem(self.tab_cohesion)
if (self.table_widget_cohesion.columnCount() < 1):
    self.table_widget_cohesion.setColumnCount(1)
    __qtablewidgetitem5 = QTableWidgetItem()
    self.table_widget_cohesion.setHorizontalHeaderItem(0,
__qtablewidgetitem5)

self.table_widget_cohesion.setObjectName(u"table_widget_cohesion")

```

```

2) self.gridLayout_6.addWidget(self.table_widget_cohesion, 1, 0, 1,

self.tabWidget.addTab(self.tab_cohesion, "")

self.verticalLayout_3.addWidget(self.tabWidget)

self.push_button_recalcular = QPushButton(self.centralwidget)

self.push_button_recalcular.setObjectName(u"push_button_recalcular")
self.push_button_recalcular.setEnabled(False)
font = QFont()
font.setPointSize(12)
font.setBold(True)
font.setItalic(False)
font.setUnderline(False)
font.setWeight(75)
font.setStrikeOut(False)
font.setKerning(True)
self.push_button_recalcular.setFont(font)

self.verticalLayout_3.addWidget(self.push_button_recalcular)

MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(MainWindow)

self.tabWidget_2.setCurrentIndex(0)
self.tabWidget_3.setCurrentIndex(0)
self.tabWidget_4.setCurrentIndex(0)
self.tabWidget_5.setCurrentIndex(0)
self.tabWidget.setCurrentIndex(0)

QMetaObject.connectSlotsByName(MainWindow)
# setupUi

def retranslateUi(self, MainWindow):

MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow",
u"Datos", None))
self.label_22.setText(QCoreApplication.translate("MainWindow",
u"\u00c1ngulo de fricci\u00f3n (Cono de fricci\u00f3n)", None))
self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_6),
QCoreApplication.translate("MainWindow", u"Planar", None))
self.label_23.setText(QCoreApplication.translate("MainWindow",
u"\u00c1ngulo de fricci\u00f3n (Cono de fricci\u00f3n)", None))
self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_4),
QCoreApplication.translate("MainWindow", u"Volteo", None))
self.label_24.setText(QCoreApplication.translate("MainWindow",
u"\u00c1ngulo de fricci\u00f3n (Cono de fricci\u00f3n)", None))
self.tabWidget_3.setTabText(self.tabWidget_3.indexOf(self.tab_5),
QCoreApplication.translate("MainWindow", u"Cu\u00fla", None))
self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab),
QCoreApplication.translate("MainWindow", u"Sectorizaci\u00f3n

```

```

cinem\u00e9tica", None))
    self.label_20.setText(QCoreApplication.translate("MainWindow",
u"Orientaci\u00f3n de la envolvente (Azimut del talud)", None))

self.spin_box_cinem_planar_orientacion.setSuffix(QCoreApplication.transla
te("MainWindow", u"\u00b0", None))
    self.tabWidget_4.setTabText(self.tabWidget_4.indexOf(self.tab_7),
QCoreApplication.translate("MainWindow", u"Planar", None))
    self.label_3.setText(QCoreApplication.translate("MainWindow",
u"Orientaci\u00f3n de la envolvente (Azimut del talud)", None))

self.spin_box_cinem_volt_orientacion.setSuffix(QCoreApplication.translate
("MainWindow", u"\u00b0", None))
    self.tabWidget_4.setTabText(self.tabWidget_4.indexOf(self.tab_8),
QCoreApplication.translate("MainWindow", u"Volteo", None))
    self.label_21.setText(QCoreApplication.translate("MainWindow",
u"Orientaci\u00f3n de la envolvente (Azimut del talud)", None))

self.spin_box_cinem_cun_orientacion.setSuffix(QCoreApplication.translate(
"MainWindow", u"\u00b0", None))

self.tabWidget_4.setTabText(self.tabWidget_4.indexOf(self.tab_11),
QCoreApplication.translate("MainWindow", u"Cu\u00fa", None))
    self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab_3),
QCoreApplication.translate("MainWindow", u"Probabilidad - An\u00e9lisis
Cinem\u00e9tico", None))
    self.groupBox_5.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e1metros del plano deslizante", None))
    self.label.setText(QCoreApplication.translate("MainWindow",
u"Inclinaci\u00f3n - Theta", None))
    self.groupBox_4.setTitle(QCoreApplication.translate("MainWindow",
u"Amenaza s\u00edsmica", None))
    self.label_9.setText(QCoreApplication.translate("MainWindow",
u"Tiempo de retorno", None))

self.spin_box_cine_plan_retorno.setSuffix(QCoreApplication.translate("Mai
nWindow", u" a\u00f1os", None))
    self.label_10.setText(QCoreApplication.translate("MainWindow",
u"Periodo", None))

self.spin_box_cine_plan_periodo.setSuffix(QCoreApplication.translate("Mai
nWindow", u" a\u00f1os", None))
    self.groupBox_2.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e1metros de la distribuci\u00f3n Beta", None))
    self.label_5.setText(QCoreApplication.translate("MainWindow",
u"Alfa", None))
    self.label_6.setText(QCoreApplication.translate("MainWindow",
u"Beta", None))
    self.groupBox_3.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e1metros del bloque deslizante", None))
    self.label_7.setText(QCoreApplication.translate("MainWindow",
u"Altura - h", None))
    self.label_18.setText(QCoreApplication.translate("MainWindow",
u"Peso unitario - Gama", None))
    self.tabWidget_5.setTabText(self.tabWidget_5.indexOf(self.tab_9),
QCoreApplication.translate("MainWindow", u"Planar", None))

```

```

        self.groupBox_7.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e9lmetros del talud", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow",
u"Inclinaci\u00f3n - theta", None))

self.spin_box_cine_volt_inclinacion.setSuffix(QCoreApplication.translate(
"MainWindow", u"\u00b0", None))
        self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e9lmetros de la distribuci\u00f3n Beta", None))
        self.label_26.setText(QCoreApplication.translate("MainWindow",
u"Alfa", None))
        self.label_27.setText(QCoreApplication.translate("MainWindow",
u"Beta", None))
        self.groupBox_6.setTitle(QCoreApplication.translate("MainWindow",
u"Amenaza s\u00e9dsmica", None))
        self.label_19.setText(QCoreApplication.translate("MainWindow",
u"Tiempo de retorno", None))

self.spin_box_cine_volt_retorno.setSuffix(QCoreApplication.translate("Mai
nWindow", u" a\u00f1os", None))
        self.label_25.setText(QCoreApplication.translate("MainWindow",
u"Periodo", None))

self.spin_box_cine_volt_periodo.setSuffix(QCoreApplication.translate("Mai
nWindow", u" a\u00f1os", None))

self.tabWidget_5.setTabText(self.tabWidget_5.indexOf(self.tab_10),
QCoreApplication.translate("MainWindow", u"Volteo", None))
        self.groupBox_8.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e9lmetros de la distribuci\u00f3n Beta", None))
        self.label_28.setText(QCoreApplication.translate("MainWindow",
u"alfa", None))
        self.label_29.setText(QCoreApplication.translate("MainWindow",
u"beta", None))
        self.groupBox_9.setTitle(QCoreApplication.translate("MainWindow",
u"Constantes", None))
        self.label_11.setText(QCoreApplication.translate("MainWindow",
u"Constante n phi", None))
        self.label_12.setText(QCoreApplication.translate("MainWindow",
u"Constante n c", None))

self.groupBox_10.setTitle(QCoreApplication.translate("MainWindow",
u"Par\u00e9lmetros de planos de cu\u00f1a", None))
        self.label_16.setText(QCoreApplication.translate("MainWindow",
u"Plano 1", None))
        self.label_17.setText(QCoreApplication.translate("MainWindow",
u"Plano 2", None))
        self.label_8.setText(QCoreApplication.translate("MainWindow",
u"Constante C", None))
        self.label_14.setText(QCoreApplication.translate("MainWindow",
u"Normal", None))
        self.label_13.setText(QCoreApplication.translate("MainWindow",
u"\u00c1rea", None))
        self.label_15.setText(QCoreApplication.translate("MainWindow",
u"phi medio", None))

```

```

self.spin_box_cine_cun_p1phi.setSuffix(QCoreApplication.translate("MainWi
ndow", u"\u00b0", None))

self.spin_box_cine_cun_p2phi.setSuffix(QCoreApplication.translate("MainWi
ndow", u"\u00b0", None))

self.groupBox_11.setTitle(QCoreApplication.translate("MainWindow",
u"Amenaza s\u00e9mica", None))
    self.label_30.setText(QCoreApplication.translate("MainWindow",
u"Tiempo de retorno", None))

self.spin_box_cine_cun_retorno.setSuffix(QCoreApplication.translate("Main
Window", u" a\u00f1os", None))
    self.label_31.setText(QCoreApplication.translate("MainWindow",
u"Periodo", None))

self.spin_box_cine_cun_periodo.setSuffix(QCoreApplication.translate("Main
Window", u" a\u00f1os", None))

self.tabWidget_5.setTabText(self.tabWidget_5.indexOf(self.tab_12),
QCoreApplication.translate("MainWindow", u"Cu\u00fla", None))
    self.tabWidget_2.setTabText(self.tabWidget_2.indexOf(self.tab_2),
QCoreApplication.translate("MainWindow", u"Probabilidad - An\u00e9lisis
cin\u00e9tico", None))

self.push_button_rumbos.setText(QCoreApplication.translate("MainWindow",
u"Leer datos desde archivo", None))
    __qtablewidgetitem =
self.table_widget_rumbos.horizontalHeaderItem(0)

__qtablewidgetitem.setText(QCoreApplication.translate("MainWindow",
u"rumbo_ang", None));
    __qtablewidgetitem1 =
self.table_widget_rumbos.horizontalHeaderItem(1)

__qtablewidgetitem1.setText(QCoreApplication.translate("MainWindow",
u"rumbo_dir", None));
    __qtablewidgetitem2 =
self.table_widget_rumbos.horizontalHeaderItem(2)

__qtablewidgetitem2.setText(QCoreApplication.translate("MainWindow",
u"buzamiento_ang", None));
    __qtablewidgetitem3 =
self.table_widget_rumbos.horizontalHeaderItem(3)

__qtablewidgetitem3.setText(QCoreApplication.translate("MainWindow",
u"buzamiento_dir", None));

self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_rumbos),
QCoreApplication.translate("MainWindow", u"Rumbos y buzamientos", None))

self.push_button_friccion.setText(QCoreApplication.translate("MainWindow"
, u"Leer datos desde archivo", None))
    __qtablewidgetitem4 =
self.table_widget_friccion.horizontalHeaderItem(0)

```

```

__qtablewidgetitem4.setText(QCoreApplication.translate("MainWindow",
u"phi", None));

self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_friccion),
QCoreApplication.translate("MainWindow", u"Fricci\u00f3n", None))

self.push_button_cohesion.setText(QCoreApplication.translate("MainWindow"
, u"Leer datos desde archivo", None))
__qtablewidgetitem5 =
self.table_widget_cohesion.horizontalHeaderItem(0)

__qtablewidgetitem5.setText(QCoreApplication.translate("MainWindow",
u"Cr2", None));

self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_cohesion),
QCoreApplication.translate("MainWindow", u"Cohesi\u00f3n", None))

self.push_button_recalcular.setText(QCoreApplication.translate("MainWindo
w", u"Actualizar datos y calcular", None))
# retranslateUi

```

app/wnd_main/controller.py

