



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA CIVIL**

TEMA:

**Cálculo de curvas de fragilidad mediante código de Python,
utilizando el Método de Espectro de Capacidad (CSM).**

AUTORES:

**Ajila Ajila, Mateo Sebastián
Molina Arias, Carlos Alfredo**

**Trabajo de titulación previo a la obtención del título de
INGENIERO CIVIL**

TUTOR:

Ing. Hernández Barredo, Jaime Francisco, M.Sc.

Guayaquil, Ecuador

03 de septiembre del 2024



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA CIVIL**

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación fue realizado en su totalidad por **Ajila Ajila, Mateo Sebastián y Molina Arias, Carlos Alfredo**, como requerimiento para la obtención del título de **Ingeniero Civil**.

TUTOR

f.  _____

Ing. Hernández Barredo, Jaime Francisco, M.Sc.

DIRECTORA DE LA CARRERA

f.  _____

Ing. Alcívar Bastidas, Stefany Esther M.Cs.

Guayaquil, a los 3 días del mes de septiembre del año 2024



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA CIVIL

DECLARACIÓN DE RESPONSABILIDAD

Nosotros, **Ajila Ajila, Mateo Sebastián y Molina Arias, Carlos
Alfredo**

DECLARAMOS QUE:

El Trabajo de Titulación, **Cálculo de curvas de fragilidad mediante código de Python, utilizando el Método de Espectro de Capacidad (CSM)**, previo a la obtención del título de **Ingeniero Civil**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías.

Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 3 días del mes de septiembre del año 2024

AUTOR:

f.

Ajila Ajila, Mateo Sebastián

f.

Molina Arias, Carlos Alfredo



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA CIVIL

AUTORIZACIÓN

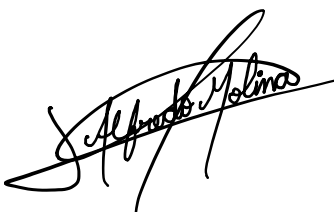
Nosotros, **Ajila Ajila, Mateo Sebastián y Molina Arias,
Carlos Alfredo**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Cálculo de curvas de fragilidad mediante código de Python, utilizando el Método de Espectro de Capacidad (CSM)**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 3 días del mes de septiembre del año 2024

AUTOR:

f. 
Ajila Ajila, Mateo Sebastián

f. 
Molina Arias, Carlos Alfredo

REPORTE COMPILATIO



AJILA_MATEO_MOLINA_CARLOS_FINAL_TESIS



Nombre del documento: AJILA_MATEO_MOLINA_CARLOS_FINAL_TESIS.docx ID del documento: 5d37d797673e536dc017abd83bfaf82c865605 Tamaño del documento original: 3.83 MB Autores: []	Depositante: Clara Catalina Glas Cevallos Fecha de depósito: 21/8/2024 Tipo de carga: interface fecha de fin de análisis: 21/8/2024	Número de palabras: 24.577 Número de caracteres: 183.294
---	--	---

Ubicación de las similitudes en el documento:



Fuentes de similitudes

Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	www.scielo.org/ve/ Metodología para la estimación de daños por sismos en edific... http://www.scielo.org/ve/scielo.php?script=sci_arttext&pid=50798-40652013000200007&lng=en&tlw... 43 fuentes similares	1%		Palabras idénticas: 1% (280 palabras)
2	www.scipedia.com https://www.scipedia.com/wd/fmages/2/22/Draft_Content_595784990Agudelo_Lopez.pdf 19 fuentes similares	< 1%		Palabras idénticas: < 1% (272 palabras)
3	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/19911/1/T-UCSG-PRE-ING-IC-444.pdf 13 fuentes similares	< 1%		Palabras idénticas: < 1% (228 palabras)
4	upcommons.upc.edu http://upcommons.upc.edu/bitstream/mh/andie/2/117793542/04CAPITULO3.pdf?sequence=4 17 fuentes similares	< 1%		Palabras idénticas: < 1% (221 palabras)
5	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/11168/3/T-UCSG-PRE-ING-IC-251.pdf.txt 8 fuentes similares	< 1%		Palabras idénticas: < 1% (192 palabras)

Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	Documento de otro usuario #d2641 El documento proviene de otro grupo	< 1%		Palabras idénticas: < 1% (39 palabras)
2	upcommons.upc.edu http://upcommons.upc.edu/bitstream/mv211795825/1/T-RDUG1de1.pdf	< 1%		Palabras idénticas: < 1% (40 palabras)
3	repositorio.uta.edu.ec https://repositorio.uta.edu.ec/bitstream/123456789/34103/1/Tesis I.C. 1552 - Navarrete López PabL...	< 1%		Palabras idénticas: < 1% (34 palabras)
4	dspace.uazuay.edu.ec http://dspace.uazuay.edu.ec/bitstream/datos/11256/3/16793.pdf.txt	< 1%		Palabras idénticas: < 1% (37 palabras)
5	tesis.pucp.edu.pe https://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/12488/LOPEZ_OTNIANO_ROD...	< 1%		Palabras idénticas: < 1% (39 palabras)

Fuentes ignoradas Estas fuentes han sido retiradas del cálculo del porcentaje de similitud por el propietario del documento.

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/5384/3/T-U-CSG-PRE-ING-IC-152.pdf.txt	2%		Palabras idénticas: 2% (560 palabras)
2	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/5384/1/T-U-CSG-PRE-ING-IC-152.pdf	2%		Palabras idénticas: 2% (545 palabras)
3	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/5384/1/T-U-CSG-PRE-ING-IC-152.pdf	2%		Palabras idénticas: 2% (545 palabras)
4	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/5384/1/T-U-CSG-PRE-ING-IC-152.pdf	2%		Palabras idénticas: 2% (545 palabras)

f.

Ing. Hernández Barredo, Jaime Francisco,

AGRADECIMIENTO

Quiero dedicar este espacio a mis padres, quienes han sido mi fuente de inspiración, fortaleza y motivación en cada paso que he dado. A ustedes Cesar Augusto Ajila Camacho y Gladys Aidé Ajila Freire, mi padre y madre, les debo todo lo que soy y todo lo que he logrado hasta la fecha.

Gracias, papá por enseñarme el valor del esfuerzo y la perseverancia, por tu ejemplo de trabajo duro y por tus sabios consejos que siempre me han guiado en momentos difíciles. Mamá, gracias por tu amor incondicional, por ser mi refugio en los días malos y por creer en mí cuando yo mismo dudaba.

Agradezco a mi hermano mayor Cesar, por cuidarme y preocuparse por mí todos estos años, porque sé, que, si tengo problemas, él estará ahí para brindarme una mano, y a mi hermana menor Sofía, que con sus ocurrencias siempre logra sacarme una sonrisa.

A mis abuelos, por llamarme siempre "ingeniero" y por motivarme a seguir adelante con mis estudios, a mis tíos y primos por su constante apoyo durante mis años universitarios. Sus consejos, su compañía y el cariño que me han brindado han sido muy importantes para conseguir este logro.

Gracias a mis amigos por los años compartidos, por estar en las alegrías y en las tristezas. Han sido un apoyo constante, y de cada uno he aprendido valiosas lecciones. A Ale, por brindarme su cariño, acompañarme, apoyarme, y motivarme todos los días a cumplir con mis objetivos.

Le agradezco a mi tutor, el Ing. Jaime Hernández, por su guía y apoyo durante este proceso. Sus consejos fueron fundamentales para el desarrollo de esta tesis, además de agradecerle por su dedicación y compromiso.

Agradezco a Dios por permitirme cumplir esta importante meta universitaria. Me ha guiado en cada paso del camino, dándome la fuerza, la constancia y el ánimo necesario para llegar hasta aquí. Este logro es una muestra de su amor y gracia en mi vida.

Mateo Sebastián Ajila Ajila

DEDICATORIA

Este logro se lo dedico a mi familia, en especial a mis padres por su apoyo incondicional, por sus consejos, por siempre creer en mí, y confiar en el proceso, por enseñarme a ser una persona de bien, hoy soy lo que soy gracias a ellos, y sé que están orgullosos de mí. Ambos son el soporte sobre el que voy construyo mis sueños, y este logro no hubiera sido posible sin el sacrificio y apoyo que siempre me han brindado. Les dedico esta tesis con todo mi amor y agradecimiento, porque cada uno de mis logros también son los suyos.

También a mis seres queridos, que, a pesar de no compartir la misma sangre, han estado conmigo día a día, motivándome y apoyándome a seguir adelante. Les dedico este logro por las madrugadas en las que me acompañaron, por estar presentes en los días tristes y en los malos, pero también en los buenos. Este logro es para ustedes.

Mateo Sebastián Ajila Ajila

AGRADECIMIENTO

En primer lugar, agradecer a Dios por permitirme llegar a este punto en mi vida, porque Él es testigo de lo que cada uno de nosotros lucha para poder conseguir un sueño, un objetivo y una meta. También agradecer a nuestra Madre Santísima María la Virgen, la cual me ha protegido y ayudado para seguir adelante en mi preparación académica, profesional y personal.

Como no agradecer a los seres que me dieron la vida, lo máspreciado que tengo en este mundo, a mi madre Teresita de Jesús Arias Suárez y a mi padre Carlos Manuel Molina, gracias a ellos he podido terminar varias etapas de mi vida; me han otorgado los valores y la fuerza necesaria para convertirme en el hombre que soy ahora. Gracias por siempre estar presente, gracias por cuidarme, gracias por la mejor herencia que existe, no saben lo mucho que los amo y apreció, cada reconocimiento que me ha sido otorgado es gracias a ustedes a ser mi inspiración para siempre ser mejor que el día anterior.

Agradecer a mi hermana Diana Carolina Molina Arias, gracias por la paciencia y por apoyarme en todo, si la vida me hubiera dado la posibilidad de elegir qué clase de hermana tener, sin duda alguna te escogería a ti en esta y en cualquier vida. También agradecer a mi familia que siempre han estado dispuestos ayudar, particularmente a mi tía Ana Matilde Molina, gracias a ella por su apoyo incondicional y como no agradecer a los que ya no se encuentran en este mundo, en especial a mi Abuelita Mercedes Suárez o como yo le llamaba “Abuelita Miche”, donde se encuentre se lo orgullosa que está de mí.

A mi tutor, el Ing. Jaime Hernández, gracias por la confianza que me tuvo para el desarrollo de este trabajo. También agradecerle por brindarme la oportunidad de ser parte de SismoIngeniería como pasante y como no mencionar a los “Sismoslaves”, gracias por todo.

Por último y no menos importante gracias a mis amigos y conocidos que he tenido a lo largo de mi vida con cada uno de ustedes he aprendido algo y me han aportado y/o influenciado en mi vida, gracias por estar presente.

Carlos Alfredo Molina Arias

DEDICATORIA

Dedico este logro a mis padres, en especial a mi Padre el cual siempre ha sido mi modelo para seguir, saber todo lo que ha luchado en su vida para estar aquí, me inspiró y me dio a la fuerza necesaria para poder afrontar cualquier adversidad y no darme por vencido. Te lo dedico papá, el sueño que algún tuviste lo hicimos realidad, prometo no defraudarte a ti ni a mamá y enorgullecerlos más de lo que están.

Carlos Alfredo Molina Arias



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA CIVIL**

f. 

Ing. Hernández Barredo, Jaime Francisco, M.Sc.

TUTOR

TRIBUNAL DE SUSTENTACIÓN

f. 

Ing. Alcívar Bastidas, Stefany, M. Sc.

DIRECTORA DE CARRERA

f. 

Ing. Ponce Vazquez, Guillermo, M. Sc.

COORDINADOR DEL ÁREA

**JOSE ANDRES
BARROS
CABEZAS**

Firmado digitalmente por JOSE ANDRES BARROS
CABEZAS
DN: C=EC, O=SECURITY DATA S.A. 1, OU=ENTIDAD
DE CERTIFICACION DE INFORMACION,
SERIALNUMBER=030820191013, CN=JOSE ANDRES
BARROS CABEZAS
Razón: Soy el autor de este documento
Ubicación:
Fecha: 2024.09.09 09:27:34-05'00'
Foxit PDF Reader Versión: 2024.2.3

f. _____

Ing. Barros Cabezas, José PhD.

OPONENTE

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	2
i. ANTECEDENTES	2
ii. OBJETIVO GENERAL	5
iii. OBJETIVOS ESPECÍFICOS	5
iv. ALCANCE	5
CAPÍTULO I: MARCO TEÓRICO	7
1.1 AMENAZA SÍSMICA	7
1.2 VULNERABILIDAD SÍSMICA	8
1.3 RIESGO SÍSMICO	10
1.4 MÉTODO DE NEWMARK	11
1.5 MÉTODO DEL ESPECTRO DE CAPACIDAD (CSM)	12
1.5.1 ESPECTRO DE ACELERACIÓN VS DESPLAZAMIENTO FORMATO ADRS	14
1.5.2 CURVA DE CAPACIDAD Y ESPECTRO DE CAPACIDAD 15	
1.5.3 CURVA BILINEAL	17
1.5.4 PUNTO DE DESEMPEÑO	18
1.6 MÉTODOLOGÍAS DEL ESPECTRO DE CAPACIDAD	19
1.6.1 FEMA 440	19
1.6.2 ASCE 41-17	27
1.7 INGENIERÍA SÍSMICA BASADA EN DESEMPEÑO SÍSMICO 30	
1.8 UMBRALES DE DAÑO	32

1.9	CURVAS DE FRAGILIDAD.....	33
1.9.1	MÉTODO ANALÍTICO	35
CAPÍTULO II: DESARROLLO.....		37
2.1	Método de los coeficientes ASCE 41-17.....	37
2.1.1	ARCHIVOS DE ENTRADA – “INPUT”	37
2.1.2	REGISTROS SÍSMICOS	37
2.1.3	DATOS DEL ANÁLISIS PUSHOVER	39
2.1.4	CURVA BILINEAL	40
2.1.5	PUNTOS DE DESEMPEÑO ASCE 41-17	44
2.1.6	CURVAS DE FRAGILIDAD ASCE 41-17	45
2.2	FEMA440 Método C (gráfico).	47
2.2.1	PUNTOS DE DESEMPEÑO FEMA 440 MÉTODO C....	47
2.2.2	CURVAS DE FRAGILIDAD FEMA440 Método C	50
CAPÍTULO III: RESULTADOS.....		51
3.1	Resultados método ASCE 41-17	51
3.1.1	Curva de Capacidad	51
3.1.2	Análisis de la Curva de capacidad.....	52
3.1.3	Representación Bilineal de la Curva de capacidad formato ADRS	54
3.1.4	Análisis para obtención de puntos de desempeño Curva de Capacidad ASCE41-17.....	55
3.1.5	Puntos de desempeño ASCE 41-17 para Curva de Capacidad	57

3.1.6	Desplazamiento Espectral S_d (cm) vs Desplazamiento Objetivo δ_t Curva de capacidad y Análisis para curva de Frágil de la Curva de capacidad ASCE41-17.....	60
3.1.15	Curvas de Fragilidad ASCE41-17.....	63
3.2	Resultados por FEMA440 método C	64
3.2.1	Análisis para obtención de puntos de desempeño Curva de Capacidad FEMA440 Método C	65
3.2.2	Puntos de desempeño FEMA440 Método C para Curva de Capacidad	65
3.2.3	Desplazamiento Espectral S_d (cm) vs S_d Performance Point (cm) Curva de Capacidad FEMA440 Método C	69
3.2.4	Curvas de Fragilidad FEMA440 Método C	71
3.3	Comparación de Curvas de fragilidad con estudios anteriores	73
4	CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES.....	77
4.1	Conclusiones	77
4.2	Recomendaciones	79
5	REFERENCIAS BIBLIOGRÁFICAS.....	80
6	Anexos.....	84
6.1	Anexo 1 Curva de fragilidad usando metodología ASCE41-17	84
6.2	Anexo 2 Curva de fragilidad usando metodología FEMA440 alternativa C	109

ÍNDICE DE FIGURAS

Figura 1. Diagrama de metodología del espectro de capacidad.....	14
Figura 2. Representación de la Curva Bilineal.....	18
Figura 3. Intersección del espectro de capacidad y espectro de demanda conocido como “Punto de desempeño”. (ATC – 40, 1996).....	19
Figura 4. Periodo inicial, Curva bilineal, Puntos de interés.....	22
Figura 5. Máximo desplazamiento esperado utilizando el Método A.....	24
Figura 6. Máximo desplazamiento esperado utilizando el Método B.....	25
Figura 7. Máximo desplazamiento esperado utilizando el Método C.....	26
Figura 8. Curva idealizada de Fuerza vs Desplazamiento.....	28
Figura 9. Representación de las curvas de fragilidad según el estado de daño.	35
Figura 10. Metodologías para calcular curvas de fragilidad.....	36
Figura 11 Formato para cada archivo de texto correspondiente a cada registro sísmico.....	38
Figura 12. Formato Curva de Capacidad (PushOver).....	39
Figura 13. Número de punto DU-AU.....	41
Figura 14. Obtención de coordenadas D_{yi} , A_{yi}	42
Figura 15. Análisis de la representación bilineal.....	42
Figura 16. Análisis para conversión ADRS.....	43
Figura 17. Representación Bilineal formato ADRS.....	43
Figura 18. Análisis para Desplazamiento Objetivo δ_t (cm).	44
Figura 19. Desplazamiento Espectral S_d (cm) vs Desplazamiento Objetivo δ_t (cm).....	45

Figura 20. Análisis para curvas de fragilidad.	46
Figura 21. Curva de Fragilidad método de coeficientes propuesto por el ASCE41-17.	46
Figura 22. Análisis para reducción de registro sísmico.	47
Figura 23. Punto de desempeño propuesto por el FEMA440 en el Método C.	48
Figura 24. Desplazamiento Espectral Sd (cm) vs Sd Performance Point (cm).	49
Figura 25. Curva de Fragilidad método de coeficientes propuesto por FEMA 440 Método C.	50
Figura 26. Curva de Capacidad 1 para su respectivo análisis.	51
Figura 27. Curva de Capacidad 2 para su respectivo análisis.	52
Figura 28. Análisis para la obtención de la Representación Bilineal de la curva de Capacidad 1.	53
Figura 29. Análisis para la obtención de la Representación Bilineal de la curva de Capacidad 2.	53
Figura 30. Curvas de Capacidad 1 junto con su Representación Bilineal en formato ADRS.	54
Figura 31. Curvas de Capacidad 2 junto con su Representación Bilineal en formato ADRS.	55
Figura 32 Coeficientes para la obtención de puntos de desempeño según el método ASCE 41-17 para la curva de capacidad 1	56
Figura 33 Coeficientes para la obtención de puntos de desempeño según el método ASCE 41-17 para la curva de capacidad 2	56
Figura 34. Desplazamiento objetivo (δ_t) del sismo ABBAR--V para Curva de Capacidad 1.	57

Figura 35. Desplazamiento objetivo (δ_t) del sismo AGY1N para Curva de Capacidad 1.....	58
Figura 36. Desplazamiento objetivo (δ_t) del sismo APO1N para Curva de Capacidad 1.....	58
Figura 37. Desplazamiento objetivo (δ_t) del sismo ABBAR--V para Curva de Capacidad 2.....	59
Figura 38. Desplazamiento objetivo (δ_t) del sismo AGY1N para Curva de Capacidad 2.....	59
Figura 39. Desplazamiento objetivo (δ_t) del sismo APO1N para Curva de Capacidad 2.....	60
Figura 40. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 1.....	61
Figura 41. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 1.	61
Figura 42. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 2.....	62
Figura 43. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 2.	62
Figura 44. Curva de Fragilidad obtenida para la Curva de Capacidad 1.	63
Figura 45. Curva de Fragilidad obtenida para la Curva de Capacidad 2.	64
Figura 46. Reducción del espectro de respuesta para cada caso de ductilidad de la Curva de Capacidad 1.	65
Figura 47. Reducción del espectro de respuesta para cada caso de ductilidad de la Curva de Capacidad 2.	65
Figura 48. Punto de desempeño del sismo ABBAR--V para Curva de Capacidad 1.....	66

Figura 49. Punto de desempeño del sismo AGY1N para Curva de Capacidad 1.....	66
Figura 50. Punto de desempeño del sismo APO1N para Curva de Capacidad 1.....	67
Figura 51. Punto de desempeño del sismo AES2N para Curva de Capacidad 2.....	67
Figura 52. Punto de desempeño del sismo AMA1N para Curva de Capacidad 2.....	68
Figura 53. Punto de desempeño del sismo APO1N para Curva de Capacidad 2.....	68
Figura 54. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 1.....	69
Figura 55. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 1.	70
Figura 56. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 2.....	70
Figura 57. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 2..	71
Figura 58. Curva de Fragilidad obtenida para la Curva de Capacidad 1.	72
Figura 59. Curva de Fragilidad obtenida para la Curva de Capacidad 2.	72
Figura 60 Curva de Fragilidad Modelo D3-Pushover" x" (sd).....	74
Figura 61. Curva de Fragilidad Modelo D11- Pushover "y".....	75
Figura 62 Gráfica para la obtención de curvas de Fragilidad para curva de capacidad con periodo de 0.27s.....	76
Figura 63 Curva de Fragilidad obtenida para la Curva de Capacidad con periodo de 0.27s.....	76

ÍNDICE DE TABLAS

Tabla 1. Factor de modificación Co.	29
Tabla 2. Factor de masa efectiva Cm..	29
Tabla 3. Niveles de daño y desempeño para edificios.....	32

RESUMEN

En el presente trabajo de titulación se desarrolló la obtención de curvas de fragilidad mediante dos códigos Python, utilizando dos metodologías, una propuesta por ASCE 41-17 conocida como el Método de los coeficientes y la otra propuesta por el FEMA 440, alternativa C, también conocido como “Método del espectro de Capacidad”. Ambos métodos son usados para la obtención de puntos de desempeño de una determinada estructura que se desee analizar.

En el desarrollo del proyecto se identificó que existe variación en el apartado necesario para la obtención de las curvas de fragilidad, ya que el método de los coeficientes muestra una tendencia lineal creciente, consecuentemente para el método propuesto por el FEMA 440, en el mismo apartado, presenta una mejor dispersión de datos, logrando así, una probabilidad estadística más representativa para cada caso de estudio.

La implementación de un código elaborado en Python para la obtención de curvas de fragilidad es una herramienta significativa para el desarrollo y clasificación por demanda sísmica a una determinada curva de capacidad obtenida por su respectiva tipología estructural, por lo tanto, ambos códigos están abiertos a la manipulación, corrección y su respectivo mejoramiento.

Palabras Claves: Riesgo sísmico, punto de desempeño, curva de capacidad, representación bilineal, curvas de fragilidad, umbrales de daño.

ABSTRACT

In this thesis project, the development of fragility curves was carried out using two Python codes based on two methodologies: one proposed by ASCE 41-17, known as the Coefficient Method, and the other proposed by FEMA 440, Alternative C, also known as the "Capacity Spectrum Method." Both methods are used to obtain performance points for a specific structure that is to be analyzed.

During the development of the project, it was identified that there is variation in the section required for the generation of fragility curves. The Coefficient Method shows a linear increasing trend, whereas the method proposed by FEMA 440 in the same section presents better data dispersion, thus achieving a more representative statistical probability for each case study.

The implementation of a Python code for the generation of fragility curves is a significant tool for the development and classification of seismic demand for a specific capacity curve obtained according to its respective structural typology. Therefore, both codes are open to manipulation, correction, and further improvement.

Keywords: Seismic risk, performance point, capacity curve, bilinear representation, fragility curves, damage thresholds.

INTRODUCCIÓN

i. ANTECEDENTES

Como se conoce, Ecuador es un país con un alto riesgo sísmico, debido a su ubicación geográfica, así lo menciona Parra (2016):

Ecuador se encuentra localizado en el cinturón de fuego del océano Pacífico, al noroccidente de Sudamérica. siendo uno de los países más densamente poblados de la región que muestra actividad sísmica causada principalmente por la subducción de la placa oceánica de nazca y la presencia de un complejo sistema de fallas activas locales.

Debido a su peligro sísmico, Ecuador ha tenido una gran cantidad de sismos importantes, Del Salto Díaz (2023), nos relata que:

El perfil costero del Ecuador históricamente ha sufrido varios sismos de gran magnitud (en los años 1906, 1942, 1958, 1998 y 2016), los mismos que han causado centenares de muertos y heridos, además de graves daños estructurales a viviendas y edificios.

Uno de los últimos sismos de gran magnitud ocurridos en Ecuador fue el del año 2016, Del Salto Días (2023), destaca lo que fue este sismo:

El 16 de abril de 2016, se suscitó un sismo de 7.8 grados en la escala de Richter, el cual provocó estragos en todo el litoral ecuatoriano, dejando un total de 673 muertos. La población más afectada fue la que se encontraba ubicada en la localidad de Pedernales, en donde destruyó el 80% de las edificaciones.

De acuerdo con lo mencionado, Ecuador es un país con alto peligro sísmico, hoy en día, es importante conocer cómo afecta y cuál será el comportamiento de la estructura frente a un sismo de cierta magnitud, para ello, existen varias metodologías que estudian estos eventos y comportamientos, uno de ellos son las curvas de fragilidad, basado en las probabilidades que tiene una estructura a sufrir daños frente a un terremoto.

En el sismo de Pedernales del 16 de abril del 2016 en Ecuador, se tomaron los registros sísmicos de dicho evento, estos son importantes y nos permiten conocer que aceleraciones puede alcanzar un sismo de tal magnitud y cómo influye de manera negativa en las estructuras. Estos registros se los puede obtener del Instituto Geofísico de la escuela Politécnica Nacional.

Y así, Rojas (2010), de manera introductoria nos menciona la importancia de estas curvas de fragilidad, “dentro de la carrera de Ingeniería Civil, en el área de la Ingeniería Estructural y Sismorresistente, un tópico de estudio se centra en la vulnerabilidad sísmica de edificaciones y estimar daños debidos a terremotos utilizando las curvas de fragilidad”, en esa misma línea, Rojas (2010), nos explica en qué consisten estas curvas: “Estas curvas permiten representar la vulnerabilidad sísmica de las edificaciones y calcular la probabilidad de que la respuesta de una estructura alcance o exceda determinado límite, asociado al daño, como función de un parámetro indicador de la intensidad sísmica.”

Es decir, las curvas de fragilidad son una herramienta basada en probabilidad estadística, que permite conocer cuál será el daño probable que tendrá una estructura frente a un evento sísmico, y así, conocer y tomar medidas para la prevención y mitigación de daños, por eso, es importante el análisis y estudio de la vulnerabilidad de las estructuras, frente a la amenaza sísmica, y el riesgo que podría traer con ello.

En la actualidad existen programas que permiten desarrollar curvas de fragilidad, para, evaluar el daño sísmico en estructuras. Algunos de estos son:

- FRACAS desarrollado por University College London (2017)
- P-FAS desarrollado por University Of Illinois at Urbana-Champaign (2005).

La finalidad de estos programas es realizar las curvas de fragilidad en función de “pga vs probabilidad de excedencia”.

Si nos referimos al pga y la probabilidad de excedencia, Soria, (2010) nos menciona lo siguiente: “El pga refleja muy poca información tanto en la acción sísmica como en el daño en las estructuras y en varias ocasiones se

han observado daños pocos relevantes en estructuras sometidas a aceleraciones del suelo muy altas”, por otro lado, en un estudio realizado por la Universidad Politécnica de Cataluña Moreno González & Bairán García, (2010) en su caso de estudio, nos comentan que: “La aceleración espectral correspondiente a los puntos de capacidad por demanda aumenta, y los desplazamientos espectrales decrecen al disminuir el número de niveles de los edificios.”

Dentro de dicho estudio, presentan sus curvas de fragilidad en función del desplazamiento espectral (s_d) y la probabilidad de excedencia.

En este trabajo se utilizará el Método de Espectro de Capacidad (CSM). Este método fue propuesto por Freeman, es un método rápido para evaluar el riesgo sísmico. Existen varias formas de desarrollar el CSM, ya sea mediante:

- ATC (Seismic Evaluation and Retrofit of Concrete Buildings)
- FEMA (IMPROVEMENT OF NONLINEAR STATIC SEISMIC ANALYSIS PROCEDURES)
- ASCE 41-17

Como se menciona, existen diferentes formas de desarrollar el método del espectro de capacidad, Lizarazo (2022), nos describe de manera breve que consiste:

El método constituye un procedimiento simple para determinar el punto de desempeño de una estructura cuando se ve sometida a movimientos sísmicos de diferente intensidad. Mediante un procedimiento gráfico, se compara la capacidad para resistir fuerzas laterales con la demanda sísmica, representada por medio de un espectro de demanda reducido.

Si bien, ya existen programas que realizan curvas de fragilidad, como los mencionados anteriormente, estos usan como parámetro el “ p_g ” y en el presente trabajo se usará el desplazamiento espectral “ s_d ”. Para ello, se busca implementar el método del espectro de capacidad “CSM” haciendo uso de la programación utilizando Python para el desarrollo del código, con el fin de automatizar el análisis para diferentes tipos de estructuras y tomando

diferentes registros sísmicos, es decir, el código deberá leer registros sísmicos previamente seleccionados y la curva de capacidad de la estructura que se quiera estudiar, con esto, el software será capaz de procesar los datos y obtener puntos de desempeños, obteniendo así, los valores necesarios para elaborar las curvas de fragilidad.

ii. OBJETIVO GENERAL

- Elaborar un código en Python para la generación de curvas de fragilidad mediante el método del espectro de capacidad (CSM).

iii. OBJETIVOS ESPECÍFICOS

- Implementar el método del espectro de capacidad.
- Determinar la curva bilineal de la estructura de estudio.
- Generar puntos de desempeño utilizando diversos registros sísmicos.
- Generar curvas de fragilidad del estudio previo realizado por Toro y Flores.
- Evaluar y validar los resultados obtenidos mediante el código Python desarrollado en este estudio.
- Comparar las metodologías FEMA 440 y ASCE 41-17.

iv. ALCANCE

El alcance de este estudio implica el desarrollo de un código en Python, dirigido a la elaboración de curvas de fragilidad utilizando el Método del Espectro de Capacidad (CSM). Este código facilitará la determinación de los puntos de desempeño de las estructuras en función de sus características, es decir a partir de su respectiva curva de capacidad y la demanda sísmica asociada. Además, se integrarán funcionalidades para el cálculo de las curvas de fragilidad a partir de los datos obtenidos, así como la generación de gráficos para una mejor comprensión visual de los resultados. Se llevará a cabo una evaluación comparativa de los resultados obtenidos con investigaciones previas. Asimismo, se considerarán las limitaciones con método CSM, las simplificaciones realizadas y la disponibilidad de datos para garantizar una interpretación precisa de los resultados y su aplicación futura

en investigaciones relacionadas con la ingeniería estructural y la evaluación del riesgo sísmico.

CAPÍTULO I: MARCO TEÓRICO

1.1 AMENAZA SÍSMICA

Herrera (2000), define la amenaza como:

Se llama amenaza, peligro, riesgo a algún estado de naturaleza o de la sociedad que sea de una fuente de peligro con potencial para causar enfermedad, herida o muerte de personas o daño a alguna instalación al ambiente, en el contexto de amenaza sísmica, trata de los peligros y amenaza naturales que conllevan los sismos, afectando edificios y otro tipo de construcciones.

Además, Herrera (2000), relaciona el concepto de amenaza con la sismología y menciona que: “la amenaza sísmica es un término usado para describir la caracterización general del movimiento sísmico, cuantificando la ocurrencia de futuros acontecimientos sísmicos y los efectos asociados.”

Udías y Mézcua, muestran un concepto sencillo de peligrosidad sísmica: “Se entiende por peligrosidad sísmica de una zona, cualquier descripción de los efectos provocados por terremotos en el suelo de dicha zona.” (como se citó en Yépez, Barbat & Canas, 1995).

También, Yépez, Barbat & Canas, (1995), definen la peligrosidad sísmica como: “Representa la probabilidad de ocurrencia, dentro de un período específico de tiempo y dentro de un área dada, de un movimiento sísmico del terreno de una intensidad determinada.”

Yépez, Barbat & Canas, (1995), adicionan y comentan que:

Los efectos pueden venir representados mediante la aceleración, velocidad, desplazamiento, o por la intensidad sentida en el lugar y para evaluar es necesario analizar los fenómenos que ocurren desde la emisión sísmica en el foco hasta que dichas ondas alcanzan el lugar en cuestión.

Es decir, la amenaza sísmica es la probabilidad de ocurrencia de un evento sísmico, en una región específica y en un periodo determinado, además, cómo la intensidad del sismo puede afectar determinada área.

Una forma de entender la amenaza sísmica, es que, depende únicamente de la naturaleza, es decir, no puede ser controlada, ni prevista, ocurre como un evento natural, independiente de lo construido o realizado por acciones humanas. Es importante destacar este concepto, puesto que no se debe confundir, amenaza sísmica, vulnerabilidad sísmica y riesgo sísmico.

La amenaza sísmica es un parámetro fundamental, para desarrollar evaluaciones de riesgo sísmico.

1.2 VULNERABILIDAD SÍSMICA

Un concepto de vulnerabilidad sísmica es proporcionado por Yépez, Barbat & Canas, (1995), y mencionan que: “La vulnerabilidad sísmica de una estructura, grupo de estructuras, es definida, como el grado de daño debido a la ocurrencia de un movimiento sísmico del terreno de una intensidad determinada.”

Para Fournier, citado por Caicedo, Barbat, Canas & Aguiar (1994), en el tópico de fenómeno sísmico, define la vulnerabilidad de una estructura o grupo de estructuras, como: “El grave daño que resulta por la ocurrencia de un movimiento sísmico del terreno de una intensidad dada”.

Además, Caicedo, Barbat, Canas & Aguiar (1994), adicionan que: “La vulnerabilidad es una característica exclusiva de las estructuras, dependiente de la forma como hayan sido diseñadas, pero independiente de la peligrosidad sísmica del sitio donde estén ubicadas.”

Rojas, (2010), mantiene las mismas ideas que otros autores y define la vulnerabilidad sísmica como:

La vulnerabilidad sísmica es el nivel de predisposición que tiene una estructura a sufrir daños ante un evento sísmico, es decir, el límite en el que se sobrepasa el grado de reserva o el nivel de capacidad de respuesta disponible ante una amenaza sísmica conocida.

Los factores que influyen en la variación de la vulnerabilidad sísmica son:

- Factores geológicos.
- Factores estructurales.
- Factores arquitectónicos.
- Factores constructivos.
- Factores socioeconómicos

A partir de experiencia de terremotos pasados se ha observado que ciertas estructuras, dentro de la misma tipología estructural, experimentan un daño más severo que otras, a pesar de estar ubicadas en alguna zona. Si el grado de daño que sufre una estructura, ocasionado por un sismo de determinadas características, se le denomina vulnerabilidad, se puede entonces clasificar los edificios en “más vulnerables” o “menos vulnerables” frente a un mismo evento sísmico. (Yépez, Barbat & Canas, 1995).

Cada autor o autores tienen sus propios conceptos acerca de la vulnerabilidad sísmica, y coinciden que este parámetro a diferencia de la amenaza sísmica, está relacionado directamente con las estructuras, en su comportamiento y respuesta frente a las acciones producidas por un sismo, es decir, el daño que puede alcanzar la estructura cuando ocurre un terremoto con determinada magnitud. Este parámetro, así mismo, depende del diseño y construcción de la estructura.

En la actualidad estamos rodeados por un entorno con diversas construcciones, ya sea, casas, edificios, puentes, entre otras, donde su relación es que son construidas para el uso diario de seres humanos, por lo tanto, conocer la vulnerabilidad de una estructura es muy importante cuando nos referimos a seguridad de vida, pues, se puede alcanzar a conocer, prevenir, e incluso, desarrollar planes de riesgo, frente a acciones sísmicas.

Por tal motivo, Caicedo, Barbat, Canas & Aguiar, (1994) nos comenta porque son importantes la realización de estudios de vulnerabilidad:

Es una herramienta clave para los planes de mitigación de desastres por ejemplo una entidad del Gobierno puede estar interesada en conocer la condición actual de los edificios de una ciudad con el objetivo de prever anticipadamente las consecuencias negativas económicas y sociales que la ocurrencia de un sismo de una determinada magnitud puede generar en la zona y de esta forma determinar las medidas necesarias para su mitigación.

1.3 RIESGO SÍSMICO

Una vez aclarados los conceptos de Peligrosidad Sísmica y Vulnerabilidad Sísmica, Mena (2002) nos menciona lo siguiente acerca del riesgo sísmico:

Se puede observar que existe una relación directa entre ellos, es decir, para que exista verdaderamente riesgo sísmico en un lugar, ambos conceptos deben producirse y existir respectivamente, es decir, el riesgo sísmico evalúa y cuantifica las consecuencias sociales y económicas potenciales provocadas por un terremoto.

Según la UNDRRO, define al riesgo sísmico como: “El grado de pérdidas que se esperan como producto de la ocurrencia de un evento particular y como una función de la amenaza y la vulnerabilidad” (como se citó en Chávez, 2016)

Por otro lado, López y Villacañas comentan que: “El riesgo sísmico se define como la función de las probabilidades de pérdidas derivadas de los daños a un elemento o un conjunto de elementos como consecuencia de la acción de terremotos.” (como se citó en Parra, 2016)

Según Bommer, Salazar, & Samayoa: “Dicha pérdida puede ser medida en términos humanos, económicos o sociales y la escala que se escoja define el costo de la pérdida que da una medida del riesgo.” (como se citó en Chávez, 2016)

Covenin, menciona que: “El riesgo sísmico es el grado de daño o pérdidas esperado, ante la ocurrencia de un evento sísmico”, y Alonso extiende el concepto y define como: “la probabilidad de que, en un determinado sitio y durante la acción de alguna amenaza natural se produzcan

pérdidas de vidas, económicas y sociales que excedan ciertos niveles de daño preestablecidos.” (como se citó en Rojas, 2010).

Rojas (2010) acerca de la evaluación del riesgo sísmico en edificios o en una obra civil, menciona que:

La evaluar el riesgo sísmico se puede realizar en función de la amenaza sísmica, su vulnerabilidad estructural y del costo o importancia de esta. Al reducir alguna de las variables antes mencionadas, reduce el riesgo sísmico de la edificación, donde la reducción de la vulnerabilidad estructural es en la que se enfocan los ingenieros estructurales y arquitectos para lograr este objetivo.

Evaluando los conceptos, no hay que confundir peligro o amenaza sísmica con riesgo sísmico, puesto que, el riesgo sísmico está en función del peligro o amenaza sísmica y la vulnerabilidad sísmica.

1.4 MÉTODO DE NEWMARK

En 1959, N. M. Newmark desarrolló una familia de métodos de integración numérica conocidos como el método de Newmark, utilizados para resolver ecuaciones diferenciales que describen el comportamiento dinámico de los sistemas estructurales bajo cargas dinámicas.

Chopra (2014), en su libro “Dynamics of Structures: Theory and Applications to Earthquake Engineering”, describe el método de Newmark:

El método de Newmark es un procedimiento para la evaluación numérica de la respuesta dinámica de sistemas que permite calcular las respuestas de desplazamiento y velocidad en el tiempo utilizando fórmulas específicas que consideran la aceleración constante o la aceleración lineal dentro de cada intervalo de tiempo.

Este método se basa en las siguientes ecuaciones:

$$\dot{u}_{i+1} = \dot{u}_i + [(1 - \gamma)\Delta t]\ddot{u}_i + (\gamma\Delta t)\ddot{u}_{i+1}$$

$$u_{i+1} = u_i + (\Delta t)\dot{u}_i + \left[\left(\frac{1}{2} - \beta \right) (\Delta t)^2 \ddot{u}_i + \beta (\Delta t)^2 \ddot{u}_{i+1} \right]$$

Los parámetros " γ " y " β " definen la variación de la aceleración durante un paso de tiempo y determinan las características de estabilidad y precisión del método. Una selección típica para " γ " es $\frac{1}{2}$, y $\frac{1}{6} \leq \beta \leq \frac{1}{4}$ es satisfactoria desde todos los puntos de vista, incluida la precisión.

En el libro "Dynamics of Structures: Theory and Applications to Earthquake Engineering" de Anil K. Chopra, se puede encontrar el desarrollo detallado y las ecuaciones del método de Newmark, en las secciones de métodos de integración numérica.

1.5 MÉTODO DEL ESPECTRO DE CAPACIDAD (CSM)

El método del espectro de capacidad es un método basado en análisis estático no lineal, según Chiroiu, Goula, Roca, Susagna, Barbat y Pujades, (2005):

Constituye una herramienta nueva que se puede utilizar para el análisis de daños estructurales y no estructurales debidos a una acción sísmica tanto para la generación de escenarios de daños como para la rehabilitación de edificios. Basándose en parámetros espectrales como desplazamiento o aceleración.

Es un método gráfico, que depende de la capacidad de la estructura y la demanda sísmica, así lo explica Lizarazo (2022).

Mediante un procedimiento gráfico, se compara la capacidad para resistir fuerzas laterales con la demanda sísmica, representada por medio de un espectro de demanda reducido. La representación gráfica hace posible una evaluación visual de cómo podría comportarse la estructura cuando se somete a un determinado movimiento sísmico.

Lizarazo (2022), menciona también que: “La capacidad de la estructura se visualiza mediante una curva que muestra la relación entre la fuerza lateral aplicada, o cortante basal V , y el desplazamiento en la parte superior δ ”.

Esta curva puede ser obtenida mediante un análisis "pushover". Como se mencionó con anterioridad, las curvas de fragilidad relacionan la probabilidad de excedencia y un parámetro sísmico, en este método, se presenta la curva de capacidad, que es la capacidad de la estructura, obtenida con el análisis estático no lineal, transformada a coordenadas espectrales.

La demanda sísmica se representa por medio de un espectro demanda reducido en formato AD (S_a vs S_d) que considera la respuesta no lineal de la estructura. El espectro de demanda reducido se obtiene a partir de la reducción del espectro elástico lineal, por medio de un amortiguamiento histórico equivalente (β_{eq}). (Lizarazo, 2022).

Para determinar el punto de desempeño de la estructura se superponen los espectros de demanda y capacidad sísmica. Este punto debe cumplir con las siguientes condiciones:

- 1) Debe estar sobre el espectro de capacidad para representar a la estructura en un determinado desplazamiento
- 2) Debe estar sobre el espectro de demanda (reducido a partir del espectro elástico) que representa la demanda no lineal en el mismo desplazamiento estructural.

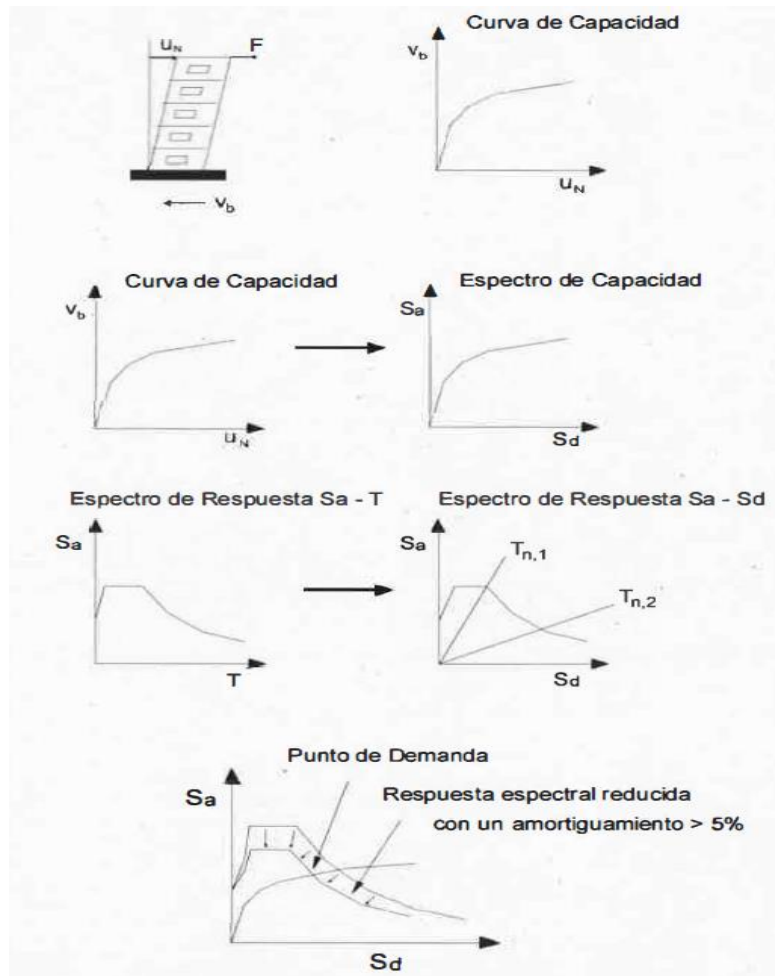


Figura 1. Diagrama de metodología del espectro de capacidad. (Fuente: Chiroiu, Goula, Roca, Susagna, Barbat y Pujades, 2005)

1.5.1 ESPECTRO DE ACELERACIÓN VS DESPLAZAMIENTO FORMATO ADRS

Cuando se menciona el espectro de aceleración vs desplazamiento, hace referencia a la demanda sísmica, la demanda sísmica es una representación gráfica que muestra cómo diferentes estructuras respondan a las excitaciones producidas por efectos del sismo. Este espectro describe la relación entre la aceleración máxima del suelo y el período natural de vibración de una estructura.

Al analizar el espectro de demanda sísmica, se pueden determinar las fuerzas sísmicas que actuará sobre una estructura y predecir su comportamiento durante un terremoto, sin embargo, debe ser transformado a

un formato ADRS, es decir, convertir a coordenadas espectrales (S_a vs S_d) y así, obtener la gráfica con la que se superpondrá la curva de capacidad para obtener el punto de desempeño.

En el presente trabajo, la demanda sísmica está dado por todos los registros sísmicos que se obtuvieron en el sismo de Pedernales el 16 de abril de 2016, sin embargo, estos valores están dados en aceleraciones registrados en un periodo de tiempo, por lo que hay que transfórmalos a coordenadas espectrales (S_a vs S_d), para ello utilizamos el método de Newmark, así los registros sísmicos pasan a ser espectros de respuesta de aceleración vs periodo.

Según ATC-40 (Applied Technology Council, 1996), para convertir el espectro seleccionado a un espectro de respuesta S_a vs S_d , el periodo es modificado según la siguiente ecuación:

$$S_d = \frac{T^2}{4\pi^2} \cdot S_a$$

1.5.2 CURVA DE CAPACIDAD Y ESPECTRO DE CAPACIDAD

La curva de capacidad se obtiene a partir de un análisis estático no lineal, conocido como “pushover”.

Dunn (2016), detalla en que consiste la curva de capacidad:

Consiste en aplicar fuerzas laterales incrementales a la estructura hasta que este alcance el colapso. En este método, a medida que incrementan las fuerzas laterales, se producen rotulas plásticas en los elementos, cambiando la rigidez de la estructura y permitiendo una redistribución de la carga.

Además, Dunn (2016) explica que: “Este proceso continúa hasta que la generación de rotulas plásticas produzca una inestabilidad en la estructura, la cual es considerada como colapso.”.

Este tipo de análisis, según menciona Dunn, (2016):

Es una técnica de análisis utilizada en el diseño por desempeño, la cual utilizando un patrón de cargas lateral, que se incrementa en cada iteración, lleva la estructura más allá del rango elástico y, posteriormente, al colapso o hasta una deformación máxima, y así, se obtiene una gráfica de fuerza desplazamiento conocida como curva de capacidad o pushover.

Así como se realizó con la demanda sísmica, se debe transformar a coordenadas espectrales, para formar el espectro de capacidad, donde las cargas laterales pasen a ser aceleración espectral (S_a) y las derivas o desplazamientos pasen a ser desplazamientos espectrales (S_d), este formato se lo conoce como ADRS.

El ATC 40 y el FEMA 440, presentan las siguientes ecuaciones para la conversión a un formato ADRS:

$$s_{ai} = \frac{V_i}{\alpha_1}$$
$$s_{di} = \frac{\delta_i}{PF_1 \cdot \phi_{1,roof}}$$

Donde:

- α_1 : *Coficiente de masa modal*
- PF_1 : *Factor de participación para el primer modo de la estructura*
- $\phi_{1,roof}$: *Nivel de amplitud del techo en el primer modo.*
- V_i : *Punto del cortante basal de la curva de capacidad.*
- W : *Peso sísmico.*
- δ_i : *Punto del desplazamiento en la curva de capacidad.*

Nota: Es importante destacar que en el presente trabajo no se realiza el análisis pushover, este se ha calculado en estudios previos y se utilizarán esos resultados.

La curva que se utilizará pertenece al estudio realizado por Flores y Toro (2024), que fue analizada en su trabajo de titulación “Cálculo de curvas de fragilidad de estructuras de hormigón armado de un piso alto con vigas

banda y cubierta liviana en residencias ubicadas en vía a la Costa, Guayaquil – Ecuador”.

1.5.3 CURVA BILINEAL

La curva de capacidad está formada por una serie de puntos, para el método del espectro de capacidad, se realiza una simplificación, creando así una representación bilineal de la curva de capacidad.

Así lo describe Moreno & Bairán, (2014):

El espectro de capacidad se representa por medio de una curva de capacidad bilineal simplificada, definida por dos puntos, el punto de capacidad de cedencia (D_y, A_y) y el punto de capacidad última (D_u, A_u), donde las áreas sobre y bajo la curva de capacidad original deben ser iguales.

Moreno & Bairán (2014), también menciona que:

El punto de capacidad de cedencia representa el desplazamiento en el que la respuesta del edificio empieza a ser no lineal y el punto de capacidad última representa el desplazamiento en el que el sistema estructural ha alcanzado el mecanismo de colapso.

Esta representación bilineal es importante para definir los umbrales de los estados de daño y, construir las curvas de fragilidad.

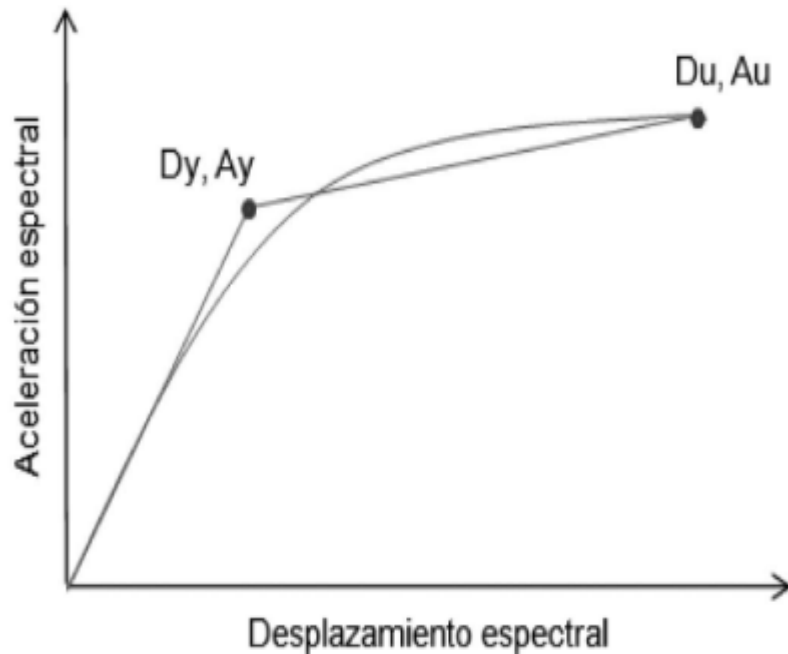


Figura 2. Representación de la Curva Bilineal. (Fuente: Moreno & Bairán, 2014)

1.5.4 PUNTO DE DESEMPEÑO

Conocido también como desplazamiento objetivo, es el punto de interés donde intersecan la curva de capacidad o su representación bilineal, con el espectro de demanda.

Moreno (2006), comenta acerca del punto de desempeño y dice que:

Superponiendo el espectro capacidad y el de demanda se obtiene la respuesta máxima del edificio en el punto de intersección de ambos espectros, este punto es conocido como punto de desempeño o bien en la literatura inglesa como “performance point”, donde el punto de desempeño representa el máximo desplazamiento estructural esperado por el terremoto de demanda.

Además, Moreno, Bairán & Rodas (2011), detallan que: “El comportamiento sísmico de un edificio sometido a un terremoto se puede cuantificar mediante su respuesta máxima, que se define mediante el punto

de capacidad por demanda o conocido también como punto de desempeño (PD).”

Calderón (2018), menciona que el punto de desempeño “define el desempeño sísmico de un edificio ante una acción sísmica definida y representa el máximo desplazamiento estructural esperado para el sismo de demanda”.

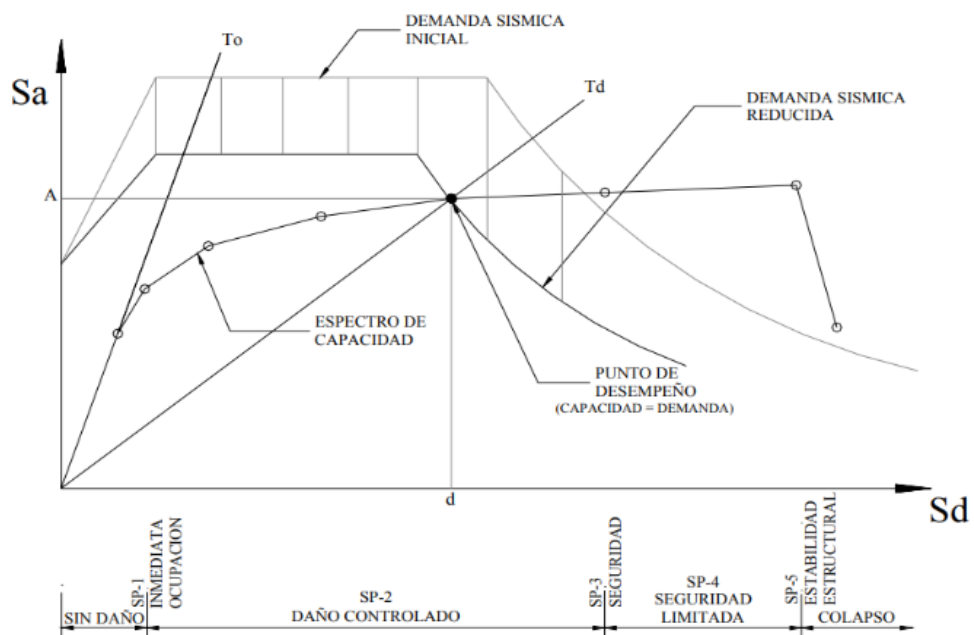


Figura 3. Intersección del espectro de capacidad y espectro de demanda conocido como “Punto de desempeño”. (ATC – 40, 1996)

1.6 MÉTODOLÓGÍAS DEL ESPECTRO DE CAPACIDAD

1.6.1 FEMA 440

La Agencia Federal para el Manejo de Emergencias por sus siglas en inglés (FEMA), en su texto “Improvement of Nonlinear Static Seismic Analysis Procedures” publicado en el año 2005, describe en qué consiste el método del espectro de capacidad:

El objetivo del Método del Espectro de Capacidad es desarrollar espectros de demanda y capacidad apropiados para la estructura y determinar su punto de intersección. Durante este proceso, también se

evalúa el desempeño de cada componente estructural. La curva de demanda se basa en el espectro de respuesta del terremoto y las curvas de capacidad se basan en el análisis pushover no lineal estático.