```

from dataclasses import dataclass

from PySide2 import QtGui as Qtg
from PySide2 import QtWidgets as Qtw
from PySide2 import QtCore as Qtc

from .ui_Interfaz import Ui_MainWindow

class MainWindowCtr(Qtw.QMainWindow):
    @dataclass
    class BTNS:
        datos = 0
        cinematica_deslizamiento = 1
        cinematica_volteo = 2
        cinematica_cunia = 3
        cinetica_deslizamiento = 4
        cinetica_volteo = 5
        cinetica_cunia = 6
        zonif_planar = 7
        zonif_volteo = 8
        zonif_cunnia = 9

    sgn_btn = Qtc.Signal(int, arguments='pressed_btn')

    def __init__(self):
        super(MainWindowCtr, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.setAttribute(Qtg.Qt.WA_DeleteOnClose, True)

        # Enlazar botones a señales
        self.ui.push_button_data.clicked.connect(lambda:

```

```

self.sgn_btn.emit(self.BTNS.datos))
    self.ui.push_button_cinematica_desliz.clicked.connect(
        lambda: self.sgn_btn.emit(self.BTNS.cinematica_deslizamiento))
    self.ui.push_button_cinematica_volteo.clicked.connect(
        lambda: self.sgn_btn.emit(self.BTNS.cinematica_volteo))
    self.ui.push_button_cinematica_cunia.clicked.connect(
        lambda: self.sgn_btn.emit(self.BTNS.cinematica_cunia))
    self.ui.push_button_cinematica_desliz.clicked.connect(lambda:
self.sgn_btn.emit(self.BTNS.cinematica_deslizamiento))
    self.ui.push_button_cinematica_volteo.clicked.connect(lambda:
self.sgn_btn.emit(self.BTNS.cinematica_volteo))
    self.ui.push_button_cinematica_cunia.clicked.connect(lambda:
self.sgn_btn.emit(self.BTNS.cinematica_cunia))
    self.ui.push_button_zonif_planar.clicked.connect(lambda:
self.sgn_btn.emit(self.BTNS.zonif_planar))
    self.ui.push_button_zonif_volteo.clicked.connect(lambda:
self.sgn_btn.emit(self.BTNS.zonif_volteo))
    self.ui.push_button_zonif_cunna.clicked.connect(lambda:
self.sgn_btn.emit(self.BTNS.zonif_cunna))

    def enable_buttons(self):
        self.ui.push_button_cinematica_cunia.setEnabled(True)
        self.ui.push_button_cinematica_volteo.setEnabled(True)
        self.ui.push_button_cinematica_desliz.setEnabled(True)
        self.ui.push_button_cinematica_desliz.setEnabled(True)
        self.ui.push_button_cinematica_volteo.setEnabled(True)
        self.ui.push_button_cinematica_cunia.setEnabled(True)

```

app/wnd_main/ui_Interfaz.py

```
# -*- coding: utf-8 -*-
```

```

#####
## Form generated from reading UI file 'InterfazqkHkFV.ui'
##
## Created by: Qt User Interface Compiler version 5.15.2
##
## WARNING! All changes made in this file will be lost when recompiling UI file!
#####

```

```

from PySide2.QtCore import *
from PySide2.QtGui import *
from PySide2.QtWidgets import *

```

```

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        if not MainWindow.setObjectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(590, 233)
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.gridLayout = QGridLayout(self.centralwidget)
        self.gridLayout.setSpacing(0)
        self.gridLayout.setObjectName(u"gridLayout")

```

```

self.gridLayout.setContentsMargins(0, 0, 0, 0)
self.frame_2 = QFrame(self.centralwidget)
self.frame_2.setObjectName(u"frame_2")
sizePolicy = QSizePolicy(QSizePolicy.Preferred, QSizePolicy.Expanding)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.frame_2.sizePolicy().hasHeightForWidth())
self.frame_2.setSizePolicy(sizePolicy)
self.frame_2 setFrameShape(QFrame.StyledPanel)
self.frame_2 setFrameShadow(QFrame.Raised)
self.verticalLayout = QVBoxLayout(self.frame_2)
self.verticalLayout.setObjectName(u"verticalLayout")
self.label = QLabel(self.frame_2)
self.label.setObjectName(u"label")

self.verticalLayout.addWidget(self.label)

self.push_button_cinematica_desliz = QPushButton(self.frame_2)

self.push_button_cinematica_desliz.setObjectName(u"push_button_cinematica_desliz")

self.verticalLayout.addWidget(self.push_button_cinematica_desliz)

self.push_button_cinematica_volteo = QPushButton(self.frame_2)

self.push_button_cinematica_volteo.setObjectName(u"push_button_cinematica_volteo")

self.verticalLayout.addWidget(self.push_button_cinematica_volteo)

self.push_button_cinematica_cunia = QPushButton(self.frame_2)

self.push_button_cinematica_cunia.setObjectName(u"push_button_cinematica_cunia")

self.verticalLayout.addWidget(self.push_button_cinematica_cunia)

self.gridLayout.addWidget(self.frame_2, 1, 1, 1, 1, Qt.AlignTop)

self.frame_3 = QFrame(self.centralwidget)
self.frame_3.setObjectName(u"frame_3")

sizePolicy.setHeightForWidth(self.frame_3.sizePolicy().hasHeightForWidth())
self.frame_3.setSizePolicy(sizePolicy)
self.frame_3 setFrameShape(QFrame.StyledPanel)
self.frame_3 setFrameShadow(QFrame.Raised)
self.verticalLayout_2 = QVBoxLayout(self.frame_3)
self.verticalLayout_2.setObjectName(u"verticalLayout_2")
self.label_2 = QLabel(self.frame_3)
self.label_2.setObjectName(u"label_2")

self.verticalLayout_2.addWidget(self.label_2)

```



```

self.push_button_cinetica_desliz = QPushButton(self.frame_3)
self.push_button_cinetica_desliz.setObjectName(u"push_button_cinetica_desliz")
self.verticalLayout_2.addWidget(self.push_button_cinetica_desliz)
self.push_button_cinetica_volteo = QPushButton(self.frame_3)
self.push_button_cinetica_volteo.setObjectName(u"push_button_cinetica_volteo")
self.verticalLayout_2.addWidget(self.push_button_cinetica_volteo)
self.push_button_cinetica_cunia = QPushButton(self.frame_3)
self.push_button_cinetica_cunia.setObjectName(u"push_button_cinetica_cunia")
self.verticalLayout_2.addWidget(self.push_button_cinetica_cunia)

self.gridLayout.addWidget(self.frame_3, 1, 2, 1, 1, Qt.AlignTop)

self.frame_4 = QFrame(self.centralwidget)
self.frame_4.setObjectName(u"frame_4")
self.frame_4 setFrameShape(QFrame.StyledPanel)
self.frame_4 setFrameShadow(QFrame.Raised)
self.verticalLayout_3 = QVBoxLayout(self.frame_4)
self.verticalLayout_3.setObjectName(u"verticalLayout_3")
self.label_4 = QLabel(self.frame_4)
self.label_4.setObjectName(u"label_4")

self.verticalLayout_3.addWidget(self.label_4)

self.push_button_zonif_planar = QPushButton(self.frame_4)
self.push_button_zonif_planar.setObjectName(u"push_button_zonif_planar")

self.verticalLayout_3.addWidget(self.push_button_zonif_planar)

self.push_button_zonif_volteo = QPushButton(self.frame_4)
self.push_button_zonif_volteo.setObjectName(u"push_button_zonif_volteo")

self.verticalLayout_3.addWidget(self.push_button_zonif_volteo)

self.push_button_zonif_cunnia = QPushButton(self.frame_4)
self.push_button_zonif_cunnia.setObjectName(u"push_button_zonif_cunnia")

self.verticalLayout_3.addWidget(self.push_button_zonif_cunnia)

self.gridLayout.addWidget(self.frame_4, 1, 0, 1, 1)

self.frame = QFrame(self.centralwidget)
self.frame.setObjectName(u"frame")
sizePolicy1 = QSizePolicy(QSizePolicy.Preferred, QSizePolicy.Preferred)

```

```

sizePolicy1.setHorizontalStretch(0)
sizePolicy1.setVerticalStretch(0)

sizePolicy1.setHeightForWidth(self.frame.sizePolicy().hasHeightForWidth())
self.frame.setSizePolicy(sizePolicy1)
self.frame.setFrameShape(QFrame.StyledPanel)
self.frame.setFrameShadow(QFrame.Raised)
self.gridLayout_2 = QGridLayout(self.frame)
self.gridLayout_2.setObjectName(u"gridLayout_2")
self.label_3 = QLabel(self.frame)
self.label_3.setObjectName(u"label_3")

self.gridLayout_2.addWidget(self.label_3, 0, 0, 1, 2)

self.push_button_data = QPushButton(self.frame)
self.push_button_data.setObjectName(u"push_button_data")

self.gridLayout_2.addWidget(self.push_button_data, 1, 0, 1, 1)

self.horizontalSpacer = QSpacerItem(40, 20, QSizePolicy.Expanding,
QSizePolicy.Minimum)

self.gridLayout_2.addItem(self.horizontalSpacer, 1, 1, 1, 1)

self.gridLayout.addWidget(self.frame, 0, 0, 1, 3)

self.verticalSpacer = QSpacerItem(20, 40, QSizePolicy.Minimum,
QSizePolicy.Expanding)

self.gridLayout.addItem(self.verticalSpacer, 2, 0, 1, 3)

MainWindow.setCentralWidget(self.centralwidget)

self.retranslateUi(MainWindow)

QMetaObject.connectSlotsByName(MainWindow)
# setupUi

def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow",
u"An\u00e1lisis probabil\u00edstico de estabilidad de taludes en roca", None))
    self.label.setText(QCoreApplication.translate("MainWindow",
u"Cinem\u00e1tica", None))

self.push_button_cinematica_desliz.setText(QCoreApplication.translate("MainWindo
w", u"Planar", None))

self.push_button_cinematica_volteo.setText(QCoreApplication.translate("MainWindo
w", u"Volteo", None))

self.push_button_cinematica_cunia.setText(QCoreApplication.translate("MainWindow
", u"Cu\u00f1a", None))
    self.label_2.setText(QCoreApplication.translate("MainWindow",

```

```

u"Cin\u00e9tica", None))

self.push_button_cinetica_desliz.setText(QCoreApplication.translate("MainWindow",
, u"Planar", None))

self.push_button_cinetica_volteo.setText(QCoreApplication.translate("MainWindow",
, u"Volteo", None))

self.push_button_cinetica_cunia.setText(QCoreApplication.translate("MainWindow",
u"Cu\u00f1a", None))
    self.label_4.setText(QCoreApplication.translate("MainWindow",
u"Zonificaci\u00f3n Polar", None))

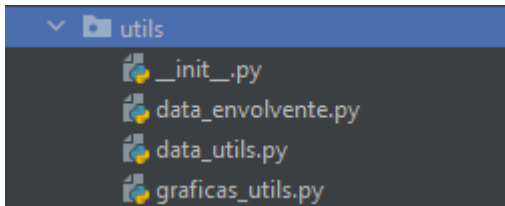
self.push_button_zonif_planar.setText(QCoreApplication.translate("MainWindow",
u"Planar", None))

self.push_button_zonif_volteo.setText(QCoreApplication.translate("MainWindow",
u"Volteo", None))

self.push_button_zonif_cunnia.setText(QCoreApplication.translate("MainWindow",
u"Cu\u00f1a", None))
    self.label_3.setText(QCoreApplication.translate("MainWindow", u"Datos",
None))
    self.push_button_data.setText(QCoreApplication.translate("MainWindow",
u"Datos de rumbo, buzamiento y fricci\u00f3n", None))
        # retranslateUi

```

3. M\u00d3DULO UTILS



utils/__init__.py

```
from .data_utils import leer_desde_csv, create_red_polar
```

utils/data_envolvente.py