Además, el FEMA 440 (2005), define y explica las características importantes del método, para encontrar los desplazamientos objetivos:

La curva de capacidad se obtiene a partir de un análisis pushover, donde, la estructura es sometida a niveles crecientes de carga, y se grafica el cortante basal versus el desplazamiento del techo de la estructura a lo largo del proceso. El espectro de capacidad se obtiene transformando el espectro de cortante basal versus desplazamiento del techo en un espectro de aceleración espectral versus desplazamiento espectral.

FEMA 440 (2005), también menciona que:

La intersección de una curva de demanda apropiada con la curva de capacidad se llama punto de desempeño. El punto de desempeño define el cortante basal y el desplazamiento estimado de la estructura cuando está sujeta al terremoto representado por la curva de demanda. El comportamiento de la estructura en el punto de desempeño se compara con criterios de aceptación predefinidos para determinar si se cumplen los objetivos de diseño.

De manera resumida, Dunn (2016), explica que “el objetivo de este método es el de estimar el desplazamiento máximo de un sistema no lineal mediante un sistema lineal equivalente usando el periodo efectivo, T_{eff} , y el amortiguamiento efectivo β_{eff} .”

El método FEMA, cuenta con 3 formas de desarrollar el método del espectro de capacidad, que se explicarán en las secciones 1.6.1.1, 1.6.1.2 y 1.6.1.3.

1.6.1.1. MÉTODO “A”

Dunn (2016), en su tesis “Rehabilitación estructural de la facultad de Ingeniería mediante aisladores sísmicos”, describe el proceso numérico y teórico que se realiza para la aplicación del método:

1. Se elige un espectro de demanda de referencia con un amortiguamiento del 5%. A continuación, se introduce el efecto de la interacción suelo-estructura, modificando el espectro a través de reducciones en las ordenadas espectrales por interacción cinemática y ajustando el amortiguamiento desde el valor inicial (β_i) hasta un valor final (β_0) que considera el amortiguamiento de la cimentación. Si se desprecia el amortiguamiento de la cimentación, β_0 será igual a β_i .
2. El espectro modificado se convierte al formato aceleración-desplazamiento (ADRS) según lo indica el ATC-40, lo que da como resultado el espectro de demanda ADRS inicial. Este procedimiento está descrito en la sección 1.5.1.

$$S_d = \frac{T^2}{4\pi^2} \cdot S_a$$

3. Luego, se genera la curva de capacidad en formato ADRS, de igual manera, se sigue las condiciones del ATC-40. este proceso se menciona en la sección 1.5.2.

$$s_{ai} = \frac{V_i}{\alpha_1}$$
$$s_{di} = \frac{\delta_i}{PF_1 \cdot \phi_{1,roof}}$$

4. Se debe definir un punto de desempeño inicial en función a la aceleración máxima a_{pi} y desplazamiento máximo d_{pi} .
5. Se procede a crear la curva bilineal explicada en la sección 1.5.3., con los puntos de fluencia (d_y, a_y), y se define el periodo inicial T_0 .

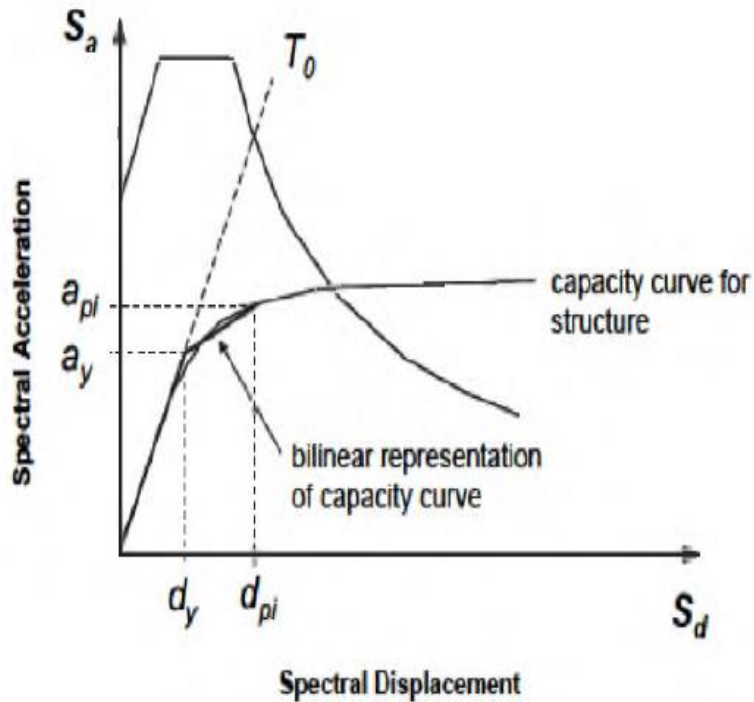


Figura 4. Periodo inicial, Curva bilineal, Puntos de interés. (Fuente: FEMA 440, 2005)

6. En base al paso 5, se calcula valores de rigidez (post - elástica α) y ductilidad (μ), y con estos valores se calcula el amortiguamiento efectivo (β_{eff}) y período efectivo (T_{eff}). Se utilizan diferentes fórmulas propuestas por el FEMA 440, según la ductilidad obtenida:

$$\alpha = \frac{\left(\frac{a_{pi} - a_y}{d_{pi} - d_y}\right)}{\frac{a_y}{d_y}}$$

$$\mu = \frac{d_{pi}}{d_y}$$

Para ductilidad $1.0 < \mu \leq 4.0$:

$$\beta_{eff} = 4.9(\mu - 1)^2 - 1.1(\mu - 1)^3 + \beta_0$$

$$T_{eff} = T_0[0.2(\mu - 1)^2 - 0.038(\mu - 1)^3 + 1]$$

Para ductilidad $4.0 < \mu \leq 6.5$:

$$\beta_{eff} = 14.0 + 0.32(\mu - 1) + \beta_0$$

$$T_{eff} = T_o[0.28 + 0.13(\mu - 1)]$$

Para ductilidad $\mu > 6.5$:

$$\beta_{eff} = 19 \left\{ \frac{0.64(\mu - 1) - 1}{[0.64(\mu - 1)]^2} \right\} \left(\frac{T_{eff}}{T_o} \right)^2 + \beta_0$$

$$T_{eff} = T_o \left\{ 0.89 \left[\sqrt{\frac{(\mu - 1)}{1 + 0.05(\mu - 2)}} - 1 \right] + 1 \right\}$$

7. Se ajusta los espectros en formato ADRS iniciales utilizando el β_{eff} .

$$B = \frac{4}{5.6 - \ln \beta_{eff}} \rightarrow \beta_{eff} (\%)$$

$$(S_a)_\beta = \frac{(S_a)_0}{B(\beta_{eff})}$$

$$(S_d)_\beta = \frac{T^2}{4\pi^2} (S_a)_\beta$$

8. Con el periodo efectivo y el espectro en formato ADRS calculado con el amortiguamiento efectivo, se determina el desplazamiento máximo esperado.

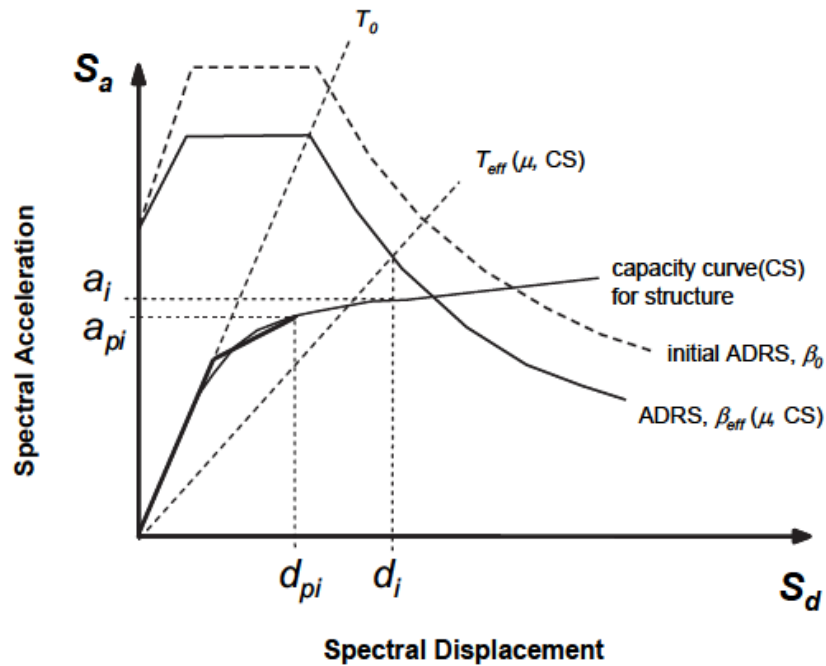


Figura 5. Máximo desplazamiento esperado utilizando el Método A. (Fuente: FEMA 440, 2005)

1.6.1.2. MÉTODO “B”

A diferencia del método “A”, este método, define el punto de desempeño o desplazamiento objetivo con un espectro de demanda en formato MADRS, donde la letra “M”, hace referencia a “modificado”.

Se deben seguir los pasos del 1 al 7 del método A, explicado en la sección 1.6.1.1.

Dunn (2016), al igual que en el método A, explica el método B a continuación:

1. Utilizando el β_{eff} , se debe calcular el factor de modificación “M”, este se debe multiplicar por los valores de aceleración, y así generar el espectro de respuesta modificado “aceleración vs desplazamiento” (MADRS).

$$M = \left(\frac{T_{eff}}{T_{sec}}\right)^2 = \left(\frac{T_{eff}}{T_0}\right)^2 \left(\frac{T_0}{T_{sec}}\right)^2$$

$$\left(\frac{T_o}{T_{sec}}\right)^2 = \frac{1 + \alpha(\mu - 1)}{\mu}$$

- Finalmente, se determina el desplazamiento objetivo con la intersección de la curva de capacidad y el espectro de respuesta modificado.

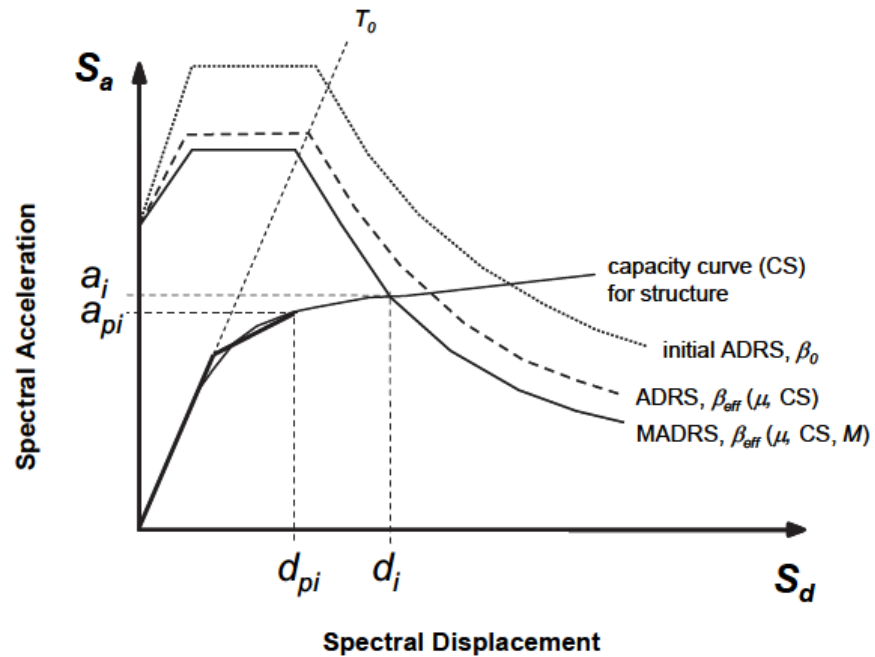


Figura 6. Máximo desplazamiento esperado utilizando el Método B. (Fuente: FEMA 440, 2005)

1.6.1.3. MÉTODO “C”

Respecto al método “C”, Dunn (2016), menciona que en este método: “Se usa el espectro de respuesta modificado para múltiples soluciones asumidas (a_{pi} y d_{pi}), y las correspondientes ductilidades para generar una línea, uniendo los múltiples puntos de desempeño.”

Este punto de desempeño se obtiene con la intersección de la línea de los puntos de desempeño posibles y con la curva de espectro de capacidad.

Para desarrollar el método C, se utiliza la misma metodología que el método “A”, y el método “B”.

Por lo tanto, se deben seguir los pasos del 1 al 7 del método “A”, y adicionar el punto 1, del método “B”, a continuación, Dunn (2016), menciona los pasos siguientes para usar la metodología “C”:

1. Se encuentra un posible punto de desempeño que es generado por la intersección del periodo secante radial T_{sec} , con el espectro de respuesta modificado (MADRS). Véase figura 7.
2. Aumentar o disminuir el punto de desempeño asumido y repetir el proceso para generar una serie de posibles puntos de desempeños.
3. El punto de desempeño actual es definido por la intersección de la unión de puntos del paso con la curva de espectro de capacidad.

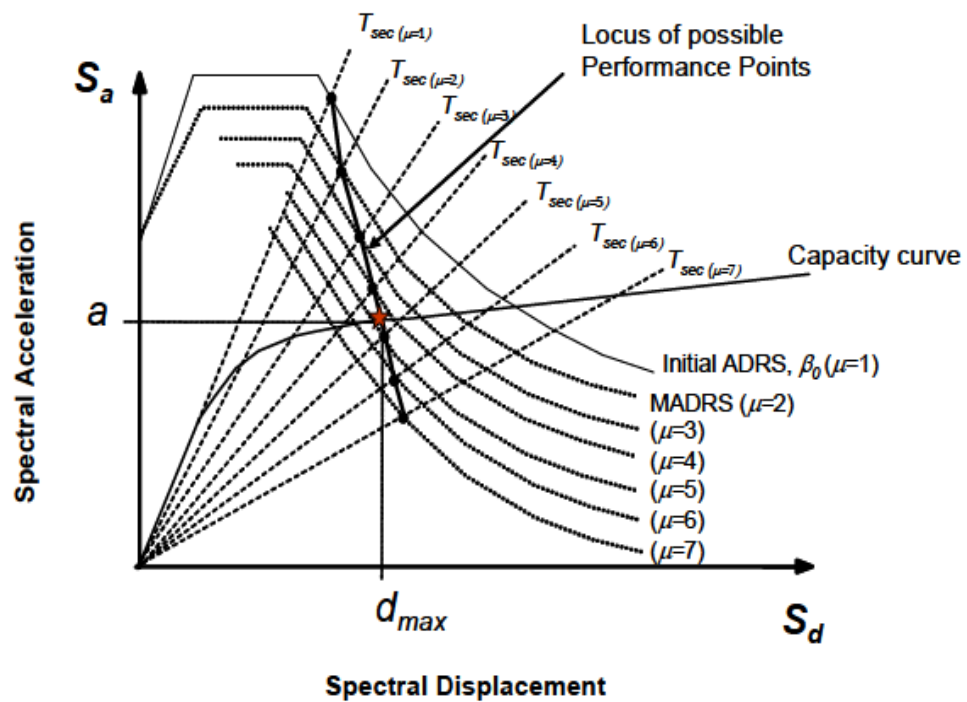


Figura 7. Máximo desplazamiento esperado utilizando el Método C. (Fuente: FEMA 440, 2005).

1.6.2 ASCE 41-17

El método del ASCE 41-17, conocido como el método de los coeficientes, es utilizado para calcular el desplazamiento objetivo. Simplifica el método de FEMA 440, puesto que utiliza coeficientes, sin embargo, utiliza la misma filosofía.

Dunn, (2016) menciona que “este método busca estimar el punto de desempeño que tendrá la estructura ante una demanda sísmica, a partir de la curva de capacidad de esta”.

El coeficiente C_o tiene en cuenta la diferencia entre el desplazamiento del techo de un edificio de múltiples grados de libertad (MDOF) y el desplazamiento del sistema equivalente de un solo grado de libertad (SDOF). Usando solo la forma modal del primer modo (ϕ_1) y el comportamiento elástico. (ASCE 41-17)

Dunn (2016), al igual como lo hizo con la metodología FEMA 440, presenta una serie de pasos para la aplicación del método de los coeficientes del ASCE 41-17:

1. Se genera la curva de capacidad de la estructura y se calcula su curva bilineal, estos conceptos se detallan en la sección 1.5.2 y 1.5.3, respectivamente. A partir de estas curvas se define la rigidez lateral efectiva (k_e), el cortante de fluencia (V_y) y el desplazamiento de fluencia (D_y).

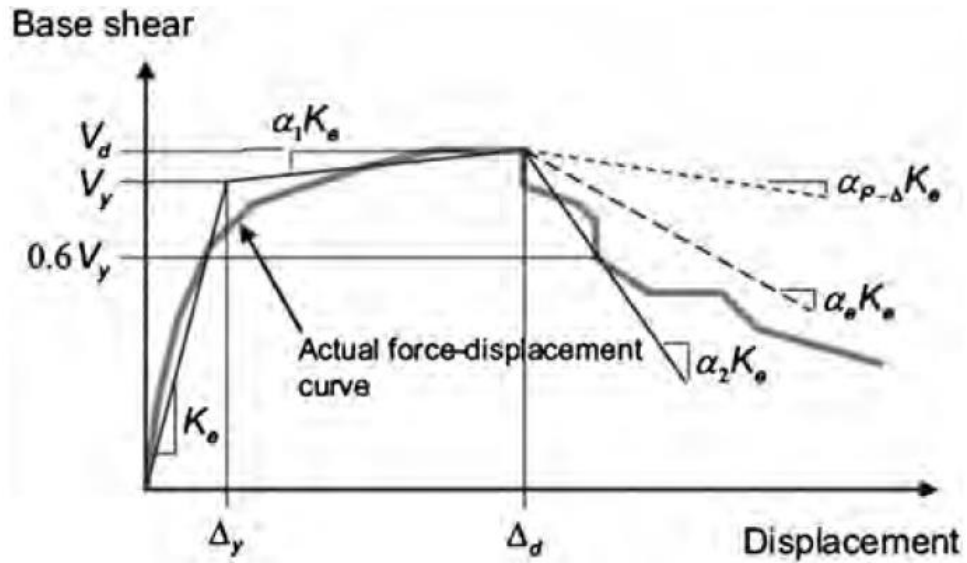


Figura 8. Curva idealizada de Fuerza vs Desplazamiento. (Fuente: ASCE 41-17).

2. Se calcula el periodo fundamental efectivo con la siguiente ecuación:

$$T_e = T_i \sqrt{\frac{k_i}{k_e}}$$

Donde:

- T_i : Periodo fundamental elástico de la estructura.
 - k_i : Rigidez lateral elástica de la estructura.
 - k_e : Rigidez lateral efectiva de la estructura.
3. Utilizando la tabla 1, del presente documento, se obtiene el factor de modificación C_o , este factor se lo utiliza para relacionar el desplazamiento espectral con el desplazamiento de techo.

Tabla 1. Factor de modificación C_0 . (ASCE 41-17)

Table 7-5. Values for Modification Factor C_0

Number of Stories	Shear Buildings ^a		Other Buildings
	Triangular Load Pattern (1.1, 1.2, 1.3)	Uniform Load Pattern (2.1)	Any Load Pattern
1	1.0	1.0	1.0
2	1.2	1.15	1.2
3	1.2	1.2	1.3
5	1.3	1.2	1.4
10+	1.3	1.2	1.5

4. El factor de masa efectiva C_m , se obtiene utilizando la tabla 2. Este valor también puede ser obtenido a partir de un análisis propio.

Tabla 2. Factor de masa efectiva C_m . (Fuente: ASCE 41-17).

Table 7-4. Values for Effective Mass Factor C_m

No. of Stories	Concrete Moment Frame	Concrete Shear Wall	Concrete Pier-Spandrel	Steel Moment Frame	Steel Concentrically Braced Frame	Steel Eccentrically Braced Frame	Other
1-2	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3 or more	0.9	0.8	0.8	0.9	0.9	0.9	1.0

NOTE: C_m shall be taken as 1.0 if the fundamental period, T , in the direction of response under consideration is greater than 1.0 s.

5. Se calcula la relación de Resistencia R .

$$R = \frac{S_a}{V_y} C_m$$

Donde:

- S_a : Aceleración espectral en el periodo fundamental efectivo de la estructura.
- V_y : Cortante de fluencia.
- C_m : Factor de masa efectiva
- W : Peso sísmico de la estructura.

6. Luego se calcula el factor de modificación C_1 , este factor relaciona los máximos desplazamientos inelásticos esperados con los desplazamientos calculados para la respuesta elástica lineal.

$$C_1 = 1 + \frac{R - 1}{\alpha T_e^2}$$

Donde:

- α : Factor de clase de sitio

Según el tipo de suelo:

A o B: $\alpha = 130$

C: $\alpha = 90$

D, E, o F: $\alpha = 60$

En los casos que el periodo sea menor a 0.1 seg., $C_1 = 0.2$, y en los casos que el periodo sea superior a 1.0 seg, $C_1 = 1.0$.

7. Se calcula el siguiente factor de modificación C_2 , este representa el efecto de forma de la histéresis afectada, la degradación de la rigidez cíclica y el deterioro de la fuerza en respuesta al máximo desplazamiento.

$$C_2 = 1 + \frac{1}{800} \left(\frac{R - 1}{T_e} \right)^2$$

8. Finalmente se calcula el desplazamiento objetivo.

$$\delta_t = C_0 C_1 C_2 S_a \frac{T_e^2}{4\pi^2} g$$

1.7 INGENIERÍA SÍSMICA BASADA EN DESEMPEÑO SÍSMICO

El FEMA menciona el desempeño sísmico como una metodología que permite diseñar edificaciones con un desempeño sísmico predecible y confiable. Esta metodología tiene como objetivo principal proteger la vida humana y controlar el daño a las estructuras, minimizando así las pérdidas económicas y sociales.

Dunn (2016), habla sobre la ingeniería sísmica basada en desempeño sísmico y la describe como:

Es el comportamiento estructural que tiene una estructura ante una demanda sísmica. Se define el nivel de desempeño sísmico determinando el desplazamiento de la estructura ante cierto evento sísmico y comparándolo con ciertos límites de desplazamiento.

Por otro lado, Medina y Music, (2018), describe que: “La Ingeniería Sísmica basada en el desempeño busca como objetivo que los daños esperados estén dentro de márgenes preestablecidos para diferentes sismos de diseño con un aceptable nivel de confiabilidad.”

En Ecuador, la Norma Ecuatoria de la Construcción (NEC-15), define 4 niveles de desempeño:

- 1-A: NIVEL OPERACIONAL.
- 2-B: NIVEL DE OCUPACIÓN INMEDIATA
- 3-C: NIVEL DE SEGURIDAD DE VIDA
- 4-E: NIVEL DE PREVENCIÓN AL COLAPSO

Así mismo, NEC - 15, presenta la siguiente tabla para el control de daño y nivel de desempeño para edificios:

Tabla 3. Niveles de daño y desempeño para edificios. (Fuente: NEC-15).

	Nivel de Prevención al Colapso (5-E)	Nivel de Seguridad de Vida (5-E)	Nivel de Ocupación Inmediata (1-B)	Nivel Operacional (1-A)
Daño Global	Severo	Moderado	Ligero	Muy Ligero
General	Pequeña resistencia y rigidez residual, pero columnas y muros cargadores funcionando. Grandes derivas permanentes. Algunas salidas bloqueadas. Parapetos no asegurados que han fallado o tienen alguna falla incipiente. El edificio está cerca del colapso	Algo de resistencia y rigidez residual ha quedado en todos los pisos. Elementos que soportan cargas gravitacionales aún funcionando. Fallas en muros dentro de su plano o parapetos inclinados. Algo de deriva permanente. Daño en paredes divisorias. El Edificio se mantiene económicamente reparable	No hay deriva permanente. La estructura aún mantiene resistencia y rigidez originales. Fisuras menores en fachadas, paredes divisorias, cielos razos así como en elementos estructurales. Los ascensores aún pueden ser encendidos. Sistema contra incendios aún operable	No hay deriva permanente. La estructura aún mantiene la resistencia y rigidez originales. Fisuras menores en fachadas, paredes divisorias, y cielos razos así como en elementos estructurales. Todos los sistemas importantes para una operación normal están en funcionamiento
Componentes No Estructurales	Daño severo	Peligro de caída de objetos mitigado pero bastante daño en sistemas: arquitectónico, mecánico y eléctrico	Equipos y contenido están seguros de manera general, pero algunos no operan debido a fallas mecánicas o falta de utilidad	Ocurre daño insignificante. La energía eléctrica y otros servicios están disponibles, posiblemente por servicios de reserva

1.8 UMBRALES DE DAÑO

El umbral de daño (S_{ds}), es un límite a partir del cual se determina que la estructura tendrá un estado de daño “ds”. En el presente trabajo se prevé calcular el daño de las estructuras en función del desplazamiento espectral “Sd”.

La RISK UE, define 4 “ds”: Leve, moderado, extensivo y completo. Así mismo, propone definir los ds, de forma simplificada, a partir del espectro de capacidad en su representación bilineal. Partiendo de los valores (D_y , A_y) y (D_u , A_u), se definen los desplazamientos espectrales correspondientes a los umbrales de los 4 estados de daños “ds_i”: (como se citó en Vargas, Pujades, Barbat, & Hurtado, 2013)

- $ds_1 = 0.7D_y$ (Leve)
- $ds_2 = D_y$ (Moderado)
- $ds_3 = D_y + 0.25(D_u - D_y)$ (Extenso)
- $ds_4 = D_u$ (Completo)

1.9 CURVAS DE FRAGILIDAD

Las curvas de fragilidad es una metodología gráfica, basada en probabilidades, donde se puede observar la probabilidad en que una estructura alcance un límite de daño, en base a un parámetro sísmico especificado.

Agudelo y López (2009), definen las curvas de fragilidad como:

Las curvas de fragilidad representan la probabilidad de que una estructura exceda un determinado estado de daño en función de un parámetro que define la intensidad sísmica. Estas curvas sirven para estimar el riesgo sísmico de grupos de edificios con características estructurales similares o de una estructura esencial.

Así, Pitilakis, Crowley y Kaynia, citado por (Calderón, 2018), dicen que: “las curvas de fragilidad son una herramienta clave en la determinación del riesgo sísmico, éstas permiten obtener probabilidades de exceder un estado de daño específico de acuerdo con un parámetro de demanda sísmica”.

Este parámetro de demanda puede ser el valor de la aceleración máxima del terreno (PGA), el desplazamiento espectral (Sd) o la aceleración espectral (Sa).

Así mismo, Coronel y López (2013), destacan que:

Las curvas de fragilidad permiten representar en términos probabilísticos la vulnerabilidad sísmica de las edificaciones y proporcionan la probabilidad de que la respuesta de una estructura alcance o exceda determinado límite asociado con el daño, como función de un parámetro indicador de la intensidad sísmica.

Por otro lado, para Díaz, “Las curvas de fragilidad representan un modo para determinar la vulnerabilidad sísmica en términos probabilísticos relacionando la respuesta estructural con la demanda sísmica, cuyo uso es recomendable para edificaciones individuales, esenciales y en zonas carente de registros sísmicos históricos”. (como se citó en Sáenz, 2019)

Además, Díaz también hace referencia que estas curvas “representan la probabilidad de que la respuesta de una estructura exceda un determinado estado límite, en función de un parámetro que define la intensidad del movimiento del suelo” (como se citó en Sáenz, 2019).

Rojas (2010), está de acuerdo con los otros autores, puesto que describe a las curvas de fragilidad como:

Son curvas que permiten representar la probabilidad de excedencia de un estado límite de daño (estado de daño) como una función de un parámetro representativo de la severidad del movimiento sísmico (aceleración, intensidad, etc.) o de la respuesta estructural (deriva, desplazamiento, etc.).

Según Coronel, López y Betancourt (2010):

Las curvas de fragilidad permiten representar en términos probabilísticos la vulnerabilidad sísmica de las edificaciones y proporcionan la probabilidad de que la respuesta de una estructura alcance o exceda determinado límite asociado al daño, como función de un parámetro indicador de la intensidad sísmica.

Las curvas de fragilidad indican la probabilidad de que se alcance o supere un estado de daño específico en relación con un parámetro que representa la acción sísmica. En este trabajo, dicho parámetro será el desplazamiento espectral.

Según Vargas, Pujades, Barbat y Hurtado (2013), para construir las curvas de fragilidad se deben considerar las siguientes hipótesis:

- 1) En los desplazamientos espectrales de los umbrales de cada estado de daño ds_i , la probabilidad de que se iguale o exceda el estado de daño es del 50%.
- 2) Las curvas de fragilidad siguen una función de probabilidad lognormal acumulativa descrita por la siguiente ecuación:

$$P [ds_i/sd] = \phi \left[\frac{1}{\beta_{ds_i}} \ln \left(\frac{sd}{ds_i} \right) \right]$$

Donde:

- ds_i = El valor medio del desplazamiento espectral en el que la estructura alcanza el umbral de daño ds .
- β_{ds_i} = Desviación estándar del logaritmo natural del desplazamiento espectral.
- sd = Desplazamiento espectral.
- ϕ = Función de distribución estándar acumulada.

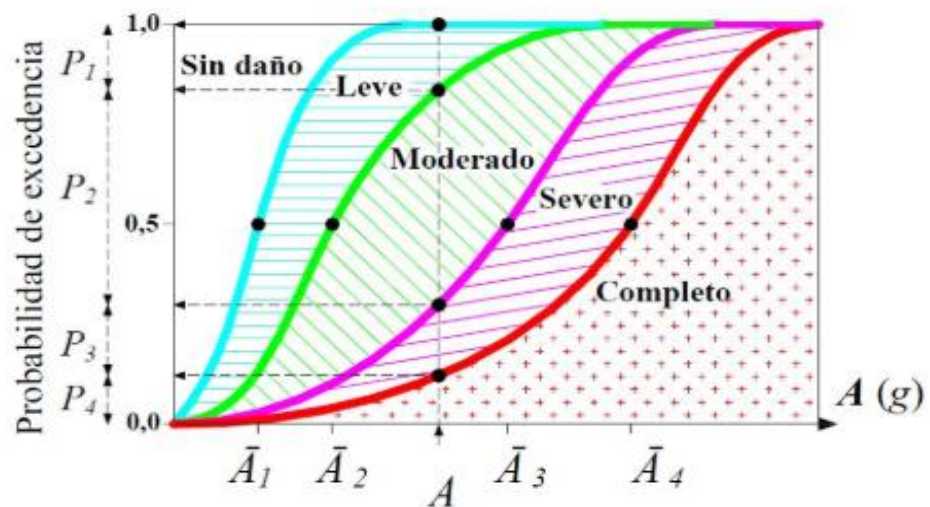


Figura 9. Representación de las curvas de fragilidad según el estado de daño. (Fuente: Coronel, López y Betancourt, 2010.)

Si bien, las curvas de fragilidad es un análisis basado en probabilidad estadística, para lograr una comparación entre la probabilidad de excedencia, y un parámetro sísmico que describa el daño de la estructura, es necesario contar con diferentes análisis previos, según la metodología que se use para el desarrollo.

1.9.1 MÉTODO ANALÍTICO

Para calcular curvas de fragilidad, existen diversas metodologías, así como se observa en la figura 10. Sin embargo, en este proyecto, se hará énfasis en el método analítico.

Así, Calderón (2018), menciona que:

Estos métodos adoptan distribuciones de daño realizadas mediante análisis de modelos estructurales tanto dinámicas, con modelos de aceleración en el tiempo; como con modelos estáticos como espectros de respuesta. En general resulta en una amplia credibilidad de la estimación de la vulnerabilidad.

Siguiendo el esquema de la figura 10, uno de los métodos analíticos para calcular curvas de fragilidad es mediante el método del espectro de capacidad (CSM), mencionado en la sección 1.5. que es la utilizada por la metodología HAZUS y la LM2 del Risk-UE, además es la metodología que se llevará a cabo en este proyecto de titulación.



Figura 10. Metodologías para calcular curvas de fragilidad. (Fuente: Pitilakis et al., 2014, citado en Calderón, 2018).

CAPÍTULO II: DESARROLLO

En el presente capítulo, se presenta la elaboración del código que se ha desarrollado para cumplir con los objetivos señalados en la introducción. Se presentarán la elaboración de dos metodologías para encontrar los puntos de desempeño: Método de los coeficientes ASCE 41-17 y FEMA440 método C.

2.1 Método de los coeficientes ASCE 41-17

2.1.1 ARCHIVOS DE ENTRADA – “INPUT”

Puesto que, la finalidad del proyecto de titulación es la obtención de curvas de fragilidad para todo tipo de estructura, se debe enfatizar que no se realizará el análisis no lineal (Push Over), para la obtención de la curva de capacidad, por lo tanto, este análisis deberá ser un “input” o archivo de entrada, y posterior, cuando se realice su respectivo análisis se debe establecer cuáles serán los datos necesarios o inputs (archivos de entrada) que necesita el código para realizar su respectivo análisis.

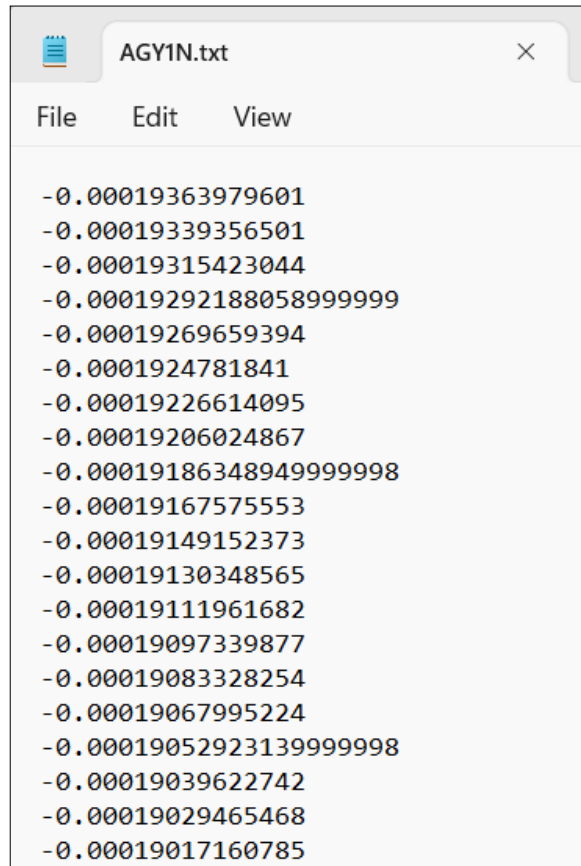
Se debe mantener un orden, por lo que, el archivo Python “.py” debe ser guardado en una carpeta, dentro de la cual, se debe contar con las siguientes sub - carpetas:

- Registros Sísmicos
- DatosPushOver

2.1.2 REGISTROS SÍSMICOS

Se debe contar con una gran cantidad de registros sísmicos, estos se pueden obtener de las páginas oficiales, por ejemplo: PEER, RENAC; se requiere que estos registros se encuentren en unidades de GALS $\left(\frac{cm}{seg^2}\right)$, es importante, ya que, el código realiza el Método de Newmark que se menciona en la sección de marco teórico, este método es necesario para obtener los espectros de respuesta de aceleración y desplazamiento.

Así mismo se debe contar con un formato para que el código lea los archivos de los registros sísmicos. Los datos del archivo de texto (.txt) de cada sismo deben comenzar desde la primera fila y en una sola columna. En la Figura 11, se muestra como debe ser el formato establecido. Todos los archivos de texto deben estar en una carpeta llamada “Registros Sísmicos”.



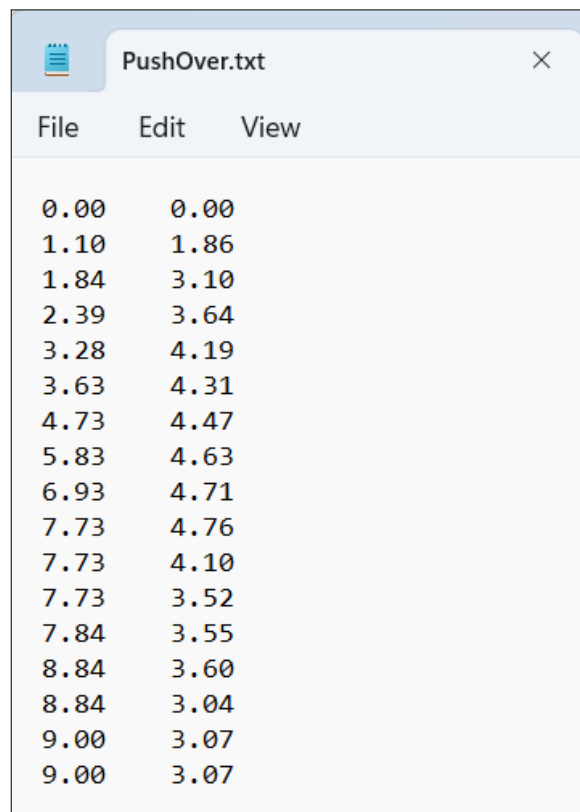
```
AGY1N.txt
File Edit View
-0.00019363979601
-0.00019339356501
-0.00019315423044
-0.00019292188058999999
-0.00019269659394
-0.0001924781841
-0.00019226614095
-0.00019206024867
-0.00019186348949999998
-0.00019167575553
-0.00019149152373
-0.00019130348565
-0.00019111961682
-0.00019097339877
-0.00019083328254
-0.00019067995224
-0.00019052923139999998
-0.00019039622742
-0.00019029465468
-0.00019017160785
```

Figura 11 Formato para cada archivo de texto correspondiente a cada registro sísmico. Fuente: Elaboración propia.

2.1.3 DATOS DEL ANÁLISIS PUSHOVER

Previamente, se debe haber realizado un análisis no lineal (Pushover), de la estructura de interés, para obtener la curva de capacidad, los datos de esta curva deben estar en un archivo en formato de texto (.txt).

Uno de los resultados de un análisis no lineal y la obtención de curva de capacidad son una lista de puntos en el cual, en el eje de las abscisas se ubica el desplazamiento y en el eje de las ordenadas se ubican la fuerza Cortante del último piso. A continuación, se presenta el formato que se debe seguir para la creación del archivo de texto que contiene la información de curva de capacidad. El nombre del archivo de texto será: "PushOver". En Figura 12, se puede observar el formato correspondiente, donde, la primera columna corresponde a desplazamientos en unidades de centímetros (cm) y en la segunda columna la fuerza cortante en unidades de toneladas (ton). Este archivo de texto debe guardarse en una carpeta llamada: "DatosPushOver".



File	Edit	View
0.00	0.00	
1.10	1.86	
1.84	3.10	
2.39	3.64	
3.28	4.19	
3.63	4.31	
4.73	4.47	
5.83	4.63	
6.93	4.71	
7.73	4.76	
7.73	4.10	
7.73	3.52	
7.84	3.55	
8.84	3.60	
8.84	3.04	
9.00	3.07	
9.00	3.07	

Figura 12. Formato Curva de Capacidad (PushOver). Fuente: Elaboración Propia.

2.1.4 CURVA BILINEAL

En los puntos anteriores, se mencionan los datos que se necesitan en la carpeta junto con el código Python, sin embargo, se necesitan otros valores de inputs para que el código desarrolle las curvas de fragilidad.

Una vez el código este en ejecución, y si se ha insertado correctamente los archivos texto, lo primero que mostrará es la curva de capacidad y pedirá las coordenadas:

- $(D_{ui}; A_{ui})$: Este punto corresponde al punto último de la curva de capacidad. La posición se basa en el punto con par de coordenada que se muestra, por ejemplo, para el par de coordenada (0,0) se toma como 1, y en este caso, el punto 10 será, el punto de coordenadas "último" de la curva de capacidad.
- $D_{ui} = \text{desplazamiento último } (\Delta)$ y $A_{ui} = \text{cortante último } (V)$.

No se debe insertar el valor de las coordenadas, se debe insertar el número del punto que corresponde dicha coordenada, así como se puede apreciar en la Figura 13.

Este método busca representar bilinealmente la curva de capacidad con el último punto de fluencia conocido como DU-AU, esto permitirá calcular la máxima ductilidad permitida a la estructura, su periodo inicial, así también la rigidez lateral y rigidez efectiva de la estructura.

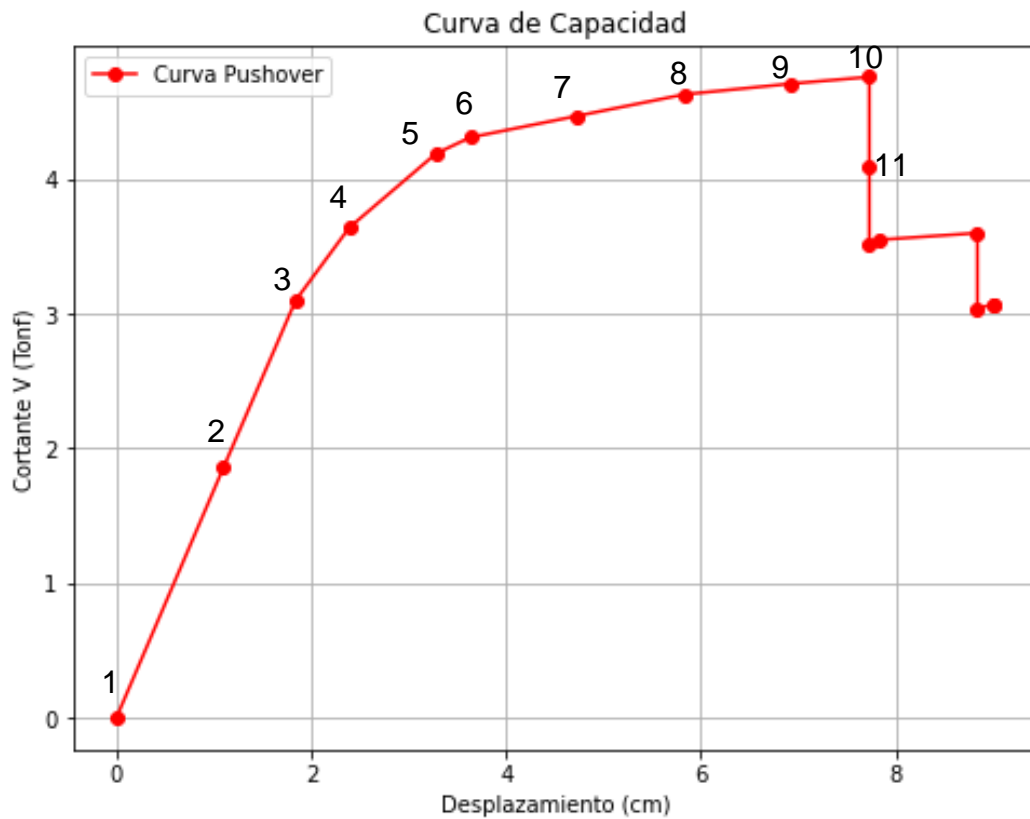


Figura 13. Número de punto DU-AU. Fuente: Elaboración propia.

Luego el terminal pedirá las coordenadas ($D_{yi}; A_{yi}$) que son propuestas por la persona que use el código. Estas coordenadas sí deberán ser tomadas en función del valor del desplazamiento y el cortante esperado. En la Figura 14, se muestra cómo tomar estas coordenadas. Dado que necesitamos este par de coordenadas para realizar la representación bilineal de la curva de capacidad que se menciona en la sección 1.5.3.

En este caso se utiliza las coordenadas (2,4). Se debe inserta en formato #desplazamiento, #cortante ($\# \Delta d; \# Vd$).

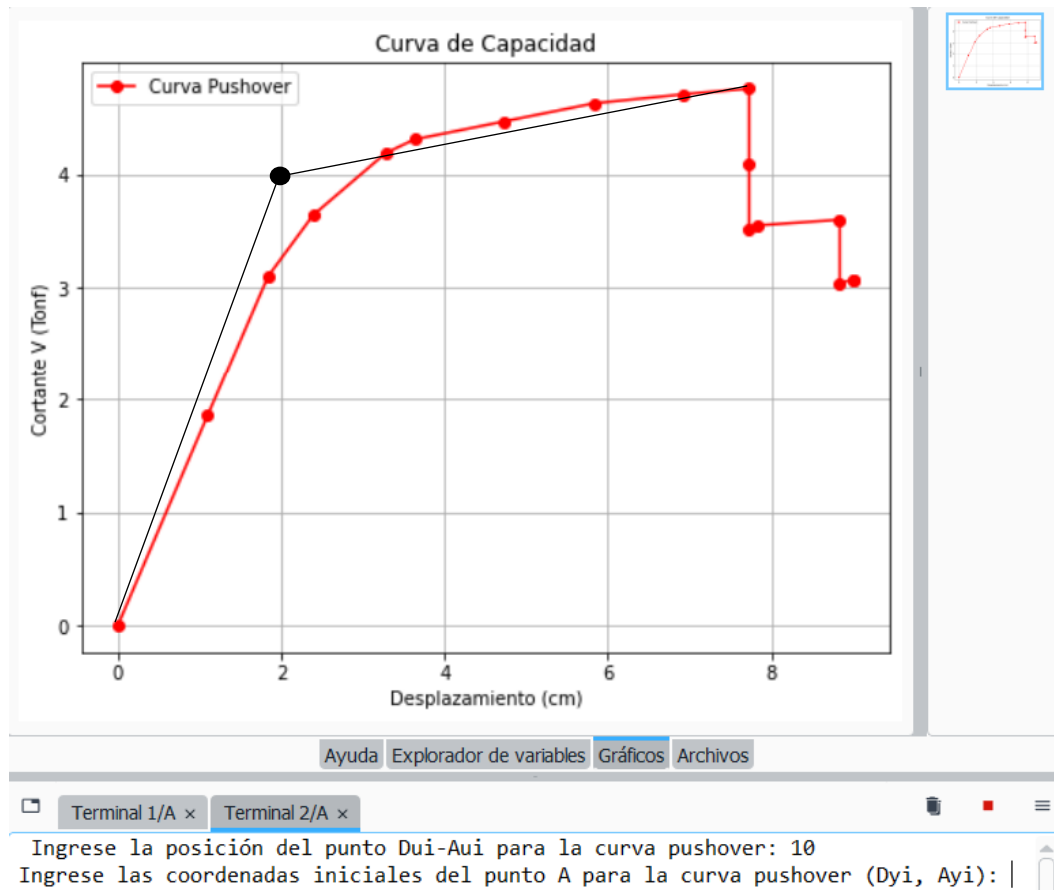


Figura 14. Obtención de coordenadas D_{yi} , A_{yi} . Fuente: Elaboración propia

Después, el terminal presentará el área bajo la curva de la representación bilineal y de la curva de capacidad o PushOver como se muestra en la Figura 15. Será decisión del usuario calcular nuevas coordenadas de ese punto.

- Si la respuesta es “Si” el código buscará ese par de coordenadas que minimice el error de la representación bilineal.

```

Área bajo la curva Pushover: 28.44
Área bajo la curva de la Representación bilineal inicial: 29.10
Error de la representación bilineal inicial: 2.30%
¿Calcular nuevas coordenadas para el punto A? (si/no): si
Nuevas coordenadas del punto A:  $D_{yi} = 1.92$ ,  $A_{yi} = 3.84$ 
Área bajo la curva de la Representación bilineal con el punto A encontrado: 28.68
Error de la representación bilineal: 0.82%
Rigidez lateral efectiva  $K_e$ : 2.00 Tonf/cm
Rigidez inicial  $K_i$ : 1.68 Tonf/cm

```

Figura 15. Análisis de la representación bilineal. Fuente: Elaboración propia.

Como se observa en la Figura 16, lo que nos pide a continuación es el peso efectivo sísmico W , este valor debe ser insertado en toneladas.

A continuación, se pide insertar los valores de participación modal (α), y factor de forma (PF), si se conocen estos valores, se procede a insertarlos, caso contrario, se asumirán valores de $\alpha = 0.8$ y $PF = 1.4$. Revisar la sección 1.6.2.

La curva Bilineal se transforma a formato ADRS como se observa en la Figura 17, la cual es necesaria para el cálculo de periodo inicial de la estructura.

```

Ingresar Peso Efectivo sísmico W (Ton): 19
¿Conoce los valores de factor de participación modal ( $\alpha$ ) y factor de forma de la
estructura (Pf)? (si/no): si
Ingrese factor de participación modal ( $\alpha$ ): 1
Ingrese factor de forma de la estructura (Pf): 1
Factor de participación modal ( $\alpha$ ): 1.0
Factor de forma de la estructura (Pf): 1.0
Período fundamental elástico  $T_i$ : 0.62 segundos
    
```

Figura 16. Análisis para conversión ADRS. Fuente: Elaboración propia.

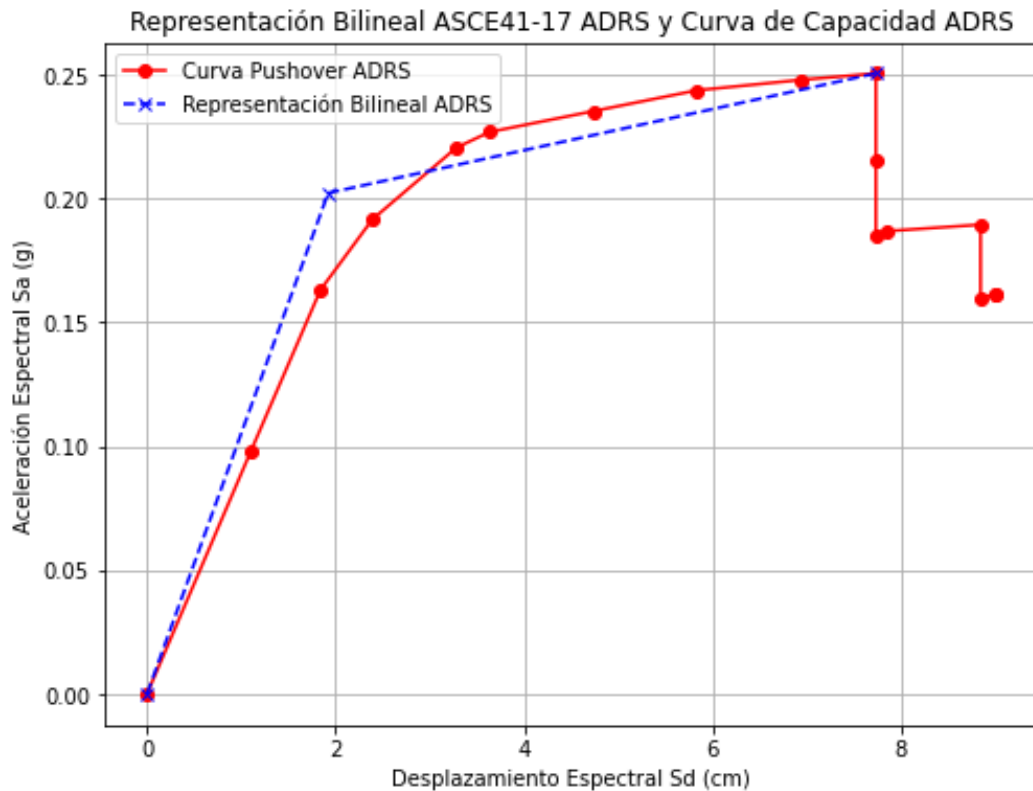


Figura 17. Representación Bilineal formato ADRS. Fuente: Elaboración propia.

2.1.5 PUNTOS DE DESEMPEÑO ASCE 41-17

Una vez realizado la conversión ADRS de la representación bilineal y los cálculos necesarios como rigidez y periodo inicial, el terminal pedirá los datos necesarios para realizar el cálculo de puntos de desempeño según el ASCE 41-17.

En la Figura 18, los inputs que se piden son:

- El número de pisos de la estructura, dependiendo del número de pisos se tomará Valores de C_o que se encuentran en las tablas del ASCE41-17.
- El valor de C_m , este se debe a que el código resuelve puntos de desempeño para estructuras de hormigón armado.
- Tipo de suelo, se debe insertar el tipo de suelo, en caso de no conocerlo, se lo debe asumir, en este estudio el tipo de suelo es D.

Además de estos coeficientes el código obtiene el periodo Efectivo T_e , el cual es un dato importante para obtener del Desplazamiento Objetivo δ_t (cm), ya que dicho valor permite la interpolación para obtener el valor S_d correspondiente a registro sísmico analizado.

```
Ingrese el número de pisos del edificio: 2
Valor de Co: 1.2
Valor de Cm: 1
¿Conoce el tipo de Suelo? (si/no): si
Ingrese Tipo de Suelo (A, B, C, D, E, F): D
Valor de a: 60
Período fundamental efectivo Te: 0.57 segundos
¿Desea ver la gráfica de puntos de desempeño correspondiente a cada sismo? (si/no): no
```

Figura 18. Análisis para Desplazamiento Objetivo δ_t (cm). Fuente: Elaboración Propia

El usuario puede ver las gráficas individuales de cada punto de desempeño, es decir, el respectivo punto de desempeño según el registro sísmico de análisis; de no ser requerido, se muestra directamente la gráfica Desplazamiento Espectral S_d (cm) vs Desplazamiento Objetivo δ_t (cm).

Esta gráfica es necesaria para el análisis de curvas de fragilidad, debido a que permitirá la creación e interpretación de los umbrales de daño y sus respectivos casos de excedencia. Posteriormente se realiza el análisis estadístico propuesto para la obtención de curvas de fragilidad.