```
import numpy as np
import pandas as pd
```

```
def calcular_envolvente(red, beta, alfa_t=0, epsilon=1e-100):
    """
    Calcula los puntos (alfa, beta) de la envolvente.
    :param beta: \u00c1ngulo beta de la envolvente
    :param alfa_t: \u00c1ngulo alfa al que se rota la envolvente
    :param epsilon: Valor infinitesimal para evitar divisiones por 0
    :return: dataframe. alfa, beta
    """
    df = pd.DataFrame()
    if beta >= 90:
```

```

    beta = 89.9
    df['b_rad'] = np.deg2rad(red['alfa'])
    df['d_rad'] = np.deg2rad(beta)
    df['sin_d'] = np.sin(df['d_rad'])
    df['p'] = np.arctan(np.sin(df['b_rad']) * np.tan(df['d_rad']))
    df['sin_r'] = np.sin(df['p']) / (df['sin_d'] + epsilon)
    df['cos_r'] = np.cos(np.arcsin(df['sin_r']))
    df['q'] = np.sqrt(1 - (df['sin_d'] * df['sin_r']) ** 2)
    df['k'] = df['sin_d'] * df['sin_r'] / (df['q'] + epsilon) * np.sqrt(2 / (1 +
df['q']))
    df['xE'] = df['k'] * np.cos(df['d_rad']) * df['sin_r']
    df['yE'] = df['k'] * df['cos_r']
    df['alfa_e'] = np.rad2deg(np.arctan(df['yE'] / (df['xE'] + epsilon))) +
alfa_t
    df['alfa_e'] = df['alfa_e'].fillna(0)
    df['alfa_e'] = df['alfa_e'].apply(lambda x: x if x < 360 else x - 360)
    df['alfa_e'] = df['alfa_e'].apply(lambda x: x if x >= 0 else 360 + x)
    df['beta_e'] = np.rad2deg(np.sqrt((df[['xE', 'yE']] ** 2).sum(axis=1)))
    df = df[['alfa_e', 'beta_e']].rename(columns={'alfa_e': 'alfa', 'beta_e':
'beta'})
    df = df.sort_values(by=['alfa', 'beta'])
    df = df.round()
    df = df.drop_duplicates()
    # print(df[['yE', 'xE', 'alfa_e', 'beta_e']])
    return df

```

```

def envolvente3d(red_df: pd.DataFrame, beta, alfa_t=0, epsilon=1e-100):
    df = red_df[['alfa', 'beta']]
    env_df = calcular_envolvente(red_df, beta, alfa_t, epsilon)
    for idx, row in env_df.iterrows():
        df.loc[df['alfa'] == row['alfa'], 'env'] = (df.loc[df['alfa'] ==
row['alfa'], 'beta'] <= row['beta']) * 1.0
    df = df.fillna(0)
    return df

```

```

def beta_min(red_df: pd.DataFrame, paleom_df, alfa_t):
    def iterar(beta_0, beta_1, partes):
        print(f'Buscando beta mínimo entre {beta_0} y {beta_1}')
        paso = (beta_1 - beta_0) // partes
        if paso == 0:
            paso = 1
        suma_env = 0.
        # rango = list()
        # print(rango)
        for b in range(beta_0, beta_1 + 1, paso):
            env_df = envolvente3d(red_df, b, alfa_t)
            env_df['env'] = paleom_df['fp'] * env_df['env']
            suma_1 = suma_env
            suma_env = env_df['env'].sum()
            # print(b, suma_env, suma_1)
            if suma_env > suma_1:
                beta_1 = b

```

```

        else:
            if suma_env > 0:
                break
    # print(beta_1 - paso, beta_1)
    if paso > 1:
        beta_1 = iterar(beta_1 - paso, beta_1, partes)
    return beta_1

return iterar(0, 90, 5)

```

utils/data_utils.py

```

import numpy as np
import pandas as pd

```

```

def leer_desde_csv(ruta: str, separador: str = ','):
    return pd.read_csv(ruta, sep=separador)

```

```

def rumbo_a_azimut(angulo_rumbo: np.ndarray, direccion_rumbo: str,
angulo_buzamiento: np.ndarray,
                    direccion_buzamiento: str) -> np.ndarray:
    """
    Convierte de rumbo a azimut
    :param angulo_rumbo: Matriz numpy de Los ángulos de rumbo
    :param direccion_rumbo: Dirección del rumbo {'SE', 'NE', 'SW', 'NW'}
    :param angulo_buzamiento: Matriz numpy de Los ángulos del buzamiento
    :param direccion_buzamiento: Dirección del buzamiento {'SE', 'NE', 'SW',
'NW'}
    :return: [azimut, buzamiento]. Matriz de numpy
    """
    const_error = '[error rumbo_a_azimut]'
    error_response = np.array([])
    direcciones = {'SE', 'NE', 'SW', 'NW'}
    # Comprobar que las direcciones son correctas
    if direccion_rumbo not in direcciones or direccion_buzamiento not in
direcciones:
        print(const_error, f'Las direcciones deben ser una de: {direcciones}')
        return error_response

    # Comprobar que los ángulos estén dentro de los rangos correctos
    if angulo_rumbo > 90.0:
        print(const_error, f'Los ángulos de rumbo y buzamiento deben ser menores
o iguales a 90°')
        return error_response

    # Calcular azimut de acuerdo a las direcciones de rumbo y buzamiento
    if direccion_buzamiento == 'SE':
        if direccion_rumbo not in {'NE', 'SW'}:
            print(const_error, f'La dirección de buzamiento no corresponde con
la dirección de rumbo seleccionada')
            return error_response
        azimut = angulo_rumbo + 90.0
        return np.hstack((azimut, angulo_buzamiento))

```

```

elif direccion_buzamiento == 'NW':
    if direccion_rumbo not in {'NE', 'SW'}:
        print(const_error, f'La dirección de buzamiento no corresponde con
la dirección de rumbo seleccionada')
        return error_response
    azimut = angulo_rumbo + 270.0
    return np.hstack((azimut, angulo_buzamiento))
elif direccion_buzamiento == 'SW':
    if direccion_rumbo not in {'NW', 'SE'}:
        print(const_error, f'La dirección de buzamiento no corresponde con
la dirección de rumbo seleccionada')
        return error_response
    azimut = -angulo_rumbo + 270.0
    return np.hstack((azimut, angulo_buzamiento))
elif direccion_buzamiento == 'NE':
    if direccion_rumbo not in {'NW', 'SE'}:
        print(const_error, f'La dirección de buzamiento no corresponde con
la dirección de rumbo seleccionada')
        return error_response
    azimut = -angulo_rumbo + 90.0
    return np.hstack((azimut, angulo_buzamiento))

```

```

def calcular_azimut(df):
    """
    Calcula el azimut a partir del ángulo de rumbo y La dirección del buzamiento
    :param df: dataframe con, buzamiento_dir y rumbo_ang
    :return: float, azimut en grados
    """
    if df['buzamiento_dir'] == 'SE':
        return df['rumbo_ang'] + 90
    elif df['buzamiento_dir'] == 'NW':
        return df['rumbo_ang'] + 270
    elif df['buzamiento_dir'] == 'SW':
        return -df['rumbo_ang'] + 270
    elif df['buzamiento_dir'] == 'NE':
        return -df['rumbo_ang'] + 90
    else:
        return 0

```

```

def rumbo_a_azimut_df(df_datos: pd.DataFrame) -> pd.DataFrame:
    """
    Calcula el azimut y buzamiento a partir del rumbo y el buzamiento
    :param df_datos: dataframe con rumbo_ang, rumbo_dir, buzamiento_ang y
buzamiento_dir
    :return: dataframe. alfa (Azimut) y beta (Buzamiento)
    """
    df = pd.DataFrame()
    df['alfa'] = df_datos.apply(calcular_azimut, axis=1)
    df['beta'] = df_datos['buzamiento_ang']
    return df

```

```

def agrupar_datos(dataframe: pd.DataFrame) -> pd.DataFrame:
    # Organizar datos por azimuth
    sorted_df = dataframe.sort_values('alfa', ascending=True)
    # Hallar la diferencia entre los alfa_i y alfa_cero
    alfa_cero = sorted_df.iloc[0]['alfa']
    sorted_df['d1'] = sorted_df['alfa'] - alfa_cero
    familia = 1
    # Hallar la diferencia entre el dato actual y el anterior
    d2 = sorted_df['d1']
    sorted_df['d2'] = sorted_df['d1'] - d2.shift(1)
    sorted_df['Familia'] = 0
    # Si la diferencia es mayor a 20 entonces asignar a nueva familia
    for idx, row in sorted_df.iterrows():
        if row['d2'] > 20:
            familia = familia + 1
            sorted_df.at[idx, 'Familia'] = familia
    # print('\nAgrupados\n', sorted_df)
    return sorted_df

def cosenos_directores(datos_df: pd.DataFrame) -> pd.DataFrame:
    """
    Calcula los cosenos directores del azimuth y buzamiento
    :param datos_df: dataframe. Azimut y Buzamiento en grados, Familia
    :return: dataframe. L, m, n, Familia
    """
    cos_directores = pd.DataFrame()
    # Cambiar de grados a radianes
    datos_rads_df = np.deg2rad(datos_df[['alfa', 'beta']])
    # Calcular cosenos directores
    cos_directores['l'] = np.cos(datos_rads_df['alfa']) *
np.cos(datos_rads_df['beta'])
    cos_directores['m'] = np.cos(datos_rads_df['beta']) *
np.sin(datos_rads_df['alfa'])
    cos_directores['n'] = np.sin(datos_rads_df['beta'])
    cos_directores['Familia'] = datos_df['Familia']
    # print('\nCosenos directores\n', cos_directores)
    return cos_directores

def sumatoria_cos_directores(cos_directores_df: pd.DataFrame) -> pd.DataFrame:
    # Agrupar datos por familia
    group_directores = cos_directores_df.groupby(by='Familia')
    # Sumar los cosenos de cada familia
    suma_directores = group_directores.sum()
    # Agregar el número de datos de cada familia
    suma_directores['N'] = group_directores.size()
    # print('\nSumatoria cos directores\n', suma_directores)

    return suma_directores

def calculo_r(suma_directores: pd.DataFrame) -> pd.DataFrame:
    r_df = pd.DataFrame()

```

```

# Elevar la suma de los cosenos directores al cuadrado
r_df = suma_directores[['l', 'm', 'n']] ** 2
# Calcular la raíz de la suma de los cuadrados de cada familia
r_df['R'] = np.sqrt(r_df[['l', 'm', 'n']].sum(axis=1))
r_df['N'] = suma_directores['N']
# print('\nRs\n', r_df)
return r_df

def calculo_representativos(r_df: pd.DataFrame, suma_directores_df:
pd.DataFrame) -> pd.DataFrame:
    # Cargar l, m, n en representativos
    representativos_df = suma_directores_df
    # Cargar R de r_df en representativos
    representativos_df['R'] = r_df['R']
    # Dividir l, m, n de cada familia por el R de cada familia
    representativos_df = representativos_df[['l', 'm',
'n']].div(representativos_df['R'], axis='index')
    # Calcular representativos
    representativos_df['beta'] = np.arcsin(representativos_df['n'])

    def calcular_alfa_f(row):
        af = np.arccos(row['l'] / np.cos(row['beta']))
        if row['m'] >= 0:
            return af
        else:
            return -af

    representativos_df['alfa'] = representativos_df.apply(calcular_alfa_f,
axis=1)