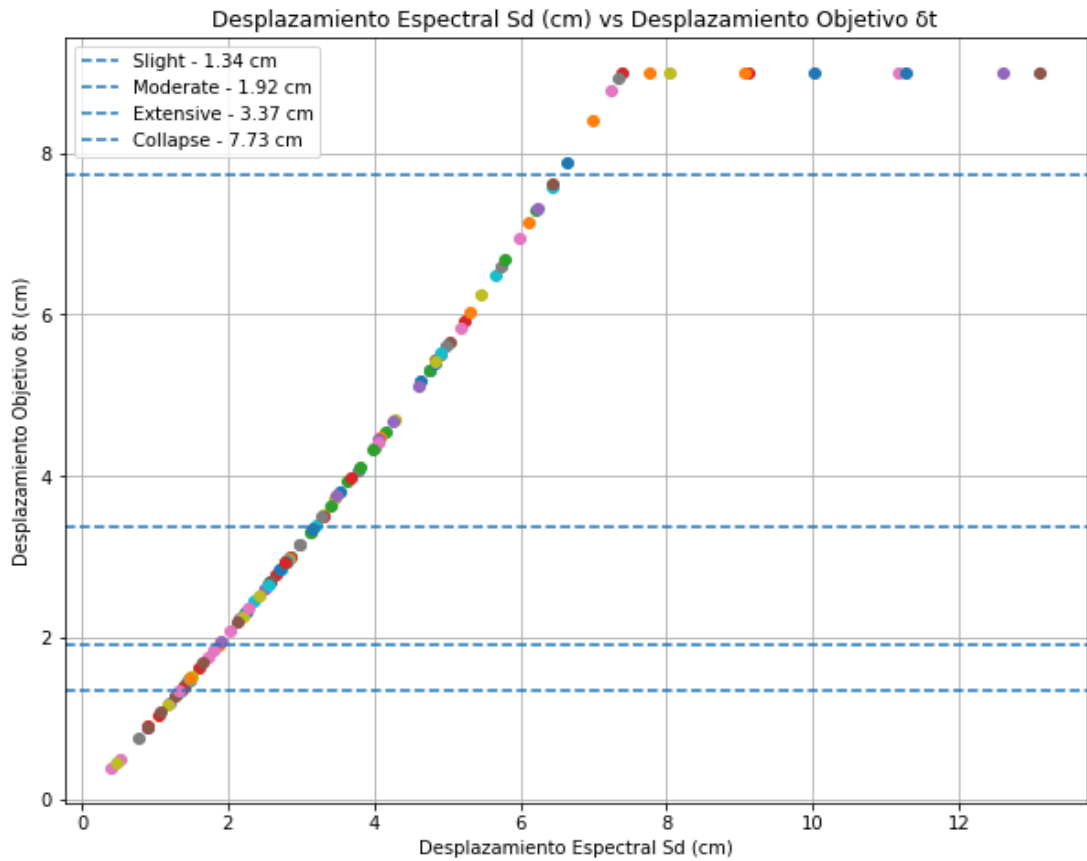


Figura 19. Desplazamiento Espectral Sd (cm) vs Desplazamiento Objetivo δt (cm). Fuente: Elaboración propia.

2.1.6 CURVAS DE FRAGILIDAD ASCE 41-17

Una vez realizado los cálculos y sus respectivas representaciones gráficas anteriores, el código analiza los puntos que exceden los límites de daño establecidos, estos límites de daño son mencionados en la sección 1.8.

El valor input de insertar valor de intervalo, se usa para establecer un análisis de rango y calcular el promedio de los puntos que se encuentran dentro de dicho rango, se sugiere que este valor sea= 1, 2, 2.5.

Para finalizar el último valor input se trata de insertar el último valor de análisis considerado por el usuario basado en el desplazamiento objetivo, tomando como ejemplo la Figura 19, se aprecia que los valores de intervalos son de 2 y el último valor de análisis es 8, ya que pasado este valor solo existe colapso o daño extenso.

```

Daño Leve: 17
Daño Moderado: 32
Daño Extenso: 44
Daño Completo: 14
Insertar valor de intervalo: 2
Inserte último valor de análisis: 8
Promedio Intervalo 1 = 0.0129 m
Promedio Intervalo 2 = 0.0290 m
Promedio Intervalo 3 = 0.0486 m
Promedio Intervalo 4 = 0.0679 m
    
```

Figura 20. Análisis para curvas de fragilidad. Fuente: Elaboración propia.

En la Figura 21, se presenta las curvas de fragilidad obtenidas para la Curva de capacidad (PushOver) que se analizó junto con sus respectivos registros sísmicos

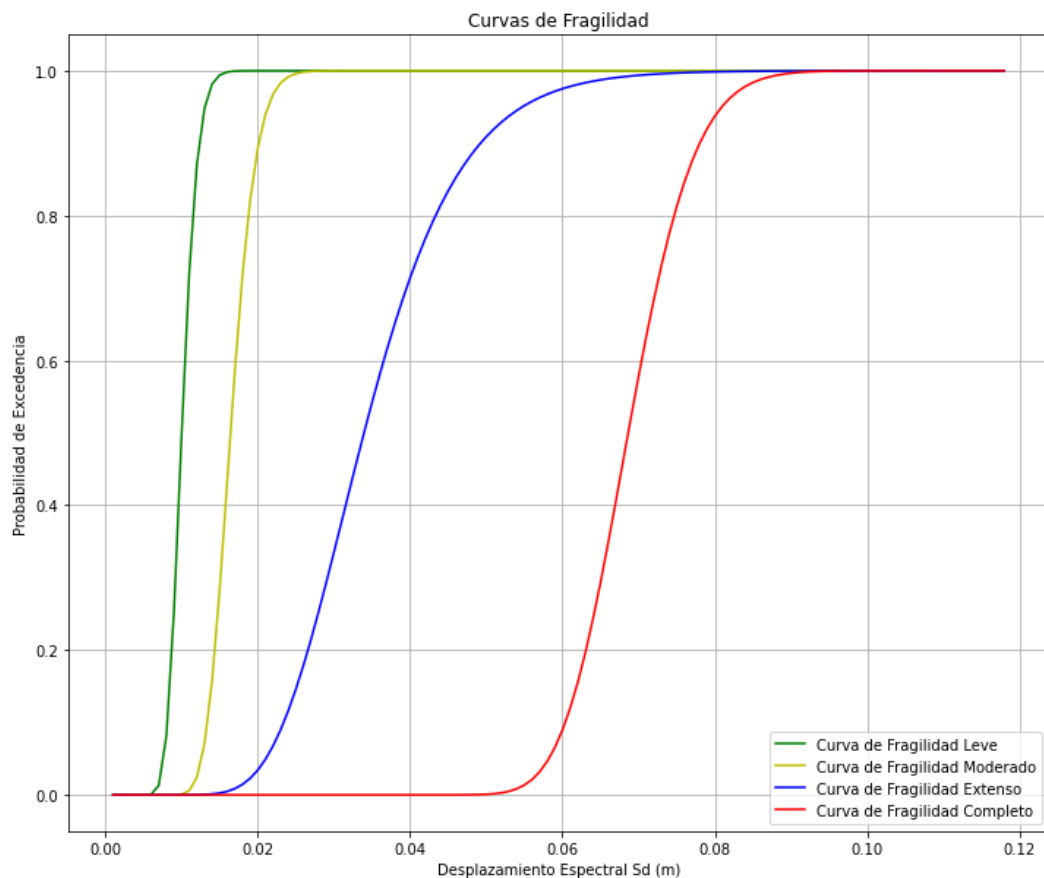


Figura 21. Curva de Fragilidad método de coeficientes propuesto por el ASCE41-17. Fuente: Elaboración Propia

2.2 FEMA440 Método C (gráfico).

Para este método se realiza los primero cuatro pasos mencionados en la sección 2.1.4.; el cambio comienza en los puntos de desempeño, el cual se trató en la sección del método C. A continuación, se presentarán los resultados obtenidos para un registro sísmico, en el capítulo 3, se mostrarán más resultados de otro análisis para otra curva de capacidad.

En la Figura 22, se presenta los resultados que muestra el terminal de Python al momento de ingresar los valores input, se debe mencionar, que no presentan en la tabla dentro del terminal, todos los factores utilizados y propuestos por el método B, para las reducciones del registro sísmico analizado.

El factor Reduction (M/B) es con el cual se reduce el registro sísmico del 5% este valor va disminuyendo mientras se obtenga una ductilidad mayor hasta llegar a la ductilidad máxima calculada.

```
Ingrese la masa del sistema (Ton): 19
Ingrese el factor de participación modal (α): 1
Ingrese el factor de forma de la estructura (Pf): 1
Rigidez post-elástica: 0.08
Ductilidad última esperada (μ): 4.02
Periodo Inicial (T0): 0.62
  Ductilidad μ   Periodo Efectivo Teff   ...   Au   Reduction(M/B)
0   1.000000     0.618311   ...   0.202231   0.997641
1   1.500000     0.646290   ...   0.210217   0.718093
2   2.000000     0.718478   ...   0.218204   0.623777
3   2.500000     0.817253   ...   0.226190   0.603668
4   3.000000     0.924994   ...   0.234177   0.613332
5   3.500000     1.024078   ...   0.242164   0.630128
6   4.000000     1.032580   ...   0.250150   0.561935
7   4.023544     1.034472   ...   0.250526   0.561458
```

Figura 22. Análisis para reducción de registro sísmico. Fuente: Elaboración Propia.

2.2.1 PUNTOS DE DESEMPEÑO FEMA 440 MÉTODO C

El presente método busca reducir el espectro de desplazamientos espectrales (S_d) vs Aceleraciones espectrales (S_a) del 5%, es decir con ductilidad igual a 1; también con intervalos dados por diferentes valores de

ductilidad, en el caso de estudio estos intervalos son cada 0.5 hasta un amortiguamiento efectivo (β_{eff}) correspondiente con el ultimo valor de ductilidad calculado como se muestra en la Figura 22.

Además de presentar los Periodos Secantes correspondientes a cada curva Sd vs Sa reducidas. Esto busca encontrar un punto de desempeño dentro de la curva de capacidad en formato ADRS. Si con la máxima ductilidad calculado junto con su respectiva curva Sd vs Sa no interseca con la curva de capacidad, se toma como punto de desempeño el último punto de la curva de capacidad ADRS.

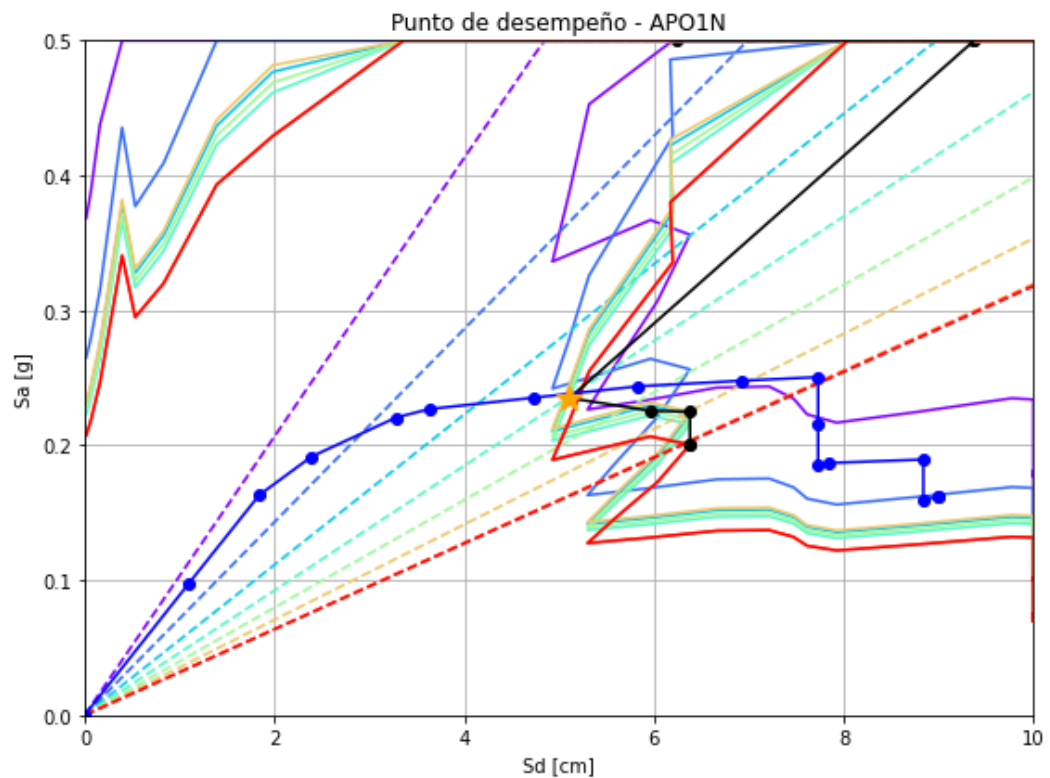


Figura 23. Punto de desempeño propuesto por el FEMA440 en el Método C. Fuente: Elaboración Propia.

Una vez obtenido todos los puntos de desempeño propuestos por el método C, se realiza un proceso similar a 1.6.1.3., en este caso, no se usará el periodo T_e para la interpolación y obtención del desplazamiento espectral S_d , se usará el periodo inicial (T_0) calculado para la estructura.

Obtenido esos valores y los valores S_d de los puntos de desempeño se procede a la obtención de la gráfica Desplazamiento Espectral S_d (cm) vs S_d Performance Point (cm), estas debe tener unidades consistentes para su posterior análisis de umbrales de daño como se muestra en la Figura 24, en esta figura también se observa que el comportamiento de los puntos presentan una variación y dejó de ser lineal comparado con el anterior método, en consecuencia a eso la estadística y probabilidad puede estar más cerca de la realidad.

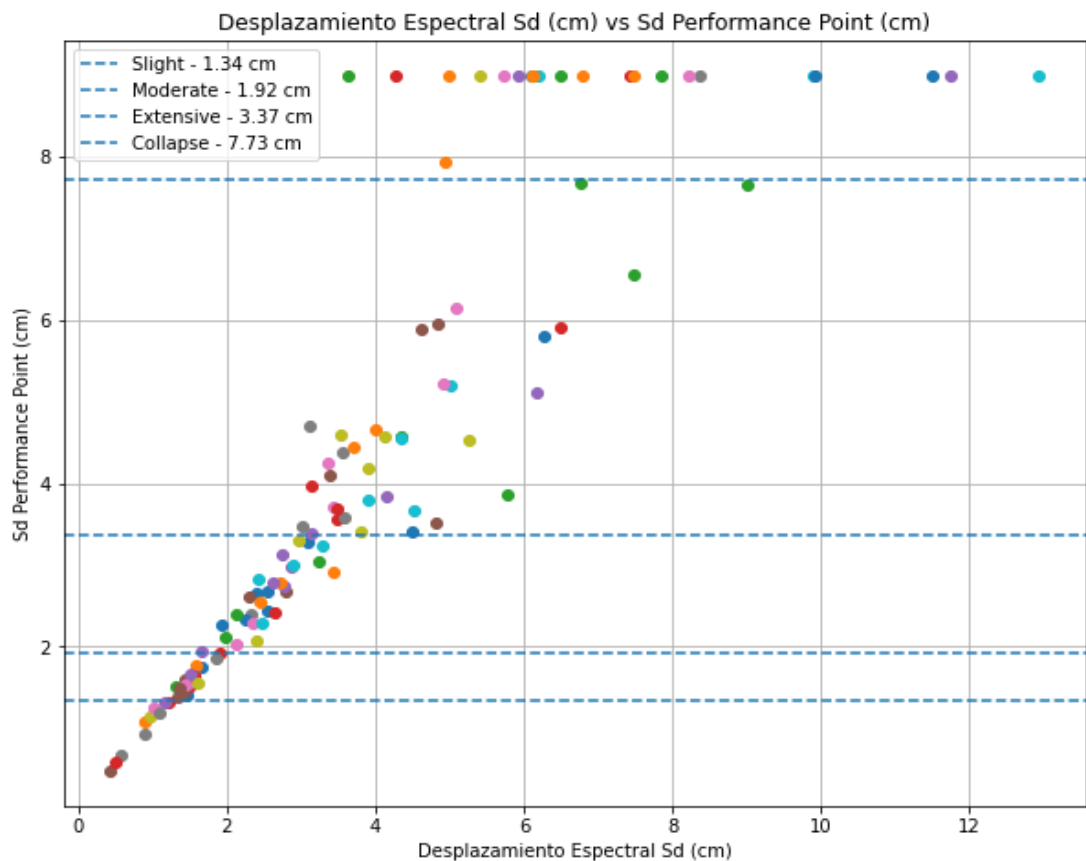


Figura 24. Desplazamiento Espectral S_d (cm) vs S_d Performance Point (cm). Fuente: Elaboración Propia.

2.2.2 CURVAS DE FRAGILIDAD FEMA440 Método C

Para la obtención de las curvas de fragilidad, se necesita la gráfica Desplazamiento Espectral S_d (cm) vs S_d Performance Point (cm) Figura 24, con esta se puede realizar la clasificación, interpretación y análisis de los datos obtenidos. La terminal de este método presenta el mismo input que se mostraban en el anterior método. En la Figura 25, se puede observar el resultado obtenido para el mismo caso de estudio.

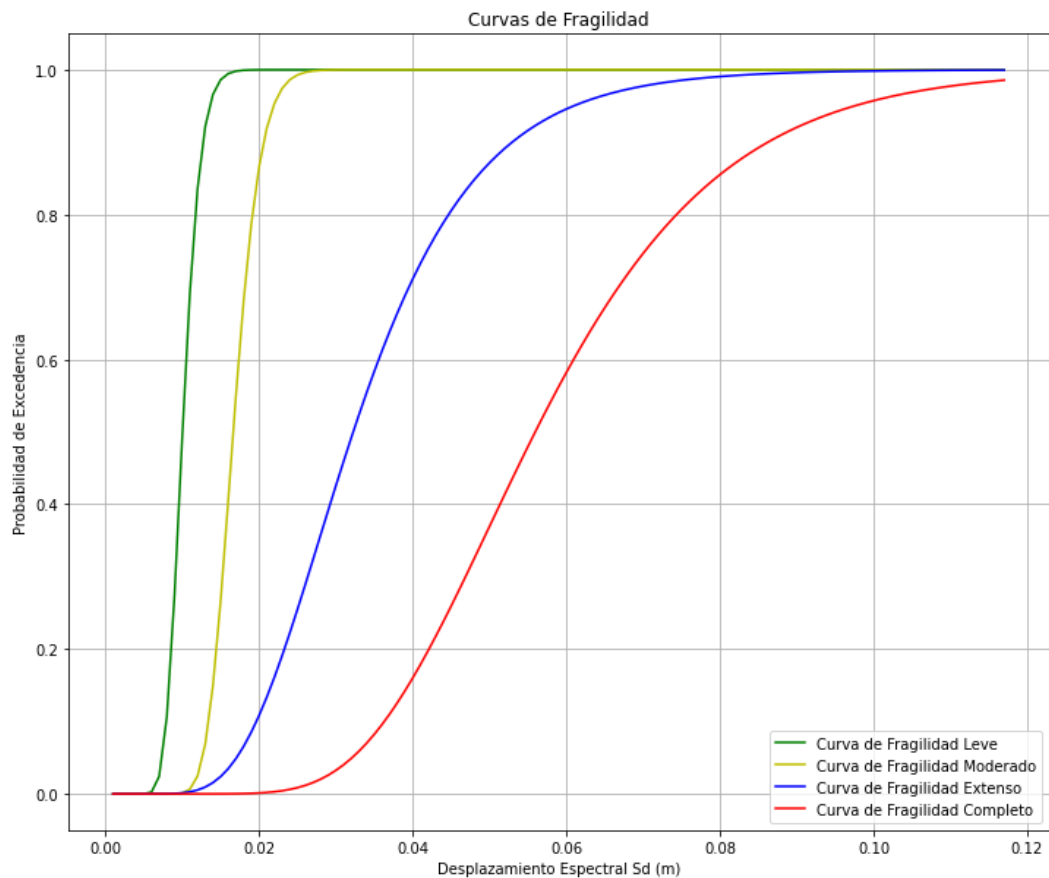


Figura 25. Curva de Fragilidad método de coeficientes propuesto por FEMA 440 Método C. Fuente: Elaboración Propia

3 CAPÍTULO III: RESULTADOS

3.1 Resultados método ASCE 41-17

Dado que en la sección anterior se presentó el desarrollo del código para una determinada curva de capacidad en este apartado presentaremos los resultados para dicha curva junto con los resultados de otra curva de capacidad, que desarrolla una ductilidad mayor a la anterior. Para poder distinguir ambos resultados se llamará Curva de Capacidad 1 a la curva que se hizo el análisis en el desarrollo y Curva de Capacidad 2 a los nuevos resultados.

Dichos análisis se hicieron con los mismos registros sísmicos para cada uno, por ende, el número de casos analizados para ambas curvas de capacidad son iguales.

3.1.1 Curva de Capacidad

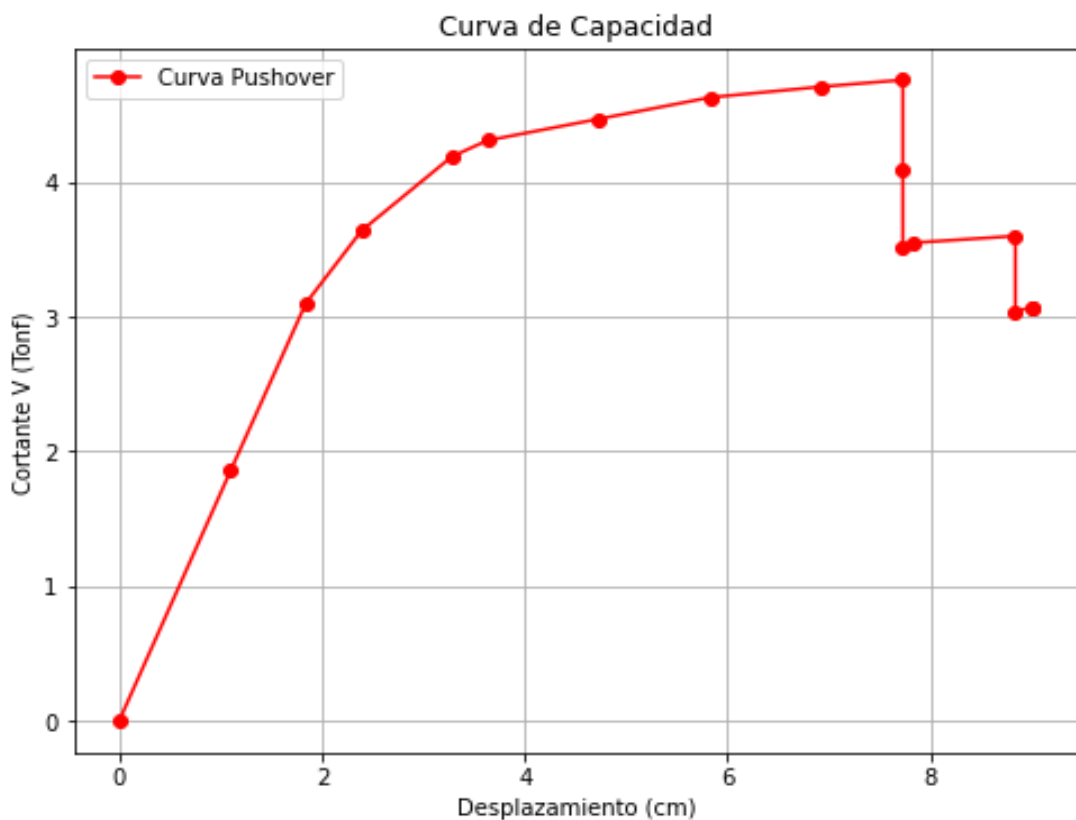


Figura 26. Curva de Capacidad 1 para su respectivo análisis. Fuente: *Elaboración Propia*

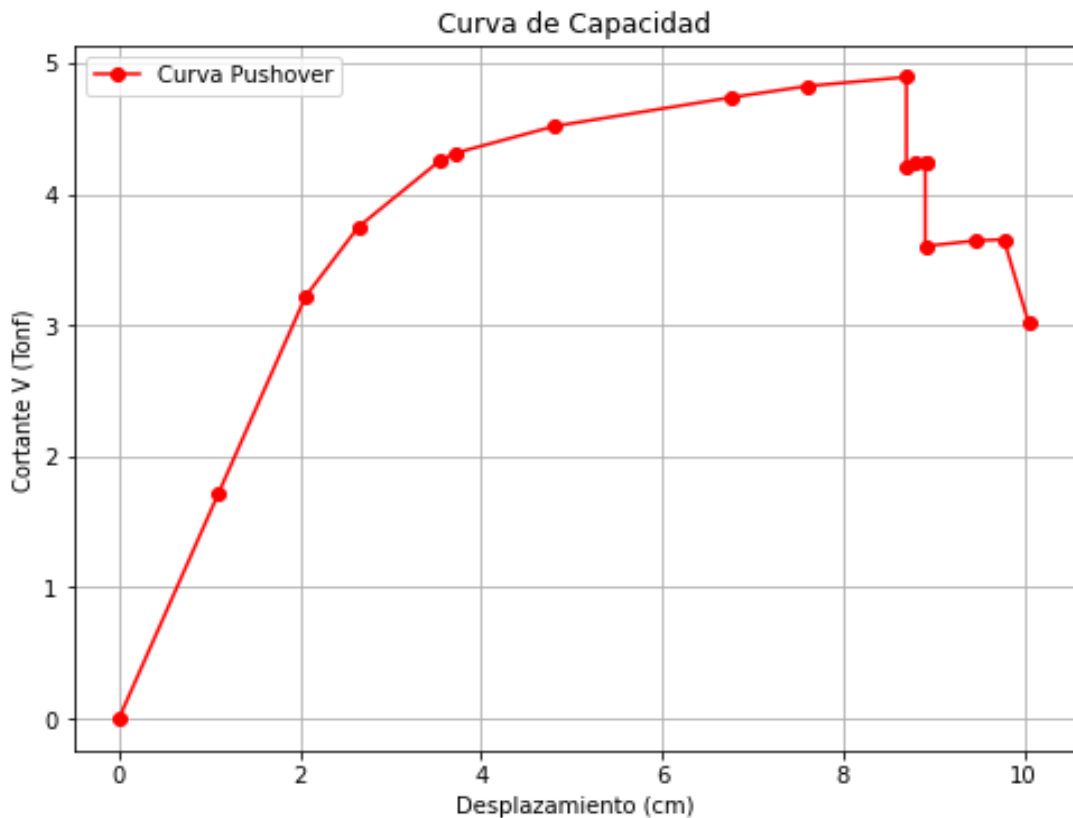


Figura 27. Curva de Capacidad 2 para su respectivo análisis. Fuente: Elaboración Propia

3.1.2 Análisis de la Curva de capacidad

En la Figura 28 y Figura 29, podemos observar los inputs ya antes descritos en la sección 2.1.4. y obteniendo como resultado para ambas curvas de capacidad un error menor al 1% en su respectiva representación Bilineal. Proceso el cual se basó en la igualación de áreas.

Además, se recalca en la necesidad de saber los valores del factor de participación modal y el factor de forma de la estructura los cuales son esenciales para la conversión a formato ADRS y posterior a eso la obtención de los periodos de las estructuras.

Con todos estos factores ya ingresados se procede a la conversión de las curvas de capacidad junto con su representación Bilineal en formato ADRS.

```
Ingrese la posición del punto Dui-Aui para la curva pushover: 10
Ingrese las coordenadas iniciales del punto A para la curva pushover (Dyi, Ayi):
2,4
Área bajo la curva Pushover: 28.44
Área bajo la curva de la Representación bilineal inicial: 29.10
Error de la representación bilineal inicial: 2.30%
¿Calcular nuevas coordenadas para el punto A? (si/no): si
Nuevas coordenadas del punto A: Dyi = 1.92, Ayi = 3.84
Área bajo la curva de la Representación bilineal con el punto A encontrado: 28.68
Error de la representación bilineal: 0.82%
Rigidez lateral efectiva Ke: 2.00 Tonf/cm
Rigidez inicial Ki: 1.68 Tonf/cm
Ingresar Peso Efectivo sísmico W (Ton): 19
¿Conoce los valores de factor de participación modal ( $\alpha$ ) y factor de forma de la
estructura (Pf)? (si/no): si
Ingrese factor de participación modal ( $\alpha$ ): 1
Ingrese factor de forma de la estructura (Pf): 1
Factor de participación modal ( $\alpha$ ): 1.0
Factor de forma de la estructura (Pf): 1.0
```

Figura 28. Análisis para la obtención de la Representación Bilineal de la curva de Capacidad 1. Fuente: Elaboración Propia.

```
Ingrese la posición del punto Dui-Aui para la curva pushover: 10
Ingrese las coordenadas iniciales del punto A para la curva pushover (Dyi, Ayi):
2,4
Área bajo la curva Pushover: 32.89
Área bajo la curva de la Representación bilineal inicial: 33.78
Error de la representación bilineal inicial: 2.72%
¿Calcular nuevas coordenadas para el punto A? (si/no): si
Nuevas coordenadas del punto A: Dyi = 1.90, Ayi = 3.80
Área bajo la curva de la Representación bilineal con el punto A encontrado: 33.17
Error de la representación bilineal: 0.86%
Rigidez lateral efectiva Ke: 2.00 Tonf/cm
Rigidez inicial Ki: 1.56 Tonf/cm
Ingresar Peso Efectivo sísmico W (Ton): 23.84
¿Conoce los valores de factor de participación modal ( $\alpha$ ) y factor de forma de la
estructura (Pf)? (si/no): si
Ingrese factor de participación modal ( $\alpha$ ): 1
Ingrese factor de forma de la estructura (Pf): 1
Factor de participación modal ( $\alpha$ ): 1.0
Factor de forma de la estructura (Pf): 1.0
```

Figura 29. Análisis para la obtención de la Representación Bilineal de la curva de Capacidad 2. Fuente: Elaboración Propia.