    # Cambiar alfa y beta a grados
    representativos_df[['alfa', 'beta']] =
np.rad2deg(representativos_df[['alfa', 'beta']])
    representativos_df['alfa_p'] =
representativos_df['alfa'].apply(alfa_a_alfa_p)
    representativos_df['beta_p'] =
representativos_df['beta'].apply(beta_a_beta_p)
    representativos_df['N'] = r_df['N']
    representativos_df['R'] = r_df['R']
    # print('\nRepresentativos\n', representativos_df[['alfa_p', 'beta_p']])

    return representativos_df

def calcular_k(rep_df: pd.DataFrame) -> pd.DataFrame:
    kf_df = pd.DataFrame()
    kf_df['K'] = (rep_df['N'] - 1) / (rep_df['N'] - rep_df['R'])
    # print('\nK_f\n', kf_df)
    return kf_df

def alfa_a_alfa_p(alfa):
    if (alfa + 180) > 360:

```



```

        alfa_p = alfa - 180
    else:
        alfa_p = alfa + 180
    return alfa_p

def beta_a_beta_p(beta):
    return 90 - beta

def create_red_polar(alfa_init=1, alfa_fin=361, beta_init=0, beta_fin=91):
    red = pd.DataFrame()
    for b in np.arange(beta_init, beta_fin):
        alfa = np.arange(alfa_init, alfa_fin)
        df = pd.DataFrame()
        df['alfa'] = alfa
        df['beta'] = np.ones_like(alfa) * b
        red = pd.concat([red, df], ignore_index=True)

    return red

def area_casquete(theta):
    if theta > 0:
        return np.pi * (2 * theta - 1) / 2_916_000 # 2_916_000 = 360 * (90**2)
    else:
        return 0

def calcular_envolvente(data_df: pd.DataFrame, alfa_inicial, alfa_name: str =
'alfa_p', beta_name: str = 'beta_p'):
    df = pd.DataFrame()
    df[['alfa', 'beta']] = data_df[['alfa_name', 'beta_name']]
    df[['alfa_rad', 'beta_rad']] = np.deg2rad(df)
    df['tan_beta'] = np.tan(df['beta_rad'])
    df['sin_alfa'] = np.sin(df['alfa_rad'])
    df['p'] = np.arctan(df['tan_beta'] * df['sin_alfa'])
    df['p_grad'] = np.rad2deg(df['p'])
    df['sin_p'] = np.sin(df['p'])
    df['sin_beta'] = np.sin(df['beta_rad'])
    df['cos_beta'] = np.cos(df['beta_rad'])
    df['r'] = np.arcsin(df['sin_p'] / df['sin_beta'])
    df['sin_r'] = np.sin(df['r'])
    df['k'] = np.sqrt(2 / (1 + df['sin_r'] * df['sin_beta']))
    df['x'] = df['k'] * df['sin_r'] * df['cos_beta']
    df['y'] = df.apply(lambda x: x['k'] * np.cos(x['r']) if x['alfa'] <= 90 else
-x['k'] * np.cos(x['r']), axis=1)
    df['pE_grad'] = 90 - df['p_grad']
    df['bE_grad'] = 180 + df['alfa']
    df['tan2_p'] = np.power(np.tan(df['p']), 2)
    df['dE'] = np.arctan(df['tan_beta'] / df['tan2_p'])
    df['rE'] = np.arcsin(np.cos(df['p']) / np.sin(df['dE']))
    df['kE'] = np.sqrt(2 / (1 + np.sin(df['rE']) * np.sin(df['dE'])))
    df['xE'] = df.apply(lambda x: -x['kE'] * np.sin(x['rE']) * np.cos(x['dE'])

```

```

if x['bE_grad'] >= 180
    else x['kE'] * np.sin(x['rE']) * np.cos(x['dE']), axis=1)
    df['yE'] = df.apply(lambda x: x['kE'] * np.cos(x['rE']) if x['alfa'] <= 90
else -x['kE'] * np.cos(x['rE']), axis=1)
    df['alfa_2_grad'] = (270 - np.rad2deg(np.arctan(df['yE'] /
df['xE']))).fillna(0)

def calcular_alfa(alfa, alfa_init):
    g = alfa + alfa_init
    if alfa_init < 270:
        if (g - 270) < 0:
            return g - 90
        else:
            return g - 270
    else:
        if (g - 270) > 360:
            return g - 630
        else:
            return g - 270

df['alfa_2_grad'] = df['alfa_2_grad'].apply(lambda x: calcular_alfa(x,
alfa_inicial))
df['beta_2_grad'] = np.rad2deg(np.sqrt((df[['xE', 'yE']] ** 2).sum(axis=1)))

print('\nEnvolvente\n', df.keys(), '\n', df[['alfa', 'beta', 'alfa_2_grad',
'beta_2_grad']])
return df[['alfa_2_grad', 'beta_2_grad']].rename(columns={'alfa_2_grad':
'alfa', 'beta_2_grad': 'beta'})

def calcular_envolvente2(data_df, azimut=0, epsilon=1e-100):
df = pd.DataFrame()
df['b_rad'] = np.deg2rad(data_df['alfa'])
df['d_rad'] = np.deg2rad(data_df['beta'])
df['sin_d'] = np.sin(df['d_rad'])
df['p'] = np.arctan(np.sin(df['b_rad']) * np.tan(df['d_rad']))
df['sin_r'] = np.sin(df['p']) / (df['sin_d'] + epsilon)
df['cos_r'] = np.cos(np.arcsin(df['sin_r']))
df['q'] = np.sqrt(1 - (df['sin_d'] * df['sin_r']) ** 2)
df['k'] = df['sin_d'] * df['sin_r'] / (df['q'] + epsilon) * np.sqrt(2 / (1 +
df['q']))
df['xE'] = df['k'] * np.cos(df['d_rad']) * df['sin_r']
df['yE'] = df['k'] * df['cos_r']
df['alfa_e'] = np.rad2deg(np.arctan(df['yE'] / (df['xE'] + epsilon))) +
azimut
df['beta_e'] = np.rad2deg(np.sqrt((df[['xE', 'yE']] ** 2).sum(axis=1)))
# print(df[['b_rad', 'd_rad', 'alfa_e', 'beta_e']])
return df[['alfa_e', 'beta_e']].rename(columns={'alfa_e': 'alfa', 'beta_e':
'beta'})

def areas(red_df: pd.DataFrame) -> pd.DataFrame:
df = red_df[['alfa', 'beta']]
df['a'] = np.pi * (2 * df['beta'] - 1) / 2_916_000

```

```

df.iloc[df['a'] < 0] = 0
return df

def prob_acc_sismica(t_retorno: float, periodo: float) -> float:
    return 1 - (1 - 1 / t_retorno) ** periodo

utils/graficas_utils.py
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

from utils.data_utils import alfa_a_alfa_p, beta_a_beta_p
from app.controlador_datos import DataOut

def grafica_polar(data: pd.DataFrame, **kwargs):
    polos_df = pd.DataFrame()
    polos_df['alfa'] = data['alfa']
    polos_df['alfa'] = data['alfa'].apply(alfa_a_alfa_p)
    polos_df['alfa'] = np.deg2rad(polos_df['alfa'])
    polos_df['beta'] = data['beta']
    polos_df['beta'] = data['beta'].apply(beta_a_beta_p)
    fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
    ax.plot(polos_df['alfa'], polos_df['beta'], ls='', marker='.')
    ax.set_rmax(90)
    ax.set_theta_zero_location('N')
    ax.set_theta_direction(-1)

    # Configurar títulos y etiquetas
    if 'titulo' in kwargs:
        ax.set_title(kwargs['titulo'])

    if 'etiquetas' in kwargs:
        # Colocar etiquetas de datos
        # ax.set
        pass

    return fig

def grafica_zonif_planar(rep_df: pd.DataFrame, angulo_friccion: float, fig:
plt.Figure):
    ax = fig.add_subplot(projection='polar')
    alfa_rad = np.deg2rad(rep_df['alfa_p'])
    ax.plot(alfa_rad, rep_df['beta_p'], ls='', marker='.')
    alfa_fric = np.deg2rad(np.arange(0, 360))
    beta_fric = np.ones_like(alfa_fric) * angulo_friccion
    ax.plot(alfa_fric, beta_fric, ls='-')
    ax.set_rmax(90)
    ax.set_theta_zero_location('N')
    ax.set_theta_direction(-1)
    ax.set_title('Zona de falla planar')

```

```

for idx, row in rep_df.iterrows():
    x = np.arange(row['alfa_p'] - 20, row['alfa_p'] + 20)
    x = np.deg2rad(x)
    ax.annotate(f'Familia {idx}\n({round(row["alfa_p"])},
{round(row["beta_p"])}),',
                (np.deg2rad(row['alfa_p']), row['beta_p']))
    if row['beta_p'] > angulo_friccion:
        y1 = np.ones_like(x) * row['beta_p']
    else:
        y1 = np.ones_like(x) * angulo_friccion
    y2 = np.ones_like(x) * 90
    ax.fill_between(x, y1, y2, alpha=0.5)

fig.canvas.draw_idle()

def grafica_zonif_volteo(rep_df: pd.DataFrame, angulo_friccion: float, fig:
plt.Figure):
    ax = fig.add_subplot(projection='polar')
    alfa_rad = np.deg2rad(rep_df['alfa_p'])
    ax.plot(alfa_rad, rep_df['beta_p'], ls='', marker='.')
    alfa_fric = np.deg2rad(np.arange(0, 360))
    beta_fric = np.ones_like(alfa_fric) * angulo_friccion
    ax.plot(alfa_fric, beta_fric, ls='--')
    ax.set_rmax(90)
    ax.set_theta_zero_location('N')
    ax.set_theta_direction(-1)
    ax.set_title('Zona de falla por volteo')

    for idx, row in rep_df.iterrows():
        alfa_v = row['alfa_p'] + 180
        x = np.arange(alfa_v - 20, alfa_v + 20)
        x = np.deg2rad(x)
        ax.annotate(f'Familia {idx}\n({round(row["alfa_p"])},
{round(row["beta_p"])}),',
                    (np.deg2rad(row['alfa_p']), row['beta_p']))
        if row['beta_p'] < angulo_friccion:
            y1 = np.ones_like(x) * row['beta_p']
            y2 = np.ones_like(x) * 90
            ax.fill_between(x, y1, y2, alpha=0.5)

    fig.canvas.draw_idle()

def grafica_zonif_cunnia(datos: DataOut, angulo_friccion: float, fig:
plt.Figure):
    ax = fig.add_subplot(projection='polar')
    alfa_rad = np.deg2rad(datos.zonif_cunnia.familias['alfa_p'])
    ax.plot(alfa_rad, datos.zonif_cunnia.familias['beta_p'], ls='', marker='.')
    alfa_fric = np.deg2rad(np.arange(0, 360))
    beta_fric = np.ones_like(alfa_fric) * angulo_friccion
    ax.plot(alfa_fric, beta_fric, ls='--')
    ax.set_rmax(90)
    ax.set_theta_zero_location('N')

```

```

ax.set_theta_direction(-1)
ax.set_title('Zona de falla cuña')

for idx, row in datos.zonif_cunnia.familias.iterrows():
    ax.annotate(f'Familia {idx}\n({round(row["alfa_p"])},
{round(row["beta_p"])}))',
                (np.deg2rad(row['alfa_p']), row['beta_p']))

for idx, row in datos.zonif_cunnia.intersec.iterrows():
    x = np.arange(row['alfa_min'], row['alfa_max'])
    x = np.deg2rad(x)
    ax.annotate(f'Intersección {idx}\n({round(row["alfa_p"])},
{round(row["beta_p"])}))',
                (np.deg2rad(row['alfa_p']), row['beta_p']))
    if row['beta_p'] > angulo_friccion:
        y1 = np.ones_like(x) * row['beta_p']
    else:
        y1 = np.ones_like(x) * angulo_friccion
        y2 = np.ones_like(x) * 90
    ax.fill_between(x, y1, y2, alpha=0.5)

fig.canvas.draw_idle()

def grafica_envolvente(data: pd.DataFrame, **kwargs):
    polos_df = pd.DataFrame()
    polos_df['alfa'] = np.deg2rad(data['alfa'])
    polos_df['beta'] = data['beta']
    fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
    ax.plot(polos_df['alfa'], polos_df['beta'], ls='-', marker='')
    ax.set_rmax(90)
    ax.set_theta_zero_location('N')
    ax.set_theta_direction(-1)

    # Configurar títulos y etiquetas
    if 'titulo' in kwargs:
        ax.set_title(kwargs['titulo'])

    if 'etiquetas' in kwargs:
        # Colocar etiquetas de datos
        # ax.set
        pass

    return fig

def grafica_contorno(df: pd.DataFrame, z_label: str, max_level, transparencia:
float, fig=None, **kwargs):
    df_rad = pd.DataFrame({'alfa': np.deg2rad(df['alfa']), 'beta': df['beta'],
'val': df[z_label]})
    data_df = df_rad.pivot(index='beta', columns='alfa', values='val')
    alfa_rad = np.deg2rad(np.arange(df['alfa'].min(), df['alfa'].max() + 1))
    alfa, beta = np.meshgrid(alfa_rad, np.arange(0, df['beta'].max() + 1))
    if fig is None:

```

```

    fig, ax = plt.subplots(subplot_kw={'projection': 'polar'})
else:
    ax = fig.axes[0]
levels = np.linspace(0, max_level, 100)
colores_fcn = plt.get_cmap('rainbow', 100)
colores_lst = colores_fcn(np.linspace(0, 1, 100), transparencia)
colores_lst[0, :] = [1., 1., 1., 0.]
cmp = plt.cm.colors.ListedColormap(colores_lst)
# print(colores(np.linspace(0, 1, 100)))
cs = ax.contourf(alfa, beta, data_df, levels, cmap=cmp)
cbar = fig.colorbar(cs)
ax.set_rmax(90)
ax.set_theta_zero_location('N')
ax.set_theta_direction(-1)

# Configurar títulos y etiquetas
if 'titulo' in kwargs:
    ax.set_title(kwargs['titulo'])

if 'x_label' in kwargs:
    ax.set_xlabel(kwargs['x_label'])

if 'y_label' in kwargs:
    ax.set_ylabel(kwargs['y_label'])

if 'z_label' in kwargs:
    ax.set_zlabel(kwargs['z_label'])

if 'anotation' in kwargs:
    fig.text(0.01, 0.01, kwargs['anotation'])

def grafica_3d(df: pd.DataFrame, z_label: str, fig=None, **kwargs):
    df_rad = pd.DataFrame({'alfa': np.deg2rad(df['alfa']), 'beta': df['beta'],
'val': df[z_label]})
    data_df = df_rad.pivot(index='beta', columns='alfa', values='val')
    alfa_rad = np.deg2rad(np.arange(df['alfa'].min(), df['alfa'].max() + 1))
    alfa, beta = np.meshgrid(alfa_rad, np.arange(0, df['beta'].max() + 1))
    x, y = beta * np.cos(alfa), beta * np.sin(alfa)
    colores_fcn = plt.get_cmap('rainbow', 100)
    colores_lst = colores_fcn(np.linspace(0, 1, 100))
    colores_lst[0, :] = [1., 1., 1., 0.]
    cmp = plt.cm.colors.ListedColormap(colores_lst)
    if fig is None:
        fig = plt.figure()
    ax = fig.add_subplot(projection='3d')
    ax.plot_surface(x, y, data_df, cmap=cmp)

    return fig

def grafica_3d_xyz(x_vector, y_vector, df: pd.DataFrame, x_column: str,
y_column: str, z_column: str, **kwargs):
    # Mapa de colores

```

```

colores_fcn = plt.get_cmap('rainbow', 100)
colores_lst = colores_fcn(np.linspace(0, 1, 100))
colores_lst[0, :] = [1., 1., 1., 0.]
cmp = plt.cm.colors.ListedColormap(colores_lst)

# Alistamiento de datos
data_df = df.pivot(index=y_column, columns=x_column, values=z_column)
x, y = np.meshgrid(x_vector, y_vector)
maxim = data_df.values.max()

# Gráfica
fig = kwargs['figure']
ax = fig.add_subplot(projection='3d')
cs = ax.contourf(x, y, data_df.values, offset=1.1 * maxim, levels=20,
cmap=cmp)
ax.plot_surface(x, y, data_df, cmap=cmp)
cbar = fig.colorbar(cs)

if 'titulo' in kwargs:
    ax.set_title(kwargs['titulo'])

if 'x_label' in kwargs:
    ax.set_xlabel(kwargs['x_label'])

if 'y_label' in kwargs:
    ax.set_ylabel(kwargs['y_label'])

if 'z_label' in kwargs:
    ax.set_zlabel(kwargs['z_label'])

if 'anotation' in kwargs:
    fig.text(0.01, 0.01, kwargs['anotation'])

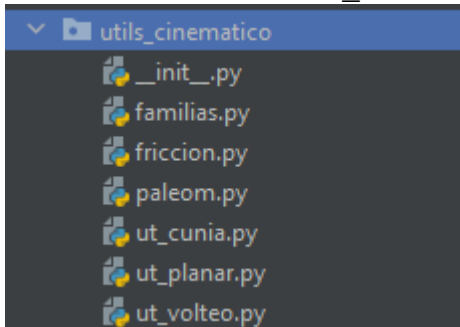
return ax

def graficar():
    plt.show()

def cerrar(fig: plt.Figure):
    plt.close(fig)

```

4. MÓDULO UTILS_CINEMATICO



```
utils_cinematico/__init__.py
```

```
from .ut_planar import cinematica_planar
from .ut_volteo import cinematica_volteo
from .ut_cunia import cinematica_cunia
```

```
utils_cinematico/familias.py
```

```
from itertools import combinations
```

```
import numpy as np
import pandas as pd
```

```
def interseccion_familias(datos_df: pd.DataFrame) -> pd.DataFrame:
    azimuts_df = rumbo2azimut_df(datos_df)
    familias_df = agrupar_datos(azimuts_df)[['alfa', 'beta', 'familia']]
    idx_familias = range(1, familias_df['familia'].max() + 1)
    perm = combinations(idx_familias, 2)
    inter_df = pd.DataFrame()

    for x, y in perm:
        familia_1 = familias_df[familias_df['familia'] == x]
        familia_2 = familias_df[familias_df['familia'] == y]

        for idx, row in familia_1.iterrows():
            df = familia_2[['alfa', 'beta']].rename(columns={'alfa':
'alfa_b', 'beta': 'beta_b'})
            df[['alfa_a', f'beta_a']] = row[['alfa', 'beta']]
            df['inter'] = f'{x}-{y}'
            inter_df = inter_df.append(df, ignore_index=True)

    inter_rad_df = np.deg2rad(inter_df[['alfa_a', 'beta_a', 'alfa_b',
'beta_b']])
    inter_rad_df['inter'] = inter_df['inter']
    inter_rad_df['tan_beta_a'] = np.tan(inter_rad_df['beta_a'])
    inter_rad_df['tan_beta_b'] = np.tan(inter_rad_df['beta_b'])
    inter_rad_df['alfa_ab'] = np.arctan((inter_rad_df['tan_beta_a'] *
np.cos(inter_rad_df['alfa_a']) -
                                inter_rad_df['tan_beta_b'] *
np.cos(inter_rad_df['alfa_b'])) /
                                (inter_rad_df['tan_beta_b'] *
np.sin(inter_rad_df['alfa_b']) -
                                inter_rad_df['tan_beta_a'] *
np.sin(inter_rad_df['alfa_a'])))

    inter_rad_df.loc[inter_rad_df['alfa_ab'] < 0, 'alfa_ab'] =
inter_rad_df.loc[
inter_rad_df['alfa_ab'] < 0, 'alfa_ab'] + np.pi
    inter_rad_df['beta_ab'] = inter_rad_df['tan_beta_b'] *
np.cos(inter_rad_df['alfa_b']) - inter_rad_df['alfa_ab'])
    inter_rad_df.loc[inter_rad_df['beta_ab'] < 0, 'beta_ab'] = -1 *
inter_rad_df.loc[
inter_rad_df['beta_ab'] < 0, 'beta_ab']

    # Aproximar ángulos a cero decimales
```



```

inter_df['alfa_ab'] = np.round(np.rad2deg(inter_rad_df['alfa_ab']))
inter_df['beta_ab'] = np.round(np.rad2deg(inter_rad_df['beta_ab']))

# Calculo de cosenos directores
inter_rad_df['l'] = np.cos(inter_rad_df['alfa_ab']) *
np.cos(inter_rad_df['beta_ab'])
inter_rad_df['m'] = np.cos(inter_rad_df['beta_ab']) *
np.sin(inter_rad_df['alfa_ab'])
inter_rad_df['n'] = np.sin(inter_rad_df['beta_ab'])

# Calculo de sumatoria de cosenos directores
suma_directores_df = sumatoria_cos_directores(
    inter_rad_df[['l', 'm', 'n', 'inter']].rename(columns={'inter':
'familia'}))

# Cálculo de R, representativos y k
r_df = calculo_r(suma_directores_df)
repr_df = calculo_representativos(r_df, suma_directores_df)
k_df = calcular_k(repr_df)

rta_df = pd.DataFrame()
rta_df[['alfa_p', 'beta_p']] = repr_df[['alfa_p', 'beta_p']]
rta_df['k'] = k_df['K']
# print(rta_df)
return rta_df

def data_familias_volteo(datos_df: pd.DataFrame) -> pd.DataFrame:
df = pd.DataFrame()
azimuts_df = rumbo2azimut_df(datos_df)
familias_df = agrupar_datos(azimuts_df)
familias_df['alfa'] = familias_df['alfa'].apply(alfa_a_alfa_p)
familias_df['beta'] = familias_df['beta'].apply(beta_a_beta_p)
cos_directores = cosenos_directores(familias_df)
sum_cos_dir = sumatoria_cos_directores(cos_directores)
r_df = calculo_r(sum_cos_dir)
representativos_df = calculo_representativos(r_df, sum_cos_dir)
k_df = calcular_k(representativos_df)
df[['alfa_p', 'beta_p']] = representativos_df[['alfa_p', 'beta_p']]
df['k'] = k_df
return df

def data_familias(datos_df: pd.DataFrame) -> pd.DataFrame:
df = pd.DataFrame()
azimuts_df = rumbo2azimut_df(datos_df)
familias_df = agrupar_datos(azimuts_df)
cos_directores = cosenos_directores(familias_df)
sum_cos_dir = sumatoria_cos_directores(cos_directores)
r_df = calculo_r(sum_cos_dir)
representativos_df = calculo_representativos(r_df, sum_cos_dir)
k_df = calcular_k(representativos_df)
df[['alfa_p', 'beta_p']] = representativos_df[['alfa_p', 'beta_p']]
df['k'] = k_df
return df

```

```

def rumbo2azimut(df):
    """
    Calcula el azimut a partir del ángulo de rumbo y la dirección del
    buzamiento
    :param df: dataframe con, buzamiento_dir y rumbo_ang
    :return: float, azimut en grados
    """
    if df['buzamiento_dir'] == 'SE':
        return df['rumbo_ang'] + 90
    elif df['buzamiento_dir'] == 'NW':
        return df['rumbo_ang'] + 270
    elif df['buzamiento_dir'] == 'SW':
        return -df['rumbo_ang'] + 270
    elif df['buzamiento_dir'] == 'NE':
        return -df['rumbo_ang'] + 90
    else:
        return 0

def rumbo2azimut_df(df_datos: pd.DataFrame) -> pd.DataFrame:
    """
    Calcula el azimut y buzamiento a partir del rumbo y el buzamiento
    :param df_datos: dataframe con rumbo_ang, rumbo_dir, buzamiento_ang y
    buzamiento_dir
    :return: dataframe. alfa (Azimut) y beta (Buzamiento)
    """
    df = pd.DataFrame()
    df['alfa'] = df_datos.apply(rumbo2azimut, axis=1)
    df['beta'] = df_datos['buzamiento_ang']
    return df

def agrupar_datos(dataframe: pd.DataFrame) -> pd.DataFrame:
    # Organizar datos por azimut
    sorted_df = dataframe.sort_values('alfa', ascending=True)
    # Hallar la diferencia entre los alfa_i y alfa_cero
    alfa_cero = sorted_df.iloc[0]['alfa']
    sorted_df['d1'] = sorted_df['alfa'] - alfa_cero
    familia = 1
    # Hallar la diferencia entre el dato actual y el anterior
    d2 = sorted_df['d1']
    sorted_df['d2'] = sorted_df['d1'] - d2.shift(1)
    sorted_df['familia'] = 0
    # Si la diferencia es mayor a 20 entonces asignar a nueva familia
    for idx, row in sorted_df.iterrows():
        if row['d2'] > 20:
            familia = familia + 1
            sorted_df.at[idx, 'familia'] = familia
    # print('\nAgrupados\n', sorted_df)
    return sorted_df

def cosenos_directores(datos_df: pd.DataFrame) -> pd.DataFrame:
    """
    Calcula los cosenos directores del azimut y buzamiento

```