3.1.3 Representación Bilineal de la Curva de capacidad formato ADRS

En la conversión a formato ADRS podemos observar las diferencias que existen entre ambas curvas de capacidad, para la primera curva se observa que tiene un mayor desarrollo con relación a la aceleración espectral S_a , mientras que la otra curva de capacidad tiene una mayor capacidad con relación al Desplazamiento Espectral.

Para cada curva de capacidad, independientemente a la presentadas, junto con sus respectivos valores solicitados anteriormente, se mostrará su respectiva representación en formato ADRS.

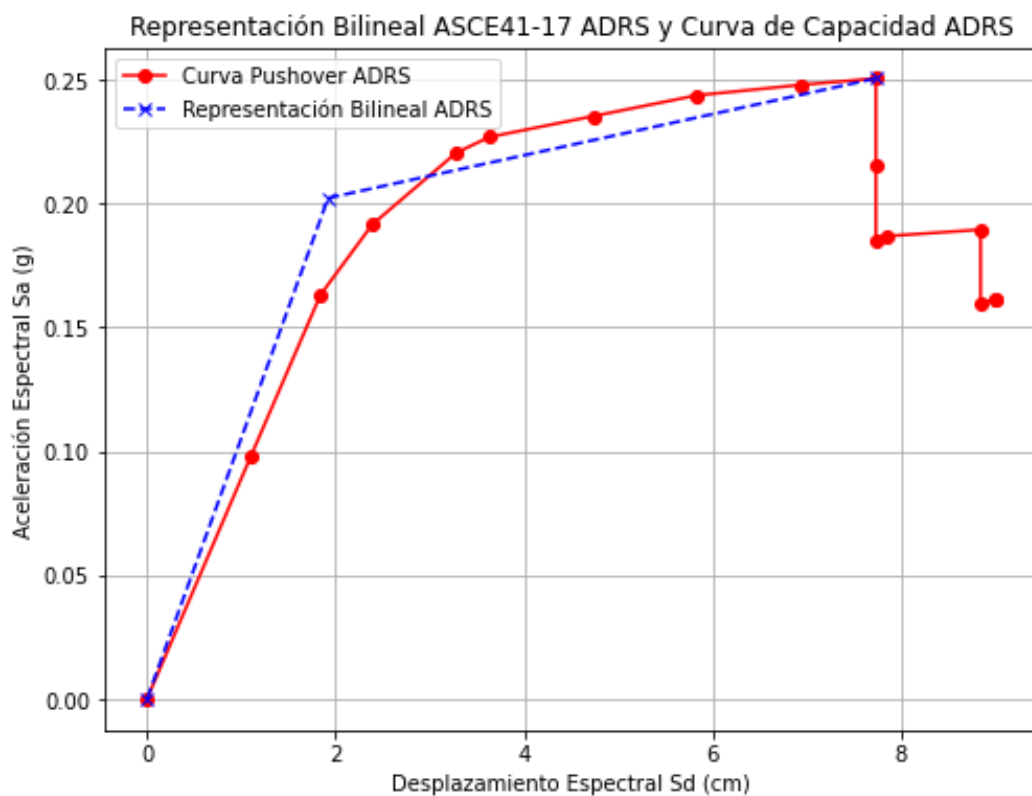


Figura 30. Curvas de Capacidad 1 junto con su Representación Bilineal en formato ADRS. Fuente: Elaboración Propia.

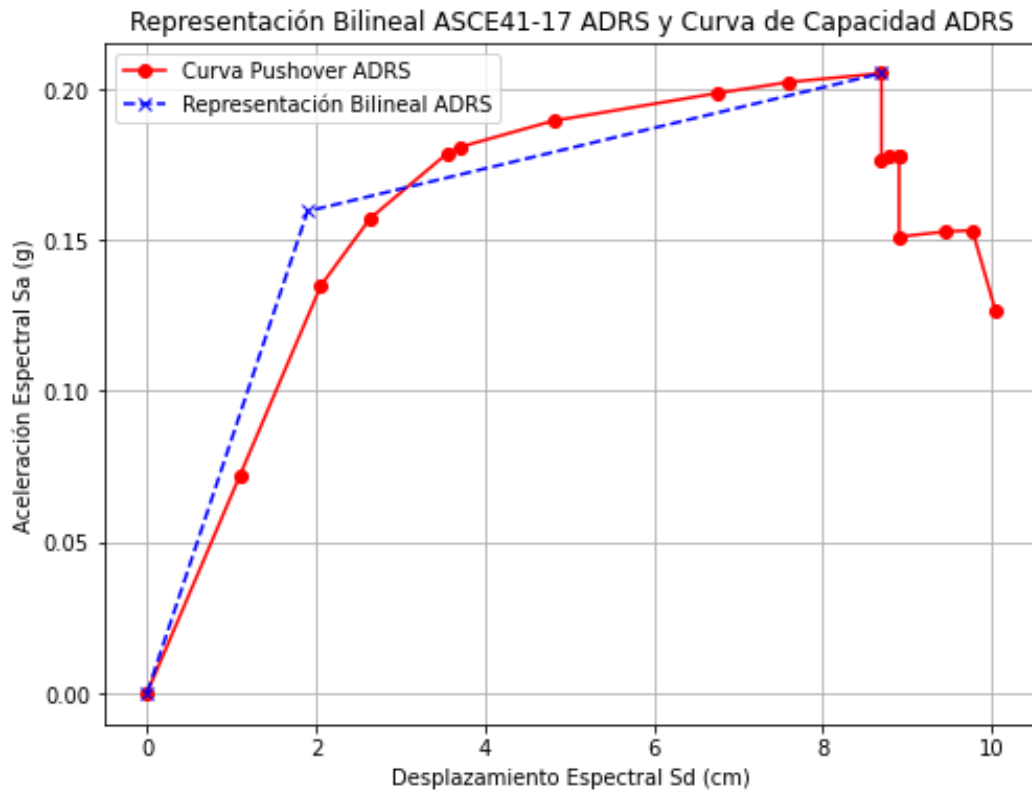


Figura 31. Curvas de Capacidad 2 junto con su Representación Bilineal en formato ADRS. Fuente: Elaboración Propia.

3.1.4 Análisis para obtención de puntos de desempeño Curva de Capacidad ASCE41-17

Una vez obtenida la representación Bilineal junto con la curva de capacidad en formato ADRS, el código presentará el periodo fundamental elástico T_i , el cual es necesario para poder calcular el Periodo fundamental efectivo, con este se procede a encontrar los desplazamientos objetivos para cada registro sísmico, es decir el punto de desempeño para cada sismo analizado.

Dado que para cada curva de capacidad presentado en este proyecto es una estructura de hormigón armado, ubicado en Guayaquil, suelo tipo D y de dos plantas; algunos valores coinciden, la diferencia principal radica en los periodos tanto el inicial como el efectivo.

Período fundamental elástico T_i : 0.69 segundos
Ingrese el número de pisos del edificio: 2
Valor de C_o : 1.2
Valor de C_m : 1
¿Conoce el tipo de Suelo? (si/no): si
Ingrese Tipo de Suelo (A, B, C, D, E, F): D
Valor de a : 60
Período fundamental efectivo T_e : 0.61 segundos
¿Desea ver la gráfica de puntos de desempeño correspondiente a cada sismo? (si/no):
si

Figura 32 Coeficientes para la obtención de puntos de desempeño según el método ASCE 41-17 para la curva de capacidad 1. Fuente: Elaboración propia.

Período fundamental elástico T_i : 0.62 segundos
Ingrese el número de pisos del edificio: 2
Valor de C_o : 1.2
Valor de C_m : 1
¿Conoce el tipo de Suelo? (si/no): si
Ingrese Tipo de Suelo (A, B, C, D, E, F): D
Valor de a : 60
Período fundamental efectivo T_e : 0.57 segundos
¿Desea ver la gráfica de puntos de desempeño correspondiente a cada sismo? (si/no):
si

Figura 33 Coeficientes para la obtención de puntos de desempeño según el método ASCE 41-17 para la curva de capacidad 2. Fuente: Elaboración propia.

3.1.5 Puntos de desempeño ASCE 41-17 para Curva de Capacidad

Se presentan los resultados de tres registros sísmicos escogidos al azar, dado que los que se analizaron fueron 120, no queremos saturar de imágenes de todos los puntos de desempeño.

Observamos la diferencia de puntos de desempeño debido al cambio de periodo de ambas estructuras, aunque en algunos casos la diferencia no se tan marcada, existe y es válido conocerla, ya que con esto podemos clasificar a la estructura dentro de un desempeño sísmico.

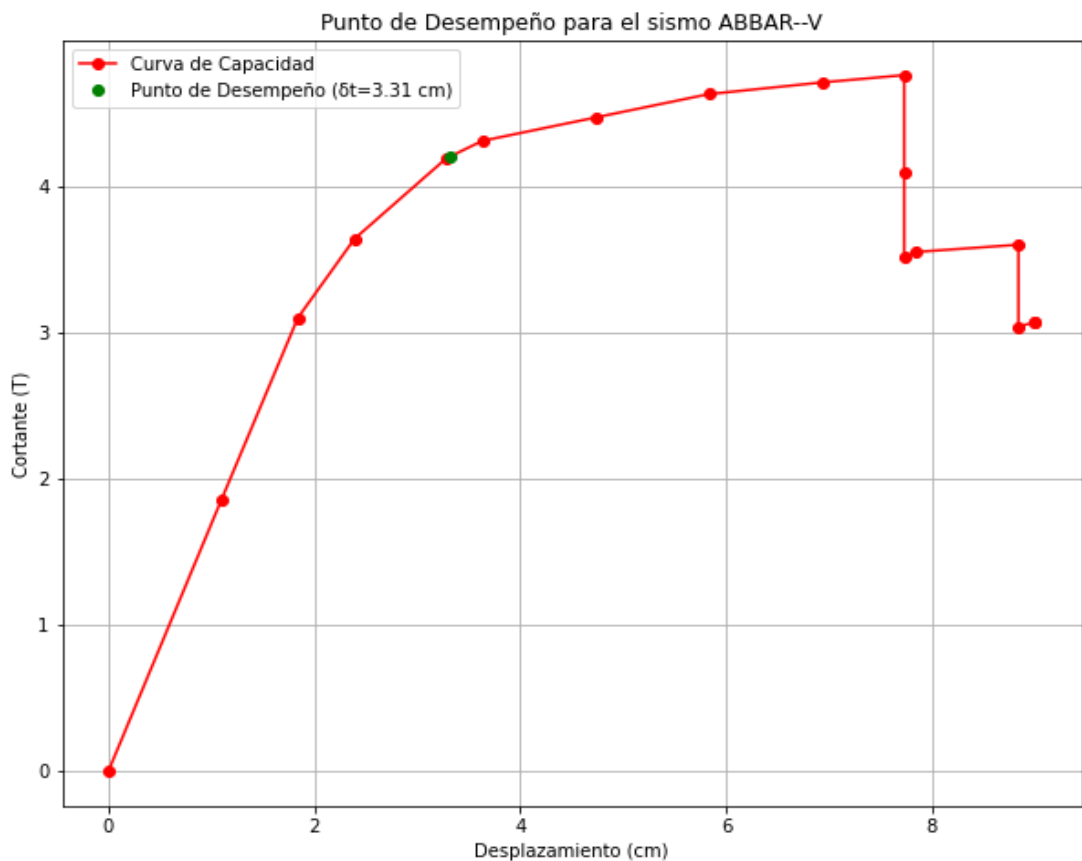


Figura 34. Desplazamiento objetivo (δ_t) del sismo ABBAR--V para Curva de Capacidad 1. Fuente: Elaboración Propia

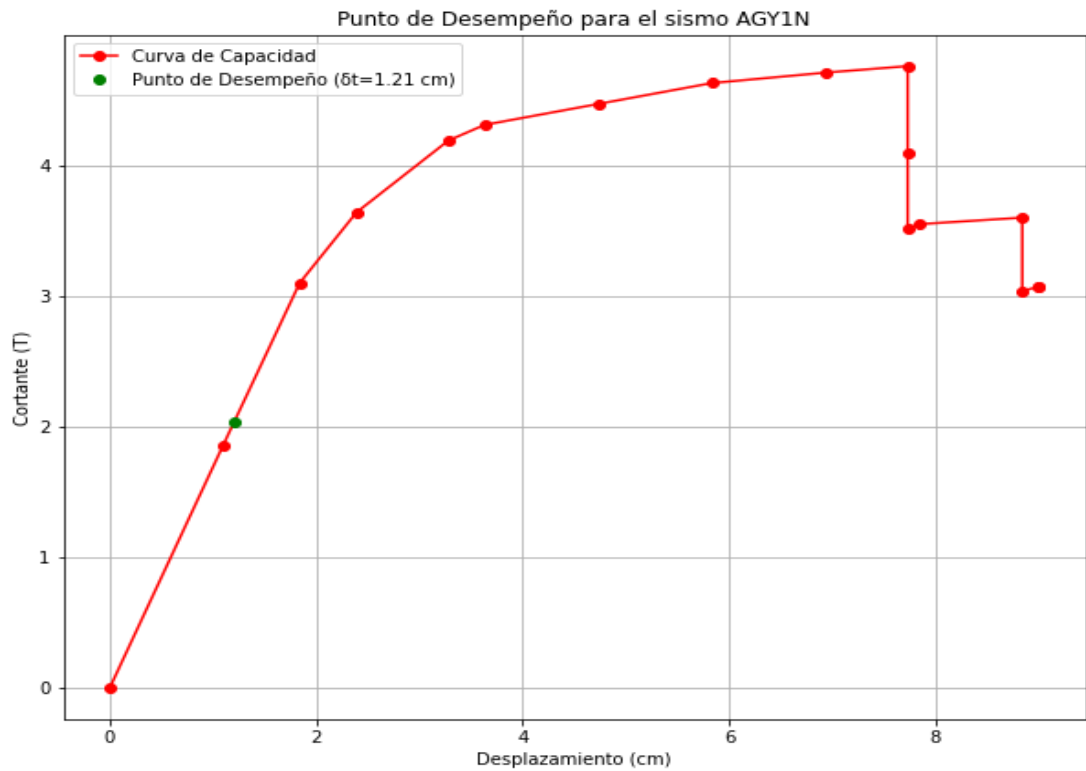


Figura 35. Desplazamiento objetivo (δ_t) del sismo AGY1N para Curva de Capacidad 1. Fuente: Elaboración Propia

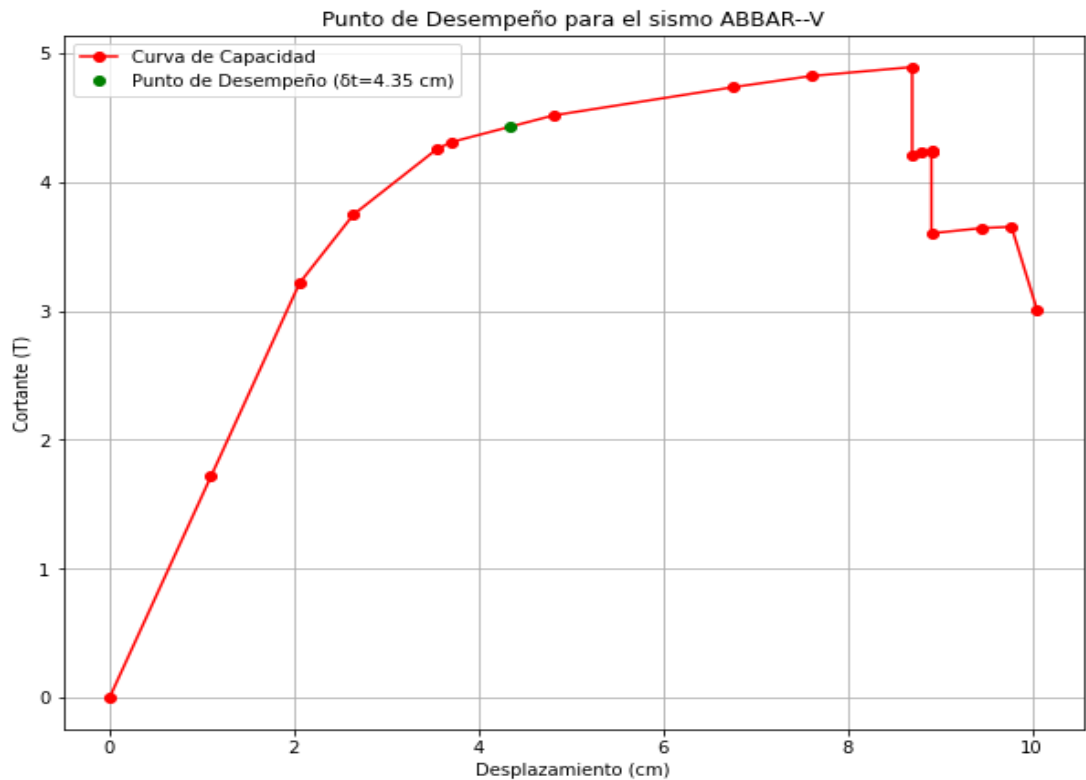


Figura 36. Desplazamiento objetivo (δ_t) del sismo APO1N para Curva de Capacidad 1. Fuente: Elaboración Propia

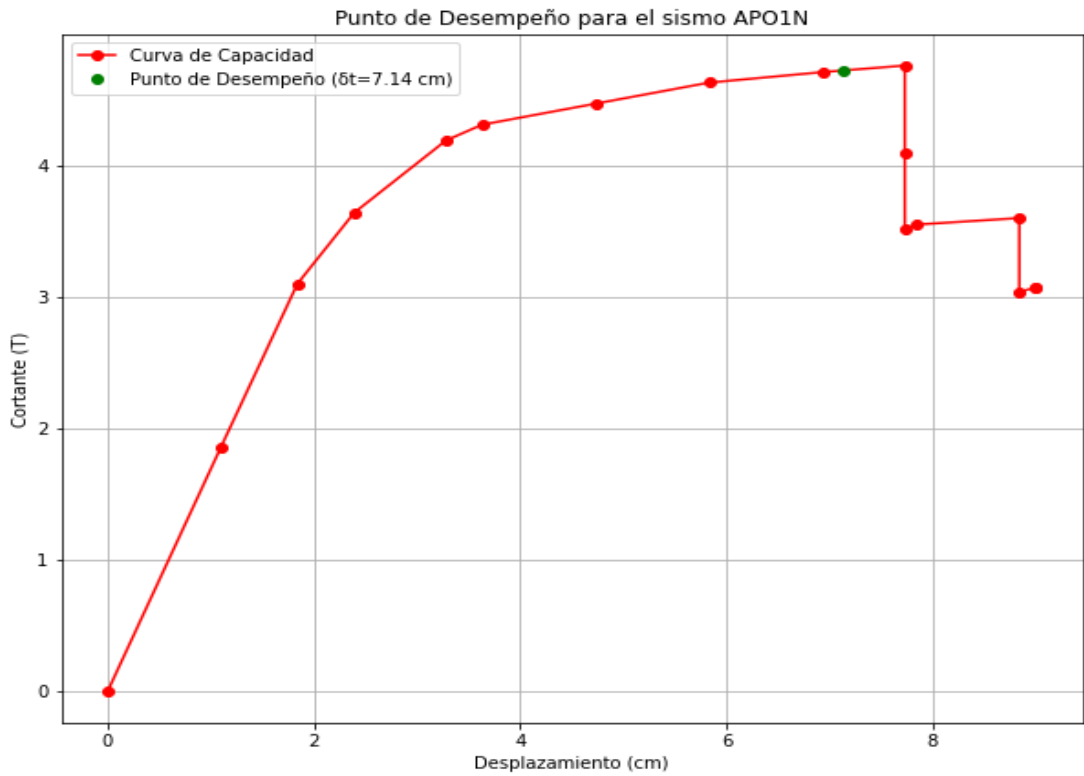


Figura 37. Desplazamiento objetivo (δ_t) del sismo ABBAR--V para Curva de Capacidad 2. Fuente: Elaboración Propia

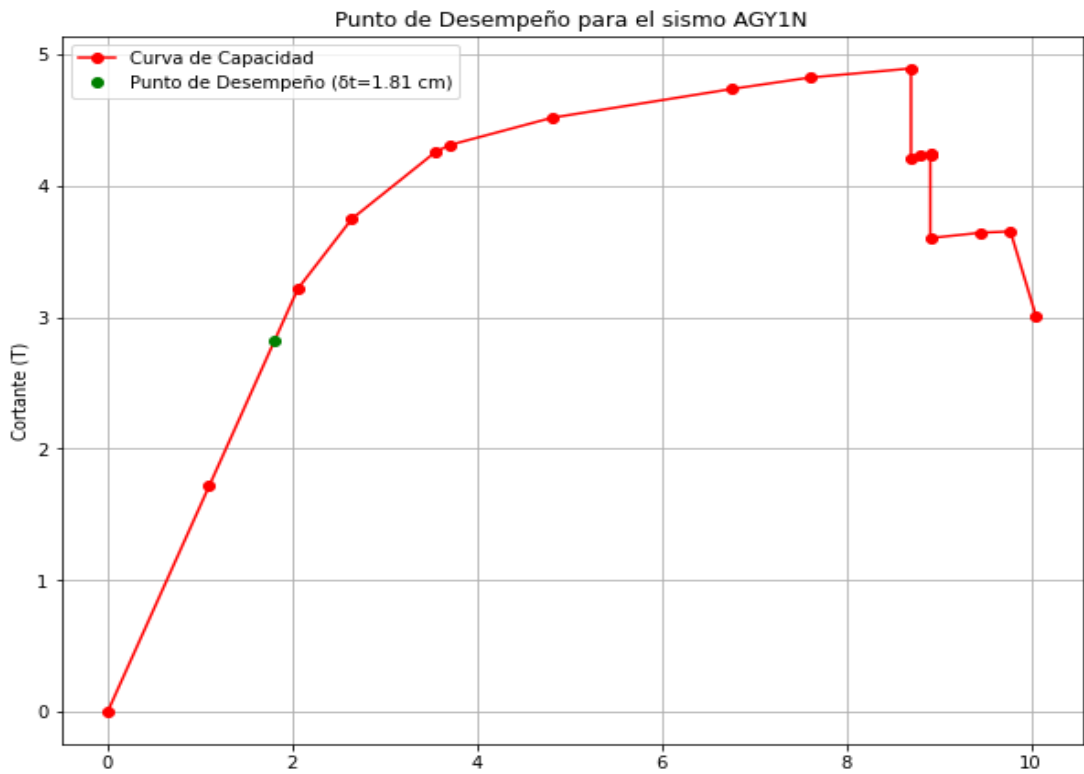


Figura 38. Desplazamiento objetivo (δ_t) del sismo AGY1N para Curva de Capacidad 2. Fuente: Elaboración Propia

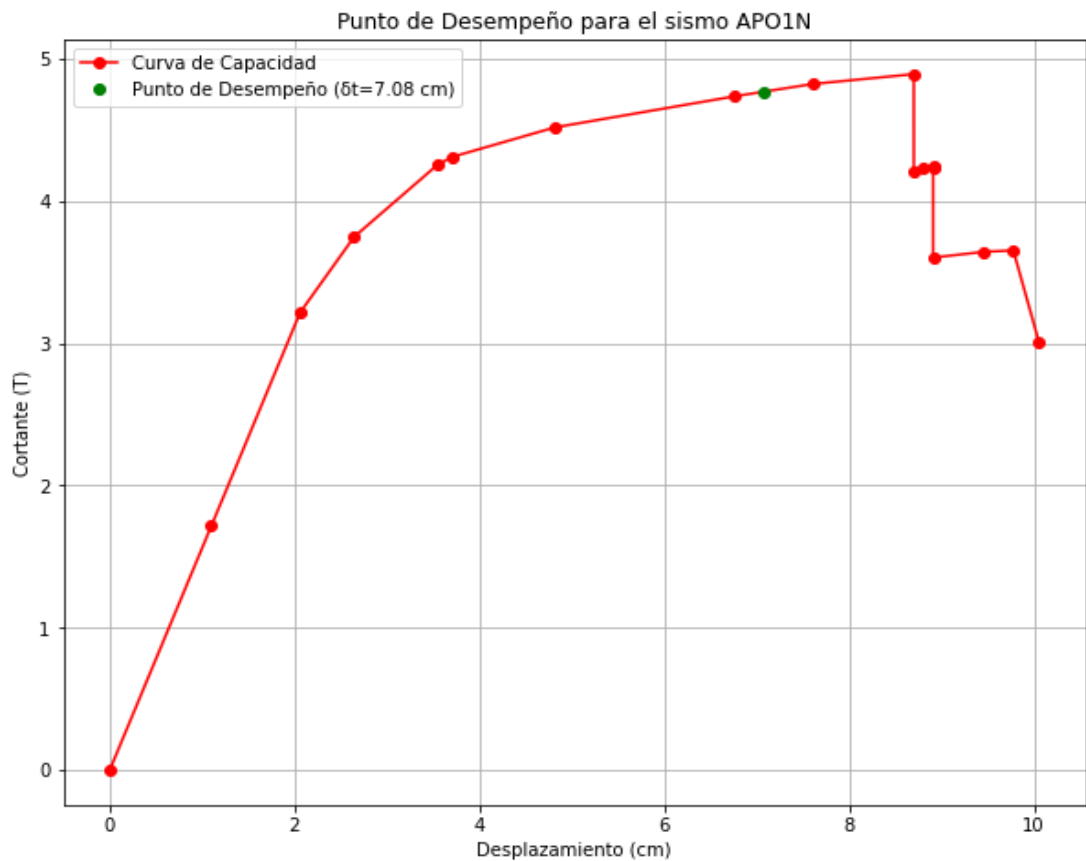


Figura 39. Desplazamiento objetivo (δ_t) del sismo APO1N para Curva de Capacidad 2. Fuente: Elaboración Propia.

3.1.6 Desplazamiento Espectral S_d (cm) vs Desplazamiento Objetivo δ_t Curva de capacidad y Análisis para curva de Frágil de la Curva de capacidad ASCE41-17

Como se observa en la Figura 40 y Figura 42 una vez se obtiene el punto de desempeño por medio del método ASCE41-17 observamos que existe una tendencia lineal, sin importa la curva de capacidad analizada, esta tendencia se produce dado por la relación que existe en el Desplazamiento espectral junto con los respectivos coeficientes necesarios para el cálculo del punto de desempeño, la tendencia que tendrá en este tipo de gráfica siempre será alcista.