```

:param datos_df: dataframe. alfa y beta en grados, Familia
:return: dataframe. l, m, n, Familia
"""
cos_directores = pd.DataFrame()
# Cambiar de grados a radianes
datos_rads_df = np.deg2rad(datos_df[['alfa', 'beta']])
# Calcular cosenos directores
cos_directores['l'] = np.cos(datos_rads_df['alfa']) *
np.cos(datos_rads_df['beta'])
cos_directores['m'] = np.cos(datos_rads_df['beta']) *
np.sin(datos_rads_df['alfa'])
cos_directores['n'] = np.sin(datos_rads_df['beta'])
cos_directores['familia'] = datos_df['familia']
# print('\nCosenos directores\n', cos_directores)
return cos_directores

def sumatoria_cos_directores(cos_directores_df: pd.DataFrame) ->
pd.DataFrame:
# Agrupar datos por familia
group_directores = cos_directores_df.groupby(by='familia')
# Sumar los cosenos de cada familia
suma_directores = group_directores.sum()
# Agregar el número de datos de cada familia
suma_directores['N'] = group_directores.size()
# print('\nSumatoria cos directores\n', suma_directores)

return suma_directores

def calculo_r(suma_directores: pd.DataFrame) -> pd.DataFrame:
r_df = pd.DataFrame()
# Elevar la suma de los cosenos directores al cuadrado
r_df = suma_directores[['l', 'm', 'n']] ** 2
# Calcular la raíz de la suma de los cuadrados de cada familia
r_df['R'] = np.sqrt(r_df[['l', 'm', 'n']].sum(axis=1))
r_df['N'] = suma_directores['N']
# print('\nRs\n', r_df)
return r_df

def calculo_representativos(r_df: pd.DataFrame, suma_directores_df:
pd.DataFrame) -> pd.DataFrame:
# Cargar l, m, n en representativos
representativos_df = suma_directores_df
# Cargar R de r_df en representativos
representativos_df['R'] = r_df['R']
# Dividir l, m, n de cada familia por el R de cada familia
representativos_df = representativos_df[['l', 'm',
'n']].div(representativos_df['R'], axis='index')
# Calcular representativos
representativos_df['beta'] = np.arcsin(representativos_df['n'])

def calcular_alfa_f(row):
af = np.arccos(row['l'] / np.cos(row['beta']))
if row['m'] >= 0:

```

```

        return af
    else:
        return -af

    representativos_df['alfa'] =
representativos_df.apply(calcular_alfa_f, axis=1)

    # Cambiar alfa y beta a grados
    representativos_df[['alfa', 'beta']] =
np.rad2deg(representativos_df[['alfa', 'beta']])
    representativos_df['alfa_p'] =
representativos_df['alfa'].apply(alfa_a_alfa_p)
    representativos_df['beta_p'] =
representativos_df['beta'].apply(beta_a_beta_p)
    representativos_df['N'] = r_df['N']
    representativos_df['R'] = r_df['R']
    # print('\nRepresentativos\n', representativos_df[['alfa_p',
'beta_p']])

    return representativos_df

def calcular_k(rep_df: pd.DataFrame) -> pd.DataFrame:
    kf_df = pd.DataFrame()
    kf_df['K'] = (rep_df['N'] - 1) / (rep_df['N'] - rep_df['R'])
    # print('\nK_f\n', kf_df)
    return kf_df

def alfa_a_alfa_p(alfa):
    if (alfa + 180) > 360:
        alfa_p = alfa - 180
    else:
        alfa_p = alfa + 180
    return alfa_p

def beta_a_beta_p(beta):
    return 90 - beta

```

utils_cinematico/friccion.py

```

from scipy.stats import truncnorm
import numpy as np
import pandas as pd

def fric(red: pd.DataFrame, phi: pd.DataFrame) -> pd.DataFrame:
    mean = phi.mean().iloc[0]
    std = phi.std().iloc[0]
    minim = phi['phi'].min()
    # maxim = phi['phi'].max()
    maxim = 90
    a, b = (minim - mean) / std, (maxim - mean) / std
    pdf = red[['alfa', 'beta']]
    pdf['pdf'] = truncnorm.cdf(red['beta'], a, b, loc=mean, scale=std)

```

```
# print(pdf)
return pdf
```

utils_cinematico/paleom.py

```
import numpy as np
import pandas as pd
```

```
def paleom(red_df: pd.DataFrame, alfa_p, beta_p, k) -> pd.DataFrame:
    """
    Calcula la probabilidad usando la función de paleomagnetismo
    :param red_df: dataframe. alfa, beta, Red polar
    :param alfa_p: alfa de la familia
    :param beta_p: beta de la familia
    :param k: coeficiente de la familia
    :return: dataframe. alfa, beta, fp
    """
    df = red_df[['alfa', 'beta']]
    cos_dir_flia = cosenos_directores(pd.DataFrame({'alfa': alfa_p,
    'beta': beta_p}, index=[0]))
    cos_dir = cosenos_directores(red_df)
    cos_theta = (cos_dir * cos_dir_flia.iloc[0]).sum(axis=1)
    df['fp'] = k / (4 * np.pi * np.sinh(k)) * np.exp(k * cos_theta)
    min_fp = df['fp'].max() * 0.01
    df.loc[df['fp'] <= min_fp, 'fp'] = 0
    return df
```

```
def cosenos_directores(datos_df: pd.DataFrame) -> pd.DataFrame:
    cos_directores = pd.DataFrame()
    datos_rads_df = np.deg2rad(datos_df)
    cos_directores['l'] = np.cos(datos_rads_df['alfa']) *
    np.cos(datos_rads_df['beta'])
    cos_directores['m'] = np.cos(datos_rads_df['beta']) *
    np.sin(datos_rads_df['alfa'])
    cos_directores['n'] = np.sin(datos_rads_df['beta'])
    return cos_directores
```

```
def areas(red_df: pd.DataFrame) -> pd.DataFrame:
    pass
```

utils_cinematico/ut_cunia.py

```
from app.controlador_datos import DataCtl, DataCinematica,
DataCinematicaFamilias, Labels
import utils.data_envolvente as denv
import utils_cinematico.familias as fam
import utils_cinematico.paleom as pal
import utils_cinematico.friccion as frc
import utils.data_utils as dut
```

```
def cinematica_cunia(datos: DataCtl, datos_out: DataCinematica):
    print(['Cinématica por cuña', 'Calculando representativos de
    familias'])
    familia_df = fam.interseccion_familias(datos.datos_rumbos)
```

```

print('[Cinemática por cuña]', 'Calculando cono de fricción')
fric_df = frc.fric(datos.red_polar, datos.datos_friccion)
areas_df = dut.areas(datos.red_polar)
datos_out.friccion_df = fric_df[['alfa', 'beta', 'pdf']]

for idx, row in familia_df.iterrows():
    print(f'Intersección {idx}'.center(100, '-'))
    print(f'[Cinemática por cuña]', 'Calculando probabilidad con la
función paleomagnética')
    paleom_df = pal.paleom(datos.red_polar, row['alfa_p'],
row['beta_p'], row['k'])
    print('[Cinemática por cuña]', 'Buscando beta de la envolvente')
    beta_envolvente = env.beta_min(datos.red_polar, paleom_df,
datos.cinemat_cunnia.ang_talud)
    print('Beta de la envolvente:', beta_envolvente)
    env_df = env.envolvente3d(datos.red_polar, beta_envolvente,
datos.cinemat_cunnia.ang_talud)
    area_envolvente = (env_df['env'] * areas_df['a']).sum()
    print('Área envolvente:', area_envolvente)
    print('[Cinemática por cuña]', 'Calculando probabilidad
conjunta')
    paleom_df['fp_a'] = paleom_df['fp'] * areas_df['a']
    constante = paleom_df['fp_a'].sum()
    paleom_df['fp_a'] = paleom_df['fp_a'] / constante
    paleom_df['conj'] = paleom_df['fp_a'] * env_df['env'] *
fric_df['pdf']
    prob_cojunta = paleom_df['conj'].sum()
    max_level = paleom_df['fp'].max()

    labels = Labels()

    labels.titulo = f'Zona de falla critica falla en cuña por {idx}'
    labels.titulo_2 = f'Probabilidad conjunta falla en cuña por
{idx}'

    datos_familia = DataCinematicaFamilias(prob_cojunta, max_level,
env_df[['alfa', 'beta', 'env']],
paleom_df[['alfa', 'beta',
'fp', 'conj']], idx, labels)

    print('Probabilidad conjunta:', prob_cojunta)

    if datos_out.calculos_familia is None:
        datos_out.calculos_familia = [datos_familia]
    else:
        datos_out.calculos_familia.append(datos_familia)

```

utils_cinematico/ut_planar.py

```

from app.controlador_datos import DataCtl, DataCinematica,
DataCinematicaFamilias, Labels
import utils_cinematico.paleom as pal
import utils.data_envolvente as env
import utils.data_utils as dut
import utils_cinematico.familias as fam
import utils_cinematico.friccion as frc

```

```

def cinematica_planar(datos: DataCtl, datos_out: DataCinematica):
    print('[Cinemática planar]', 'Calculando representativos de
familias')
    familia_df = fam.data_familias(datos.datos_rumbos)
    print('[Cinemática planar]', 'Calculando cono de fricción')
    fric_df = frc.fric(datos.red_polar, datos.datos_friccion)
    datos_out.friccion_df = fric_df[['alfa', 'beta', 'pdf']]
    areas_df = dut.areas(datos.red_polar)

    for idx, row in familia_df.iterrows():
        print(f'Familia {idx}'.center(100, '-'))
        print(f'[Cinemática planar]', 'Calculando probabilidad con la
función paleomagnética')
        paleom_df = pal.paleom(datos.red_polar, row['alfa_p'],
row['beta_p'], row['k'])
        print('[Cinemática planar]', 'Buscando beta de la envolvente')
        beta_envolvente = denv.beta_min(datos.red_polar, paleom_df,
datos.cinemat_planar.ang_talud)
        print('Beta de la envolvente:', beta_envolvente)
        env_df = denv.envolvente3d(datos.red_polar, beta_envolvente,
datos.cinemat_planar.ang_talud)
        area_envolvente = (env_df['env'] * areas_df['a']).sum()
        print('Área envolvente:', area_envolvente)
        print('[Cinemática planar]', 'Calculando probabilidad conjunta')
        paleom_df['fp_a'] = paleom_df['fp'] * areas_df['a']
        constante = paleom_df['fp_a'].sum()
        paleom_df['fp_a'] = paleom_df['fp_a'] / constante
        paleom_df['conj'] = paleom_df['fp_a'] * env_df['env'] *
fric_df['pdf']
        prob_cojunta = paleom_df['conj'].sum()
        max_level = paleom_df['fp'].max()

        labels = Labels()

        labels.titulo = f'Zona de falla critica falla planar por F{idx}'
        labels.titulo_2 = f'Probabilidad conjunta de falla planar por
F{idx}'

        data_familia = DataCinematicaFamilias(prob_cojunta, max_level,
env_df[['alfa', 'beta', 'env']],
paleom_df[['alfa', 'beta',
'fp', 'conj']], idx, labels)

        if datos_out.calculos_familia is None:
            datos_out.calculos_familia = [data_familia]
        else:
            datos_out.calculos_familia.append(data_familia)

```

utils_cinematico/ut_volteo.py