Por otro lado, en la Figura 41 y Figura 43, se observa los valores de excedencia para cada caso de daño, así como la clasificación de los intervalos

y el valor promedio de estos, para la posterior obtención de curvas de fragilidad.

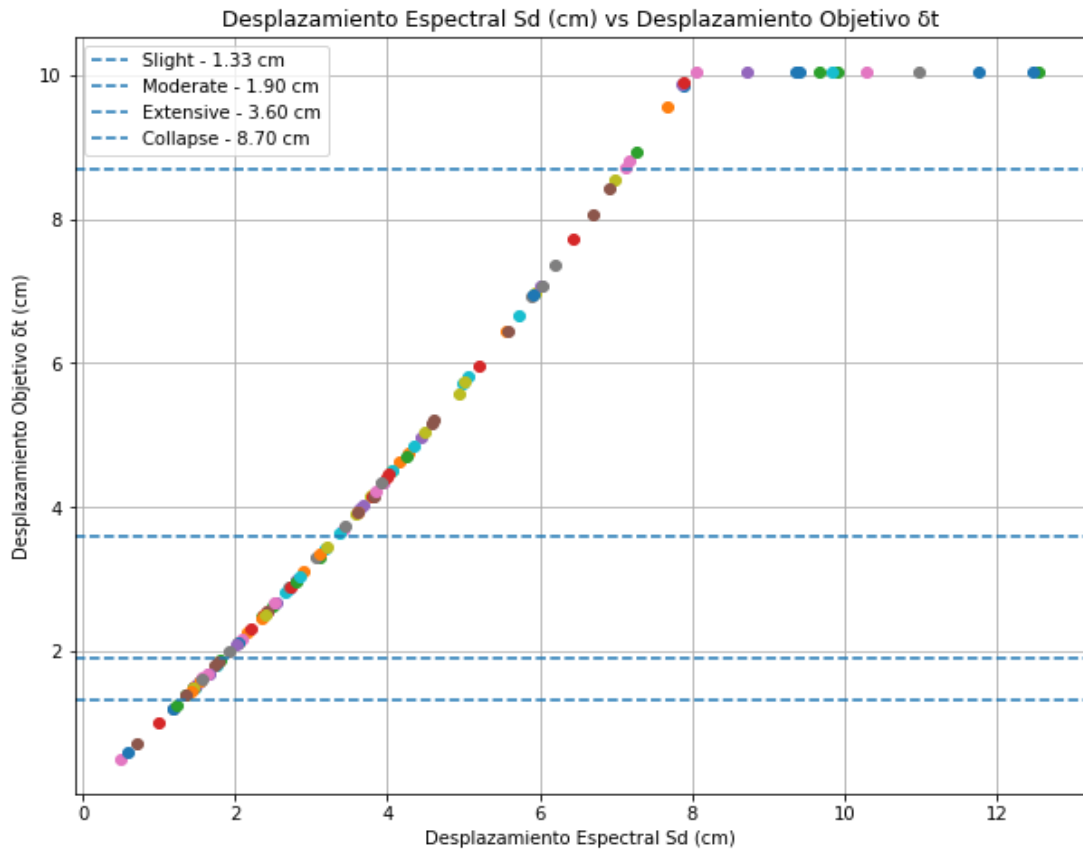


Figura 40. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 1. Fuente: Elaboración propia.

Daño Leve: 19
Daño Moderado: 29
Daño Extenso: 46
Daño Completo: 19
Insertar valor de intervalo: 2
Inserte último valor de análisis: 10
Promedio Intervalo 1 = 0.0140 m
Promedio Intervalo 2 = 0.0301 m
Promedio Intervalo 3 = 0.0492 m
Promedio Intervalo 4 = 0.0709 m
Promedio Intervalo 5 = 0.0927 m

Figura 41. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 1. Fuente: Elaboración Propia.

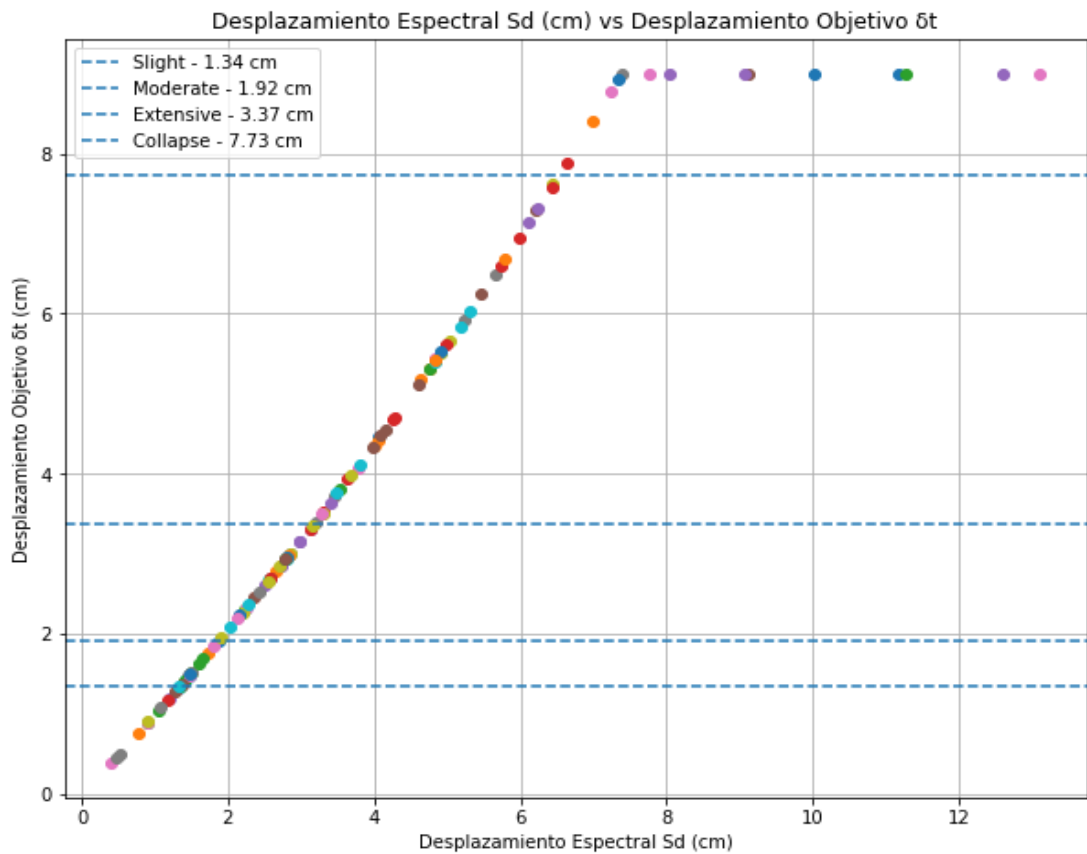


Figura 42. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 2. Fuente: Elaboración propia.

```

Daño Leve: 17
Daño Moderado: 32
Daño Extenso: 44
Daño Completo: 14
Insertar valor de intervalo: 2
Inserte último valor de análisis: 10
Promedio Intervalo 1 = 0.0129 m
Promedio Intervalo 2 = 0.0290 m
Promedio Intervalo 3 = 0.0486 m
Promedio Intervalo 4 = 0.0679 m
Promedio Intervalo 5 = 0.0874 m

```

Figura 43. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 2. Fuente: Elaboración Propia.

3.1.15 Curvas de Fragilidad ASCE41-17

Dado que el código fue hecho para la obtención de curvas de fragilidad en base al desplazamiento espectral S_d , podemos apreciar que no existe una gran diferencia entre ambas curvas de fragilidad obtenidas por el método de coeficientes propuesto por el ASCE41-17. Aparentemente con el bajo porcentaje de ductilidad que se agregó a la estructura de la curva de capacidad 1, esta no se diferencia tanto del análisis de la curva de capacidad 2.

También nos damos cuenta de que el método de coeficientes propuesto por el ASCE 41-17, no permite realizar una correcta probabilidad estadística, dado a la tendencia lineal que se presentó anteriormente, por esto, se decidió realizar el método propuesto por el FEMA 440 para no obtener linealidad y obtener curvas de fragilidad con una mejor probabilidad estadística.

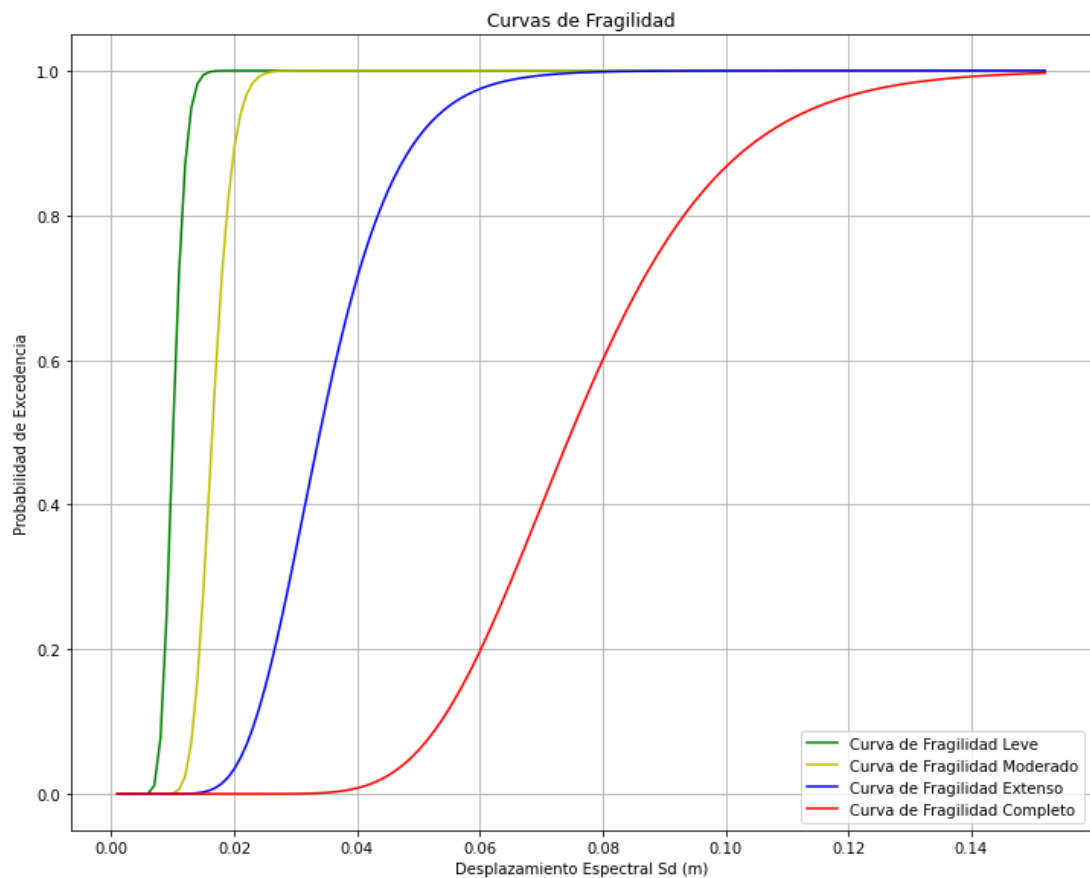


Figura 44. Curva de Fragilidad obtenida para la Curva de Capacidad 1. Fuente: Elaboración Propia.

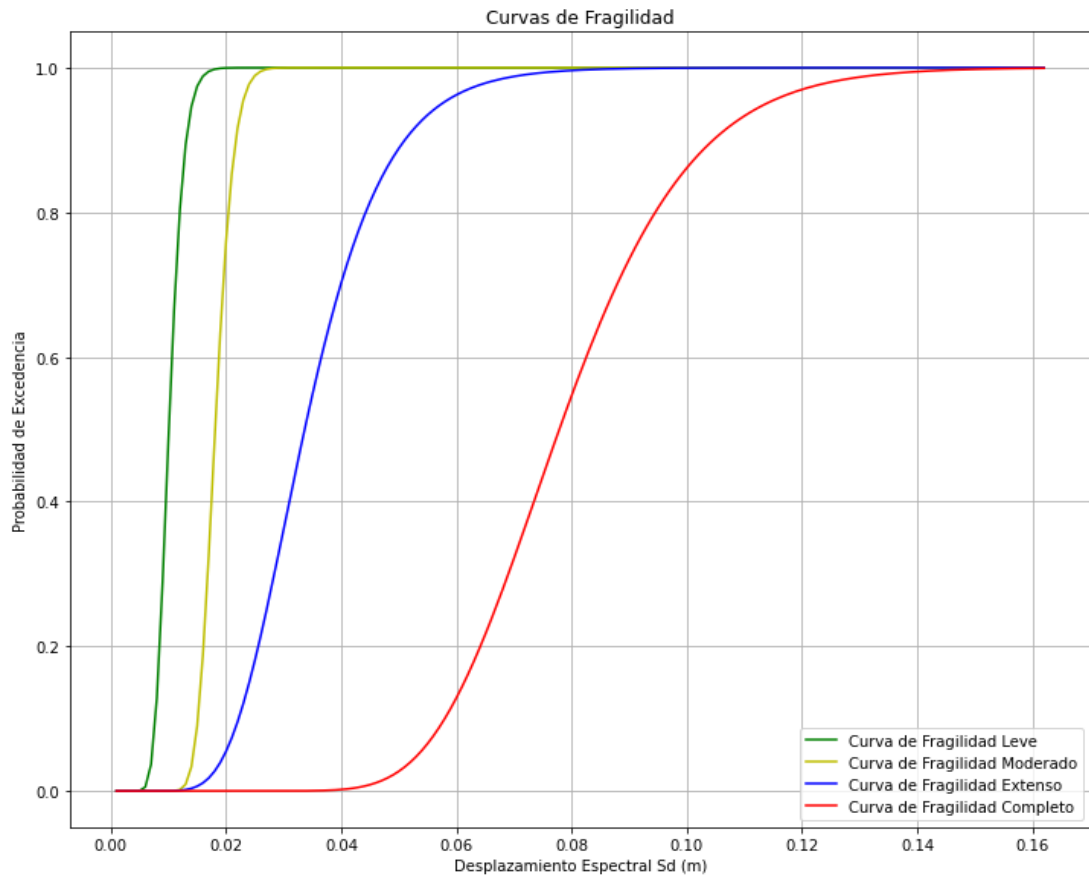


Figura 45. Curva de Fragilidad obtenida para la Curva de Capacidad 2.

Fuente: Elaboración Propia.

3.2 Resultados por FEMA440 método C

En este apartado se usará los mismos registros sísmicos del caso anterior para ambas Curvas de Capacidad, además de presentar los resultados correspondientes para cada análisis de curva de capacidad, no se presentarán las gráficas, ya que se las presentaron en la sección 3.1.1 y sección 0.

Además, que su análisis para la obtención de la Representación Bilineal y su respectiva conversión a unidades ADRS es la misma para ambas Curvas revisar secciones: 0, 0, 3.1.4, 0; en esta sección se enfocará directamente en los puntos de desempeño por el método correspondiente.

3.2.1 Análisis para obtención de puntos de desempeño Curva de Capacidad FEMA440 Método C

Rigidez post-elástica: 0.08					
Ductilidad última esperada (μ): 4.02					
Periodo Inicial (T_0): 0.62					
	Ductilidad μ	Periodo Efectivo T_{eff}	...	Au	Reduction(M/B)
0	1.000000	0.618311	...	0.202231	0.997641
1	1.500000	0.646290	...	0.210217	0.718093
2	2.000000	0.718478	...	0.218204	0.623777
3	2.500000	0.817253	...	0.226190	0.603668
4	3.000000	0.924994	...	0.234177	0.613332
5	3.500000	1.024078	...	0.242164	0.630128
6	4.000000	1.032580	...	0.250150	0.561935
7	4.023544	1.034472	...	0.250526	0.561458

Figura 46. Reducción del espectro de respuesta para cada caso de ductilidad de la Curva de Capacidad 1. Fuente: Elaboración Propia.

Rigidez post-elástica: 0.08					
Ductilidad última esperada (μ): 4.57					
Periodo Inicial (T_0): 0.69					
	Ductilidad μ	Periodo Efectivo T_{eff}	...	Au	Reduction(M/B)
0	1.000000	0.692602	...	0.159562	0.997641
1	1.500000	0.723942	...	0.165950	0.718466
2	2.000000	0.804803	...	0.172338	0.624402
3	2.500000	0.915446	...	0.178725	0.604543
4	3.000000	1.036132	...	0.185113	0.614478
5	3.500000	1.147121	...	0.191501	0.631551
6	4.000000	1.156645	...	0.197889	0.563408
7	4.500000	1.201664	...	0.204276	0.556290
8	4.573392	1.208272	...	0.205214	0.555690

Figura 47. Reducción del espectro de respuesta para cada caso de ductilidad de la Curva de Capacidad 2. Fuente: Elaboración Propia

3.2.2 Puntos de desempeño FEMA440 Método C para Curva de Capacidad

Al igual que en la sección 3.1.5, acerca de los puntos de desempeño según el ASCE 41-17, en este apartado se siguieron los pasos propuestos por el FEMA440 Método C sección 1.6.1.3, donde, se reduce el espectro a diferentes valores de ductilidades como se muestra en la Figura 46 y Figura 47, que van desde el 1 hasta el último valor de ductilidad que alcanza la estructura, obteniendo el porcentaje de reducción de cada espectro.

A continuación, se presenta gráficamente los puntos de desempeño para diferentes registros sísmicos:

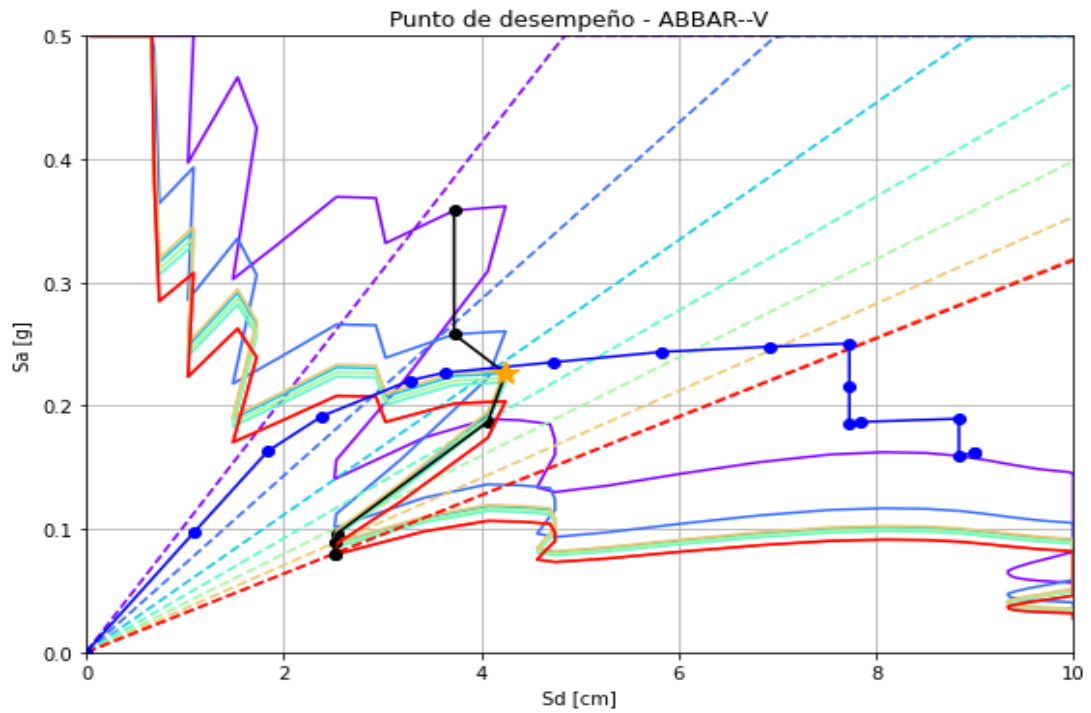


Figura 48. Punto de desempeño del sismo ABBAR--V para Curva de Capacidad 1. Fuente: Elaboración Propia

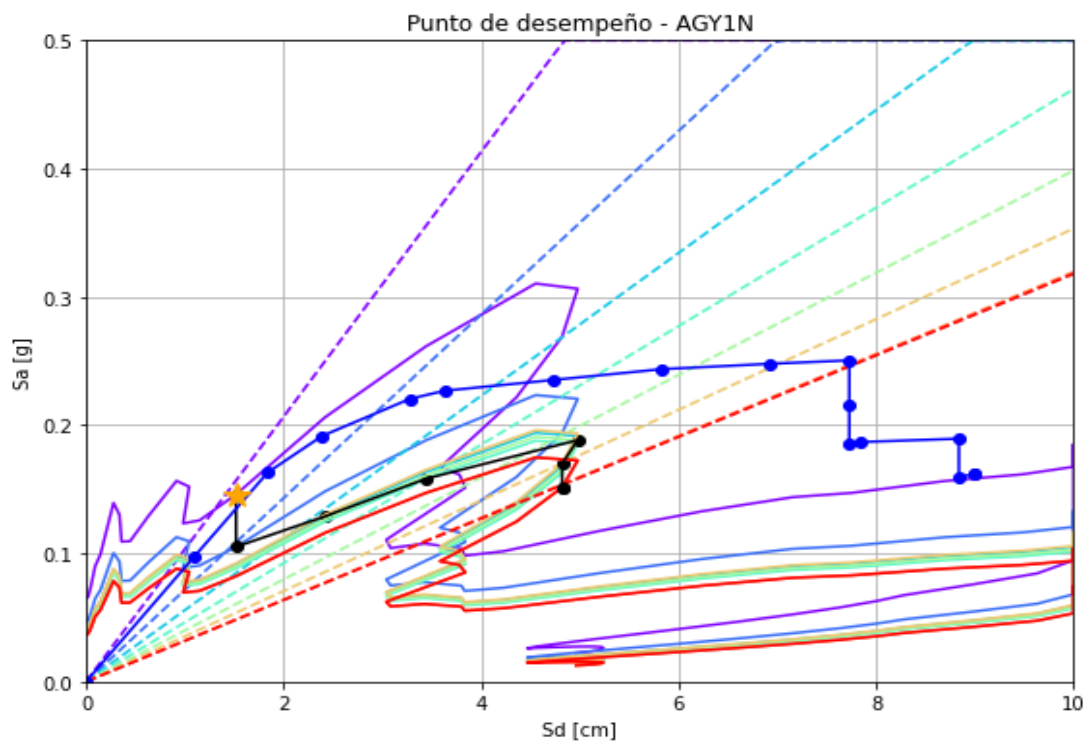


Figura 49. Punto de desempeño del sismo AGY1N para Curva de Capacidad 1. Fuente: Elaboración Propia

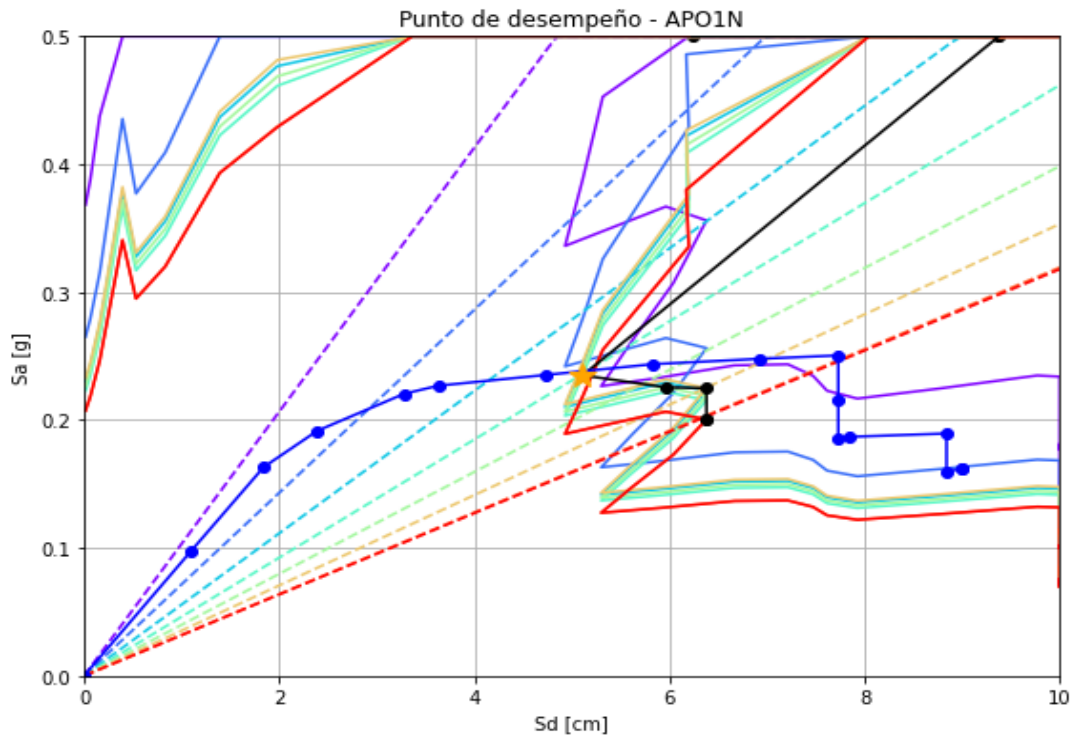


Figura 50. Punto de desempeño del sismo APO1N para Curva de Capacidad
1. Fuente: Elaboración Propia

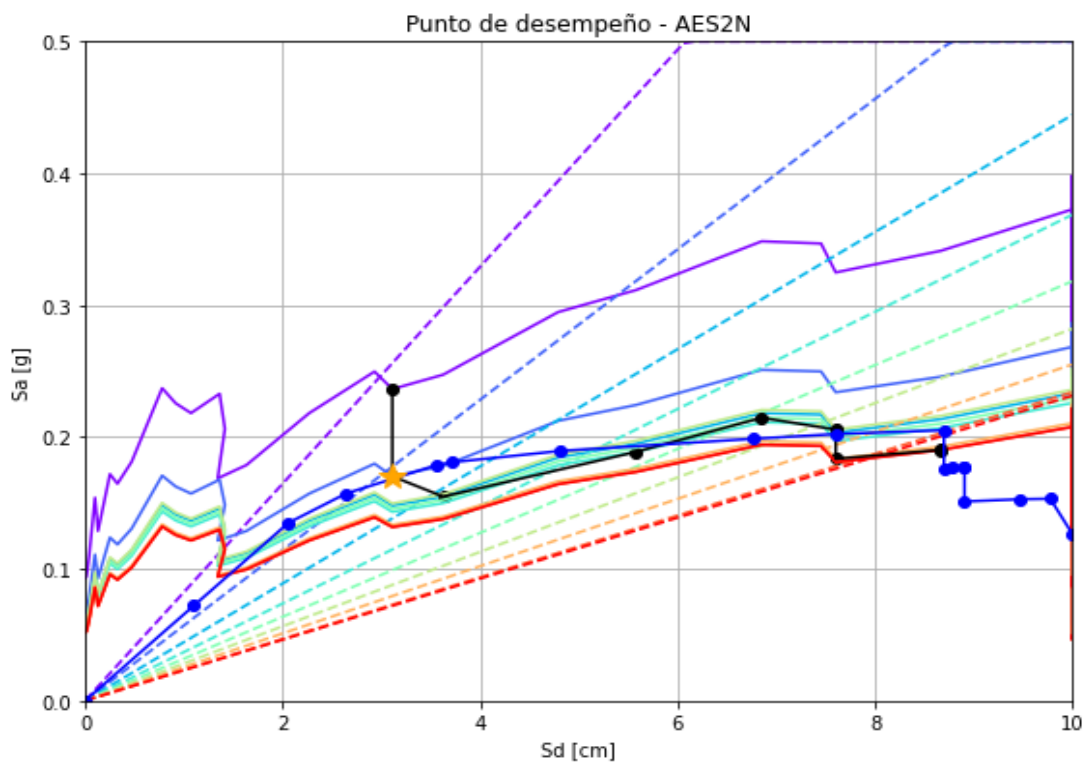


Figura 51. Punto de desempeño del sismo AES2N para Curva de Capacidad
2. Fuente: Elaboración Propia

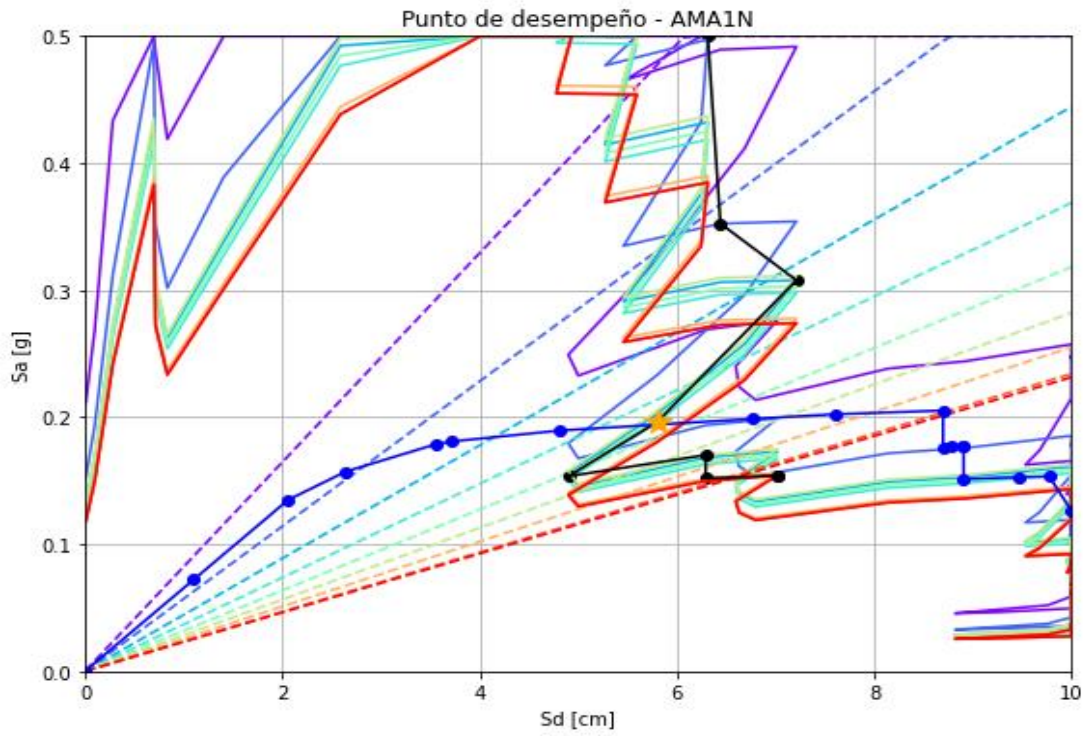


Figura 52. Punto de desempeño del sismo AMA1N para Curva de Capacidad 2.