```

from app.controlador_datos import DataCtl, DataCinematica,
DataCinematicaFamilias, Labels
import utils_cinematico.paleom as pal
import utils.data_envolvente as denv
import utils_cinematico.familias as fam
import utils_cinematico.friccion as frc
import utils.data_utils as dut

```

```

def cinematica_volteo(datos: DataCtl, datos_out: DataCinematica):
    print('[Cinemática por volteo]', 'Calculando representativos de
familias')
    familia_df = fam.data_familias_volteo(datos.datos_rumbos)
    print('[Cinemática por volteo]', 'Calculando cono de fricción')
    fric_df = frc.fric(datos.red_polar, datos.datos_friccion)
    areas_df = dut.areas(datos.red_polar)

    datos_out.friccion_df = fric_df[['alfa', 'beta', 'pdf']]

    for idx, row in familia_df.iterrows():
        print(f'Familia {idx}'.center(100, '-'))
        print(f'[Cinemática por volteo]', 'Calculando probabilidad con la
función paleomagnética')
        paleom_df = pal.paleom(datos.red_polar, row['alfa_p'],
row['beta_p'], row['k'])
        print(f'[Cinemática por volteo]', 'Buscando beta de la
envolvente')
        beta_envolvente = env.beta_min(datos.red_polar, paleom_df,
datos.cinemat_volteo.ang_talud)
        print('Beta de la envolvente:', beta_envolvente)
        env_df = env.envolvente3d(datos.red_polar, beta_envolvente,
datos.cinemat_volteo.ang_talud)
        area_envolvente = (env_df['env'] * areas_df['a']).sum()
        print('Área envolvente:', area_envolvente)
        print(f'[Cinemática por volteo]', 'Calculando probabilidad
conjunta')
        paleom_df['fp_a'] = paleom_df['fp'] * areas_df['a']
        constante = paleom_df['fp_a'].sum()
        paleom_df['fp_a'] = paleom_df['fp_a'] / constante
        paleom_df['conj'] = paleom_df['fp_a'] * env_df['env'] *
fric_df['pdf']
        prob_cojunta = paleom_df['conj'].sum()
        max_level = paleom_df['fp'].max()

        labels = Labels()

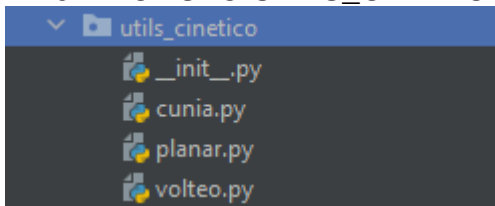
        labels.titulo = f'Zona de falla critica falla por volteo F{idx}'
        labels.titulo_2 = f'Probabilidad conjunta falla por volteo
F{idx}'

        calculo_familias = DataCinematicaFamilias(prob_cojunta,
max_level, env_df[['alfa', 'beta', 'env']],
paleom_df[['alfa',
'beta', 'fp', 'conj']], idx, labels)

    if datos_out.calculos_familia is None:
        datos_out.calculos_familia = [calculo_familias]
    else:
        datos_out.calculos_familia.append(calculo_familias)

```


5. MÓDULO UTILS_CINETICO



utils_cinetico/__init__.py

```
from .planar import cinetica_planar
from .volteo import cinetica_volteo
from .cunia import cinetica_cunia
```

utils_cinetico/cunia.py

```
import numpy as np
import pandas as pd
from scipy.stats import beta

from app.controlador_datos import DataCtl, DataOut, DataCinetica, Labels
import utils.data_utils as du
```

```
def cinetica_cunia(datos: DataCtl, datos_out: DataOut):
```

```
    prob_friccion = cinetica_cunia_friccion(datos,
datos_out.cinetica_cunia_friccion)
    prob_cohesion = cinetica_cunia_cohesion(datos,
datos_out.cinetica_cunia_cohesion)
    probabilidad = prob_cohesion * prob_friccion *
du.prob_acc_sismica(datos.cinetic_cunnia.t_retorno,
```

```
datos.cinetic_cunnia.periodo)
    datos_out.cinetica_cunia_cohesion.probabilidad = probabilidad
    datos_out.cinetica_cunia_friccion.probabilidad = probabilidad
```

```
def cinetica_cunia_friccion(datos: DataCtl, datos_out: DataCinetica):
    num_divisiones = 100
```

```
    # Cálculo de phi1 a partir de phi_2
    df = pd.DataFrame({'p2': datos.datos_friccion['phi']})
    df['tan_p2'] = np.tan(np.deg2rad(datos.datos_friccion['phi']))
    r_phi = datos.cinetic_cunnia.plano1.N *
np.tan(np.deg2rad(datos.cinetic_cunnia.plano1.phi)) \
    + datos.cinetic_cunnia.plano2.N *
np.tan(np.deg2rad(datos.cinetic_cunnia.plano2.phi))
    t_req = datos.cinetic_cunnia.plano1.c * datos.cinetic_cunnia.plano1.A
+ \
        datos.cinetic_cunnia.plano2.c * datos.cinetic_cunnia.plano2.A
+ r_phi
    df['tan_p1'] = - datos.cinetic_cunnia.plano2.N * df['tan_p2'] /
datos.cinetic_cunnia.plano1.N + \
        t_req * datos.cinetic_cunnia.n_phi * r_phi /
datos.cinetic_cunnia.plano1.N
    df['p1'] = np.rad2deg(np.arctan(df['tan_p1']))
```

```

# Calculo de los valores máximos y mínimos de phi_1
mean1 = df['p1'].mean()
std1 = df['p1'].std()
minim1 = mean1 - std1 * np.sqrt(28 / 4)
maxim1 = mean1 + std1 * np.sqrt(28 / 4)

# Calculo de los valores máximos y mínimos de phi2
mean2 = datos.datos_friccion['phi'].mean()
std2 = datos.datos_friccion['phi'].std()
minim2 = mean2 - std2 * np.sqrt(28 / 4)
maxim2 = mean2 + std2 * np.sqrt(28 / 4)

# Calculo de red p1, p2
step1 = (maxim1 - minim1) / num_divisiones
p1_range = np.arange(minim1, maxim1 + 0.9 * step1, step1)
step2 = (maxim2 - minim2) / num_divisiones
p2_range = np.arange(minim2, maxim2 + 0.9 * step2, step2)
tan_p1 = np.tan(np.deg2rad(p1_range))
tan_p2 = np.tan(np.deg2rad(p2_range))

index = pd.MultiIndex.from_product([p2_range, p1_range], names=['p2',
'p1'])
red_df = pd.DataFrame(index=index).reset_index()
red_df['x2'] = (red_df['p2'] - minim2) / (maxim2 - minim2)
red_df['x1'] = (red_df['p1'] - minim1) / (maxim1 - minim1)

# Calculo de tangentes de la red
red_df['tan_p1'] = np.tan(np.deg2rad(red_df['p1']))
red_df['tan_p2'] = np.tan(np.deg2rad(red_df['p2']))

# Calculo de distribuciones
red_df['beta1'] = beta.pdf(red_df['x1'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)
red_df['beta2'] = beta.pdf(red_df['x2'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)

# Matriz de fricción
red_df['prob'] = red_df['beta1'] * red_df['beta2']
constant = red_df['prob'].sum() * (step1 * step2)
red_df['prob'] = red_df['prob'] / constant
red_df = red_df.round(6)

# Selección de la probabilidad
red_df['prob_f'] = (red_df['tan_p1'] <= (-
datos.cinetic_cunnia.plano2.N * red_df['tan_p2'] +
t_req *
datos.cinetic_cunnia.n_phi * r_phi) /
datos.cinetic_cunnia.plano1.N) * red_df['prob']

datos_out.eje1, datos_out.eje2, datos_out.eje_z1, datos_out.eje_z2 =
'tan_p2', 'tan_p1', 'prob', 'prob_f'
datos_out.axes_df = pd.DataFrame({'tan_p2': tan_p2})
datos_out.axes2_df = pd.DataFrame({'tan_p1': tan_p1})
datos_out.red_df = red_df[['tan_p2', 'tan_p1', 'prob', 'prob_f']]
datos_out.probabilidad = red_df['prob_f'].sum() * step1 * step2

```

```

labels = Labels()

labels.titulo = 'Volumen de probabilidad final para ángulo de
fricción'
labels.titulo_2 = 'Volumen de probabilidad final para ángulo de
fricción'
labels.x_label = '$\tan(\phi_2)$'
labels.y_label = '$\tan(\phi_1)$'
labels.z_label = '$P(R_{\phi_1}) * P(R_{\phi_2})$'

datos_out.labels = labels

cun_data = datos.cinetic_cunnia

return red_df['prob_f'].sum() * step1 * step2

def cinetica_cunia_cohesion(datos: DataCtl, datos_out: DataCinetica):
    num_divisiones = 100

    # Cálculo de phi1 a partir de phi_2
    df = pd.DataFrame({'Cr2': datos.datos_cohesion['Cr2']})
    c_phi = datos.cinetic_cunnia.plano1.c * datos.cinetic_cunnia.plano1.A
+ \
        datos.cinetic_cunnia.plano2.c * datos.cinetic_cunnia.plano2.A
    r_phi = datos.cinetic_cunnia.plano1.N *
np.tan(np.deg2rad(datos.cinetic_cunnia.plano1.phi)) \
        + datos.cinetic_cunnia.plano2.N *
np.tan(np.deg2rad(datos.cinetic_cunnia.plano2.phi))
    t_req = c_phi + r_phi
    df['Cr1'] = (t_req * datos.cinetic_cunnia.n_c * c_phi -
datos.cinetic_cunnia.plano2.A * df['Cr2']) / \
        datos.cinetic_cunnia.plano1.A

    # Calculo de los valores máximos y mínimos de Cr1
    mean1 = df['Cr1'].mean()
    std1 = df['Cr1'].std()
    minim1 = mean1 - std1 * np.sqrt(28 / 4)
    maxim1 = mean1 + std1 * np.sqrt(28 / 4)

    # Calculo de los valores máximos y mínimos de Cr2
    mean2 = datos.datos_cohesion['Cr2'].mean()
    std2 = datos.datos_cohesion['Cr2'].std()
    minim2 = mean2 - std2 * np.sqrt(28 / 4)
    maxim2 = mean2 + std2 * np.sqrt(28 / 4)

    # Calculo de red Cr1, Cr2
    step1 = (maxim1 - minim1) / num_divisiones
    cr1_range = np.arange(minim1, maxim1 + 0.9 * step1, step1)
    step2 = (maxim2 - minim2) / num_divisiones
    cr2_range = np.arange(minim2, maxim2 + 0.9 * step2, step2)

    index = pd.MultiIndex.from_product([cr2_range, cr1_range],
names=['Cr2', 'Cr1'])
    red_df = pd.DataFrame(index=index).reset_index()
    red_df['x2'] = (red_df['Cr2'] - minim2) / (maxim2 - minim2)

```

```

red_df['x1'] = (red_df['Cr1'] - minim1) / (maxim1 - minim1)

# Calculo de distribuciones
red_df['beta1'] = beta.pdf(red_df['x1'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)
red_df['beta2'] = beta.pdf(red_df['x2'], datos.cinetic_cunnia.alfa,
datos.cinetic_cunnia.beta)

# Matriz de Cohesión
red_df['prob'] = red_df['beta1'] * red_df['beta2']
constant = red_df['prob'].sum() * (step1 * step2)
red_df['prob'] = red_df['prob'] / constant
red_df = red_df.round(6)

# Selección de la probabilidad
red_df['prob_f'] = (red_df['Cr1'] <= (-datos.cinetic_cunnia.plano2.A
* red_df['Cr2'] +
                                t_req *
datos.cinetic_cunnia.n_c * c_phi) / datos.cinetic_cunnia.plano1.A) \
    * red_df['prob']

# Guardar los datos de salida
datos_out.eje1, datos_out.eje2, datos_out.eje_z1, datos_out.eje_z2 =
'Cr2', 'Cr1', 'prob', 'prob_f'
datos_out.axes_df = pd.DataFrame({'Cr2': cr2_range})
datos_out.axes2_df = pd.DataFrame({'Cr1': cr1_range})
datos_out.red_df = red_df[['Cr2', 'Cr1', 'prob', 'prob_f']]
datos_out.probabilidad = red_df['prob_f'].sum() * step1 * step2

labels = Labels()

labels.titulo = 'Volumen de probabilidad final para cohesión'
labels.titulo_2 = 'Volumen de probabilidad final para cohesión'
labels.x_label = '$C_{r1}$'
labels.y_label = '$C_{r2}$'
labels.z_label = '$P(C_{r1}) * P(C_{r2})$'

datos_out.labels = labels

return red_df['prob_f'].sum() * step1 * step2

```

utils_cinetico/planar.py

```

import numpy as np
from scipy.stats import beta
import pandas as pd

from app.controlador_datos import DataCtl, DataCinetica, Labels
import utils.data_utils as du

def cinetica_planar(datos: DataCtl, datos_out: DataCinetica):
    # Construir los ejes de la red phi vs cr
    minim = datos.datos_friccion['phi'].min()
    maxim = datos.datos_friccion['phi'].max()
    step = 0.01
    datos.cinetic_planar.theta = 89 if datos.cinetic_planar.theta > 89

```

```

else datos.cinetic_planar.theta
    theta_rad = np.deg2rad(datos.cinetic_planar.theta)
    axes = pd.DataFrame({'phi': np.arange(minim, maxim + step, step)})
    axes['cr'] = calculo_cr(axes['phi'], theta_rad,
datos.cinetic_planar.h, datos.cinetic_planar.gamma)

    # Construir la red
    index = pd.MultiIndex.from_product([axes['phi'], axes['cr']])
    red_df = pd.DataFrame(index=index).reset_index()

    # Normalizar ejes de 0 a 1
    red_df['x_phi'] = (red_df['phi'] - minim) / (maxim - minim)
    minim = red_df['cr'].min()
    maxim = red_df['cr'].max()
    red_df['x_cr'] = (red_df['cr'] - minim) / (maxim - minim)

    # Calcular las distribuciones Beta
    red_df['pdf_phi'] = beta.pdf(red_df['x_phi'],
datos.cinetic_planar.alfa, datos.cinetic_planar.beta)
    red_df['pdf_cr'] = beta.pdf(red_df['x_cr'],
datos.cinetic_planar.alfa, datos.cinetic_planar.beta)

    # Calcular probabilidad conjunta
    red_df['prob'] = red_df['pdf_phi'] * red_df['pdf_cr']

    # Normalizar función de probabilidad conjunta
    constant = red_df['prob'].sum() * (step ** 2)
    red_df['prob_n'] = red_df['prob'] / constant

    # Calcular pr
    red_df['prob_f'] = (calculo_cr(red_df['phi'], theta_rad,
datos.cinetic_planar.h, datos.cinetic_planar.gamma)
    <= red_df['cr']) * red_df['prob_n']

    # Limpiar datos
    red_df = red_df.fillna(0)

    # Guardar datos de salida en datos_out
    probabilidad_conjunta = red_df['prob_f'].sum() * (step ** 2)
    datos_out.probabilidad = probabilidad_conjunta *
du.prob_acc_sismica(datos.cinetic_planar.t_retorno,
datos.cinetic_planar.periodo)
    datos_out.red_df = red_df[['phi', 'cr', 'prob_n', 'prob_f']]
    datos_out.axes_df = axes[['phi']]
    datos_out.axes2_df = axes[['cr']]
    datos_out.eje1, datos_out.eje2, datos_out.eje_z1, datos_out.eje_z2 =
'phi', 'cr', 'prob_n', 'prob_f'

    labels = Labels()

    labels.titulo = 'Volumen de probabilidad conjunta con funciones de
densidad Beta'
    labels.titulo_2 = 'Probabilidad conjunta y FS = 1, zona de falla'
    labels.x_label = '$\phi$'
    labels.y_label = 'Cr'

```

```

labels.z_label = '$P(\phi) * P(C_r)$'

datos_out.labels = labels

print('[Cinética Planar] Probabilidad conjunta: ',
probabilidad_conjunta)
print('[Cinética Planar] Probabilidad de falla: ',
datos_out.probabilidad)

def calculo_cr(phi: pd.DataFrame, theta_rad: float, h: float, gamma:
float):
    return gamma * h * np.sin(theta_rad) * (1 - np.tan(np.deg2rad(phi)) /
np.tan(theta_rad))

utils_cinetico/volteo.py
from scipy.stats import beta
import numpy as np
import pandas as pd

from app.controlador_datos import DataCtl, DataCinetica, Labels
import utils.data_utils as du

def cinetica_volteo(datos: DataCtl, datos_out: DataCinetica):
    # Datos de inclinación
    step = 0.1
    beta_df = pd.DataFrame({'beta': np.arange(0, 90 + step, step)})

    # Datos fricción
    maxim = datos.datos_friccion['phi'].max()
    minim = datos.datos_friccion['phi'].min()
    df = pd.DataFrame({'phi': np.arange(minim, maxim + step, step)})

    # Normalización de variables
    df['x_p'] = (df['phi'] - minim) / (maxim - minim)
    beta_df['x_b'] = beta_df['beta'] / 90
    theta = datos.cinetic_volteo.theta / 90 if datos.cinetic_volteo.theta
<= 90 else 1

    # Creación de la red
    index = pd.MultiIndex.from_product([df['x_p'], beta_df['x_b']])
    red_df = pd.DataFrame(index=index).reset_index()

    # Calculo de distribuciones de probabilidad
    red_df['f_b'] = beta.pdf(red_df['x_b'], datos.cinetic_volteo.alfa,
datos.cinetic_volteo.beta)
    red_df['f_p'] = beta.pdf(red_df['x_p'], datos.cinetic_volteo.alfa,
datos.cinetic_volteo.beta)

    # Calculo de probabilidad conjunta
    red_df['prob'] = red_df['f_p'] * red_df['f_b']
    constant = red_df['prob'].sum() * (step ** 2)
    red_df['prob_n'] = red_df['prob'] / constant

    # Calculo probabilidad conjunta dado beta

```

```

red_df['prob_f'] = (red_df['x_b'] <= theta - red_df['x_p']) *
red_df['prob_n']
volumen = red_df['prob_f'].sum() * (step ** 2)
prob_falla = du.prob_acc_sismica(datos.cinetic_volteo.t_retorno,
datos.cinetic_volteo.periodo) * volumen

datos_out.axes_df = df[['x_p']]
datos_out.axes2_df = beta_df[['x_b']]
datos_out.red_df = red_df[['x_p', 'x_b', 'prob_n', 'prob_f']]
datos_out.probabilidad = prob_falla
datos_out.eje1, datos_out.eje2, datos_out.eje_z1, datos_out.eje_z2 =
'x_p', 'x_b', 'prob_n', 'prob_f'

labels = Labels()

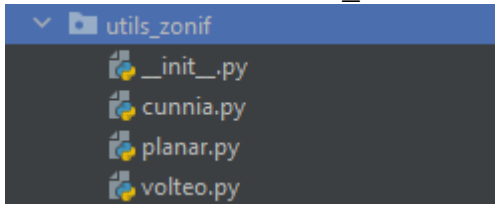
labels.titulo = 'Volumen probabilidad conjunta'
labels.titulo_2 = 'Probabilidad conjunta'
labels.x_label = '$\phi$'
labels.y_label = '$\beta$'
labels.z_label = '$P(\phi) * P(\beta)$'

datos_out.labels = labels

print(['Cinética por volteo] Probabilidad conjunta para beta',
volumen)
print(['Cinética por volteo] Probabilidad de falla', prob_falla)

```

6. MÓDULO UTILS_ZONIF



utils_zonif/__init__.py

```

from .planar import zonif_planar
from .volteo import zonif_volteo
from .cunnia import zonif_cunnia

```

utils_zonif/cunnia.py

```

import pandas as pd

```

```

from app.controlador_datos import DataCtl, DataOut
from utils_cinematico.familias import interseccion_familias,
data_familias

```

```

def zonif_cunnia(datos: DataCtl, datos_out: DataOut):
intersecciones = interseccion_familias(datos.datos_rumbos)
datos_out.zonif_cunnia.familias = data_familias(datos.datos_rumbos)
familias = datos_out.zonif_cunnia.familias['alfa_p']
inter_validas = pd.DataFrame()

```

```

for idx, row in intersecciones.iterrows():
    indices = str(idx).split('-')
    indice_1 = int(indices[0])
    indice_2 = int(indices[1])
    if familias[indice_1] > familias[indice_2]:
        if familias[indice_1] > row['alfa_p'] > familias[indice_2]:
            row['alfa_max'] = familias[indice_1] - 20
            row['alfa_min'] = familias[indice_2] + 20
            inter_validas = inter_validas.append(row)
        else:
            break
    else:
        if familias[indice_2] > row['alfa_p'] > familias[indice_1]:
            row['alfa_max'] = familias[indice_2] - 20
            row['alfa_min'] = familias[indice_1] + 20
            inter_validas = inter_validas.append(row)
        else:
            break

datos_out.zonif_cunnia.intersec = inter_validas

```

utils_zonif/planar.py

```

from app.controlador_datos import DataCtl, DataOut
import utils_cinematico.familias as fam

def zonif_planar(datos: DataCtl, datos_out: DataOut):
    familia_df = fam.data_familias(datos.datos_rumbos)
    datos_out.zonif_planar.repr_df = familia_df

```

utils_zonif/volteo.py

```

from app.controlador_datos import DataOut, DataCtl
from utils_cinematico.familias import data_familias_volteo

def zonif_volteo(datos: DataCtl, datos_out: DataOut):
    repr_df = data_familias_volteo(datos.datos_rumbos)
    repr_df['alfa_p'] = repr_df['alfa_p'] + 180
    datos_out.zonif_volteo.repr_df = repr_df

```

7. MAIN.PY

```

import sys

from app.controlador import Controller

# Punto de entrada de la aplicación
if __name__ == '__main__':
    app = Controller(sys.argv)
    sys.exit(app.exec_())

app/dlg_progreso/controller.py
from PySide2 import QtWidgets as Qtw

from .ui_dialogo_progreso import Ui_Dialog

```



```
class ProgressWindow(Qtw.QDialog):
    def __init__(self, parent: Qtw.QWidget, minimum, maximum,
message=''):
    super(ProgressWindow, self).__init__(parent)
    self.ui = Ui_Dialog()
    self.ui.setupUi(self)
    self.setFixedSize(self.size())
    self.ui.progressBar.setMinimum(minimum)
    self.ui.progressBar.setMaximum(maximum)
    self.update_progress(10, message)

    def update_progress(self, value, message):
        self.ui.progressBar.setValue(value)
        self.ui.label.setText(message)

    def close(self) -> bool:
        return super(ProgressWindow, self).close()
```