Fuente: Elaboración Propia

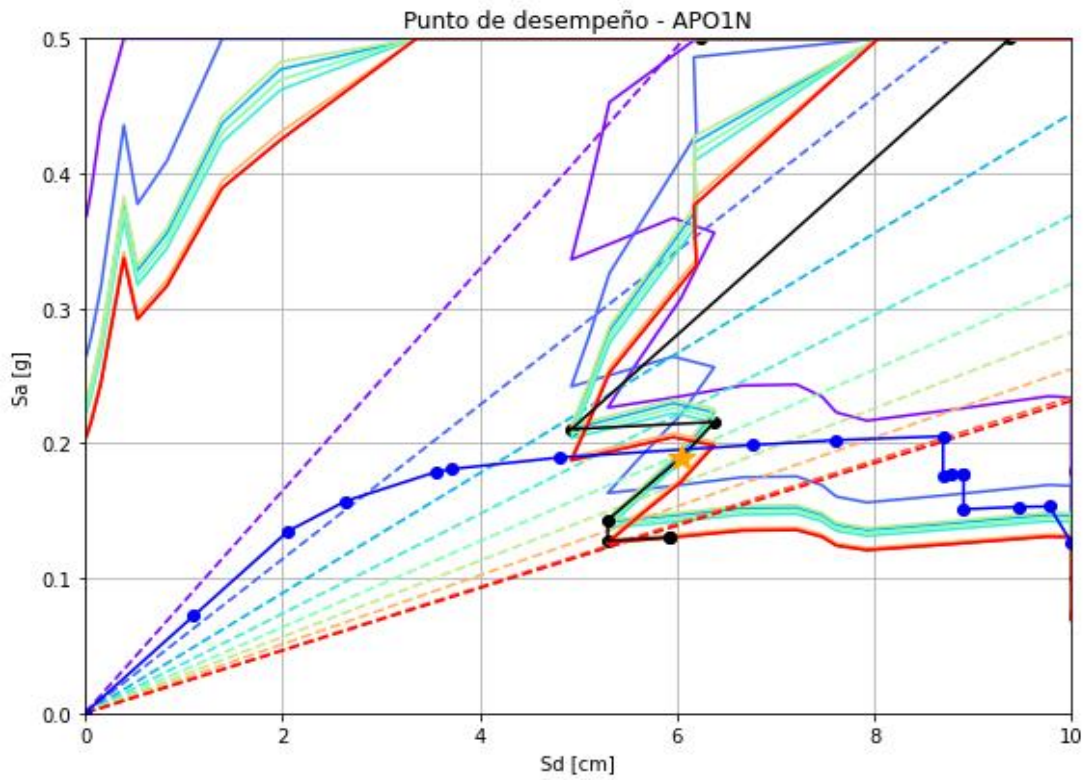


Figura 53. Punto de desempeño del sismo APO1N para Curva de Capacidad 2.

Fuente: Elaboración Propia

3.2.3 Desplazamiento Espectral Sd (cm) vs Sd Performance Point (cm) Curva de Capacidad FEMA440 Método C

Observamos que los umbrales de daño son los mismo para cada curva de capacidad respectivamente comparándoles con el método de coeficientes del ASCE 41-17, la mayor diferencia apreciada es la tendencia que tiene los puntos graficados, dejó de ser lineal, y existe una mayor dispersión, causando una mayor aproximación a la realidad y una probabilidad estadística más real. La Figura 55 y Figura 57 es prácticamente lo dicho en el método anterior.

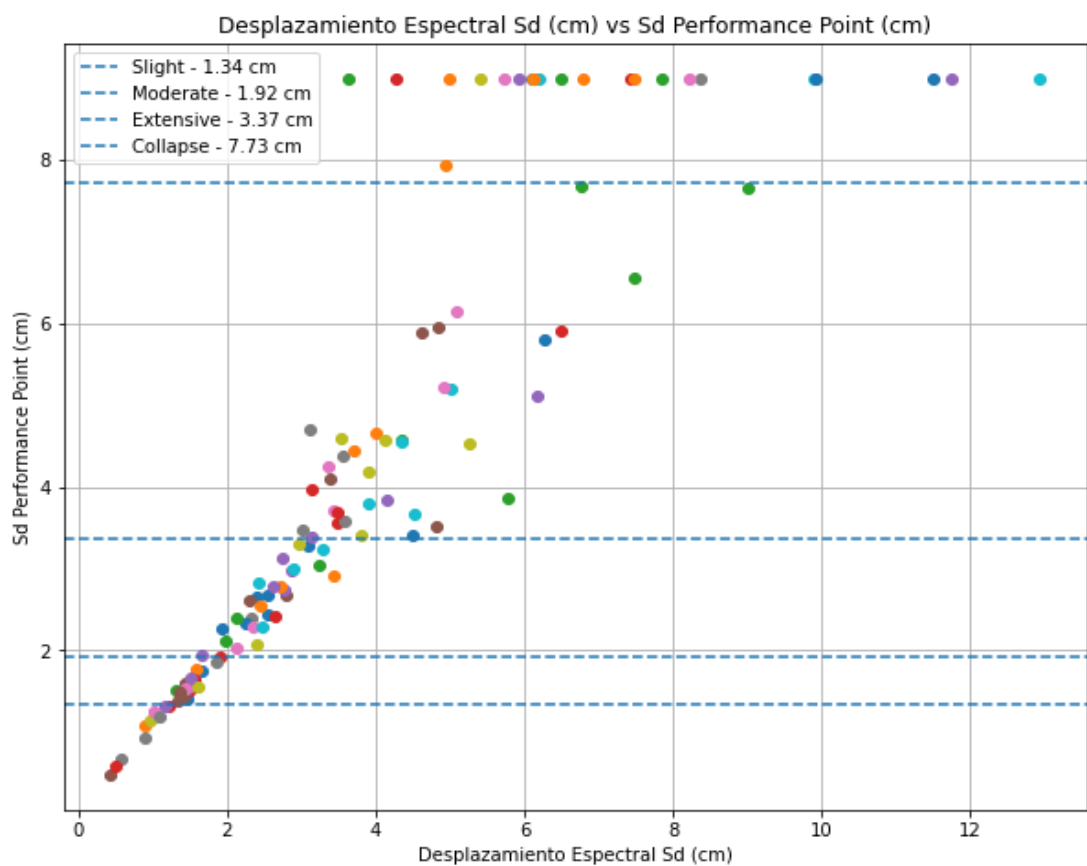


Figura 54. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 1. Fuente: Elaboración propia.

Daño Leve: 19
 Daño Moderado: 29
 Daño Extenso: 37
 Daño Completo: 23
 Insertar valor de intervalo: 2
 Inserte último valor de análisis: 10
 Promedio Intervalo 1 = 0.0135 m
 Promedio Intervalo 2 = 0.0298 m
 Promedio Intervalo 3 = 0.0484 m
 Promedio Intervalo 4 = 0.0670 m
 Promedio Intervalo 5 = 0.0908 m

Figura 55. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 1. Fuente: Elaboración Propia.

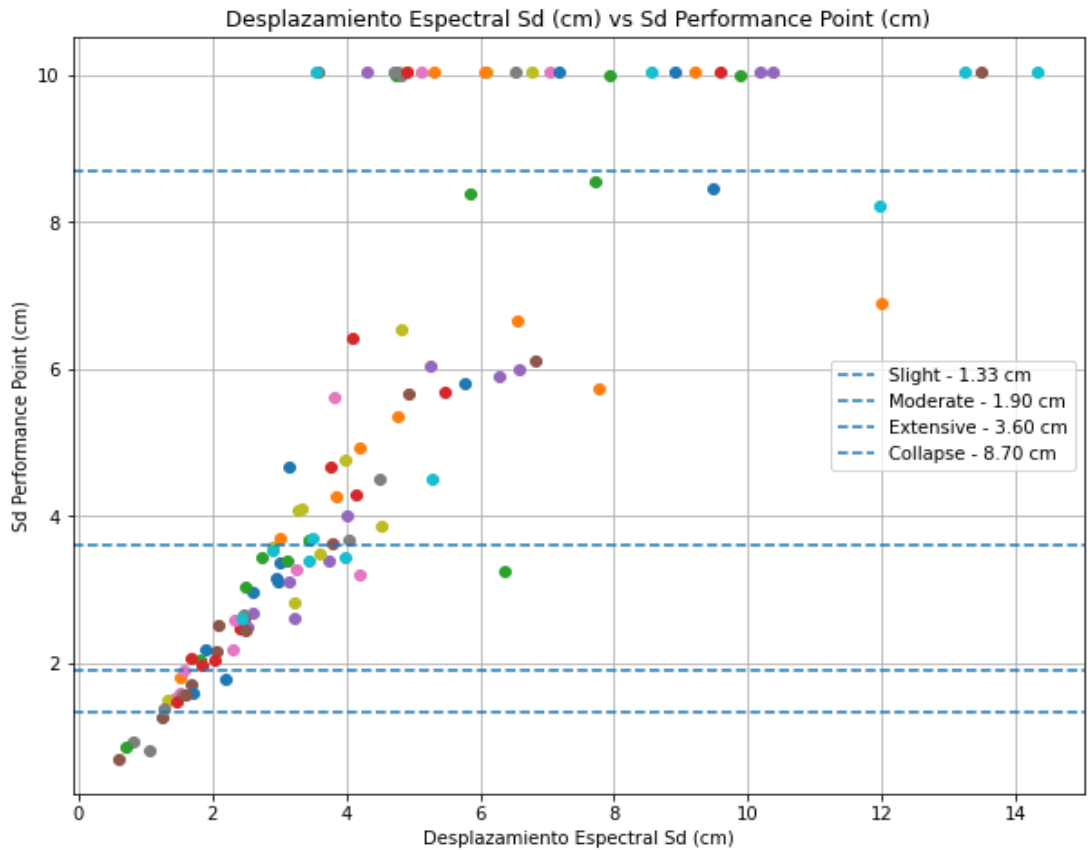


Figura 56. Gráfica para la obtención de curvas de Fragilidad de la Curva de Capacidad 2. Fuente: Elaboración propia.

Daño Leve: 13
Daño Moderado: 35
Daño Extenso: 35
Daño Completo: 28
Insertar valor de intervalo: 2
Inserte último valor de análisis: 10
Promedio Intervalo 1 = 0.0142 m
Promedio Intervalo 2 = 0.0302 m
Promedio Intervalo 3 = 0.0478 m
Promedio Intervalo 4 = 0.0683 m
Promedio Intervalo 5 = 0.0927 m

Figura 57. Análisis necesario para la aplicación de probabilidades estadísticas de la Curva de Capacidad 2. Fuente: Elaboración Propia.

3.2.4 Curvas de Fragilidad FEMA440 Método C

Para ambas curvas de fragilidad obtenidas por los puntos de desempeño propuestos por el FEMA440, se observa una diferencia un poco más marcada, en comparación con los resultados obtenidos con anterioridad en la sección 3.1.15.

También se aprecia que la aportación de ductilidad a la estructura incrementó un poco el desempeño sísmico de esta. Por ejemplo, para un desplazamiento espectral de 0.06 m para la curva de capacidad 1 se obtiene una probabilidad de excedencia para el daño extenso de un 95% aproximadamente mientras que para la curva de capacidad 2 la probabilidad de excedencia es del 85%

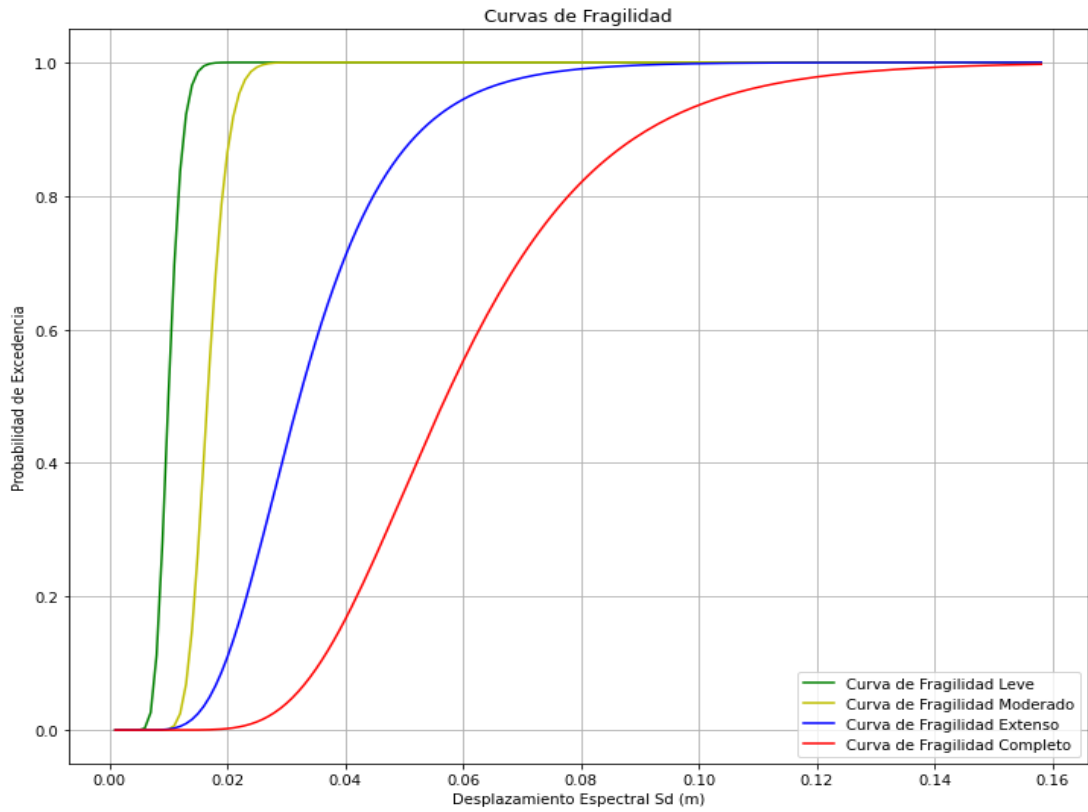


Figura 58. Curva de Fragilidad obtenida para la Curva de Capacidad 1. Fuente: *Elaboración Propia.*

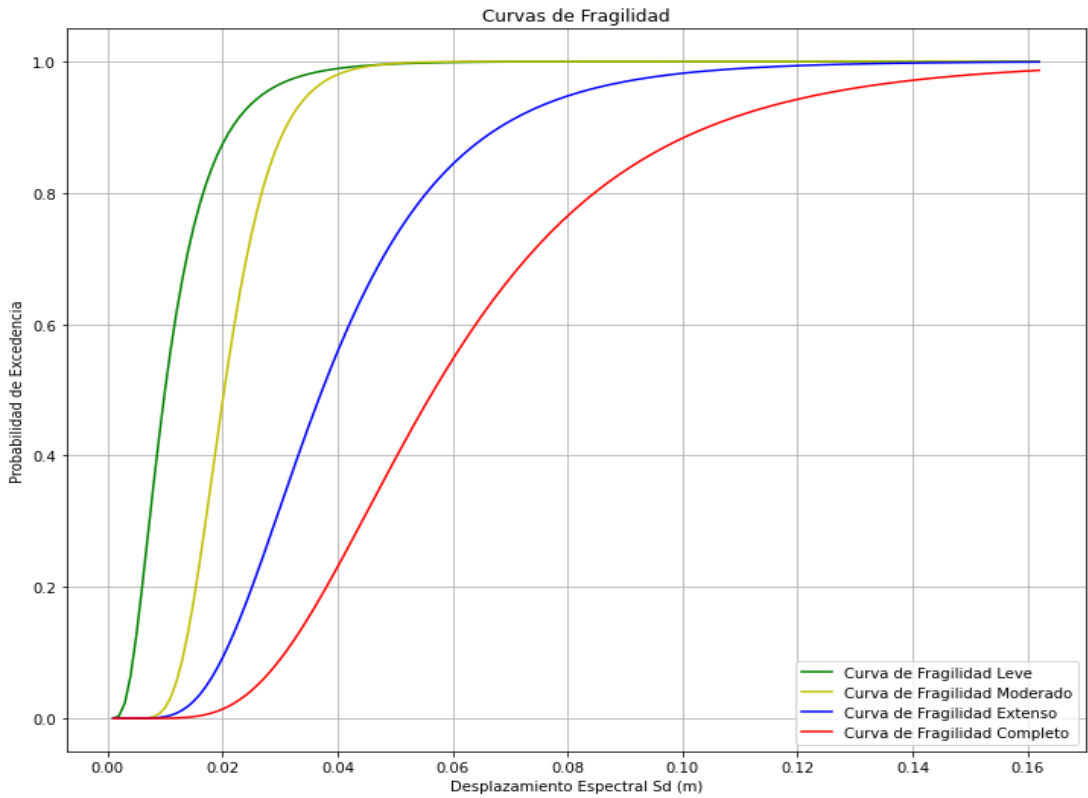


Figura 59. Curva de Fragilidad obtenida para la Curva de Capacidad 2. Fuente: *Elaboración Propia.*

3.3 Comparación de Curvas de fragilidad con estudios anteriores

Para comenzar esta sección, se debe mencionar que las curvas de capacidad utilizadas para la presentación de los resultados fueron obtenidas del Trabajo de Titulación del Ingeniero Cristhian Toro y Raúl Flores con título: Cálculo de curvas de fragilidad de estructuras de hormigón armado de un piso alto con vigas banda y cubierta liviana en residencias ubicadas en vía a la Costa, Guayaquil – Ecuador.

Dicho esto, se procede a comentar las diferencias observadas entre las curvas de fragilidad presentadas en su trabajo y las curvas de fragilidad generadas en el presente estudio a partir de la misma curva de capacidad desarrollada por ellos. Estas diferencias permitirán identificar variaciones en los resultados y comprender mejor cómo los métodos y enfoques utilizados en ambos estudios influyen en la caracterización de la vulnerabilidad sísmica de la estructura. La curva de Fragilidad obtenida para la curva capacidad 1, elaborada en este trabajo de titulación, es la misma que ellos presentan como Curva de Fragilidad Modelo D3-Pushover "x" (sd) Figura 60. Y Curva de Fragilidad obtenida para la Curva de Capacidad 2, es equivalente a Curva de Fragilidad Modelo D11- Pushover "x" Figura 61.

En el desarrollo del presente trabajo de titulación, se ha observado que, a diferencia del estudio anterior, este emplea una mayor cantidad de registros sísmicos, lo que permite obtener un mayor número de puntos de desempeño y, en consecuencia, mejorar la precisión en la elaboración de las Curvas de Fragilidad. Cabe destacar que el objetivo principal del trabajo de titulación anterior fue desarrollar la representación de una curva de capacidad para una estructura, optando por hacerlo mediante pórticos representativos tanto en la dirección X como en la dirección Y. Este procedimiento es válido, siempre y cuando se asegure que la representación refleje adecuadamente el comportamiento global de la estructura, considerando correctamente las rigideces y la determinación de los periodos.

Con lo anterior, se identificó que en el trabajo previo se cometió un error al utilizar el periodo de la estructura completa en lugar del periodo del pórtico analizado para la obtención del gráfico de Desplazamiento Espectral S_d (cm) vs S_d Performance Point (cm) o Desplazamiento Espectral S_d (cm) vs Desplazamiento Objetivo δ_t , secciones 3.2.3 y 3.1.6, respectivamente. A partir de esta observación, se concluye que la comparación de resultados entre ambos trabajos de título sugiere que, al incrementar la ductilidad de una estructura, se logra un mejor desempeño sísmico.

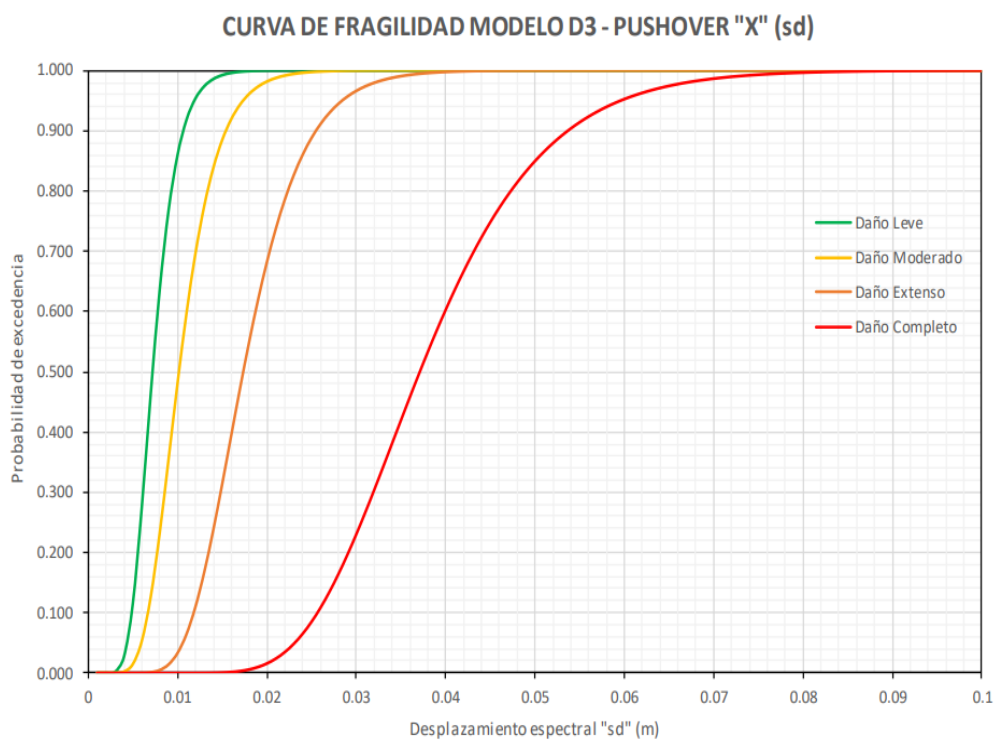


Figura 60 Curva de Fragilidad Modelo D3-Pushover" x" (sd). Fuente: Toro & Flores, 2024

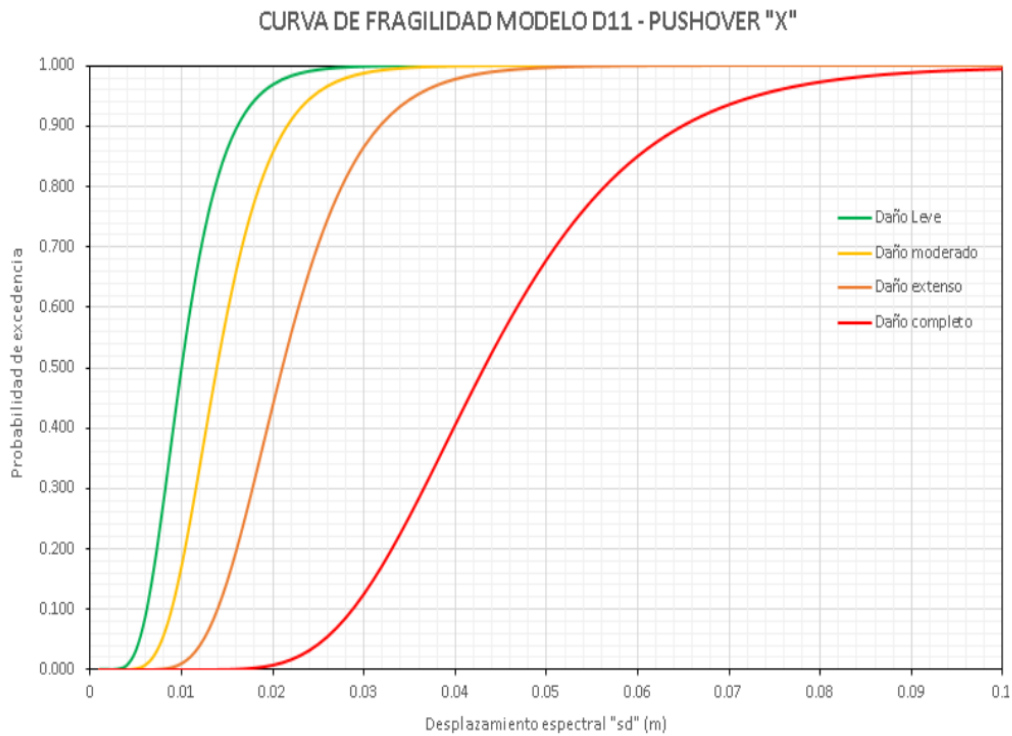


Figura 61. Curva de Fragilidad Modelo D11- Pushover "y". Fuente: Toro & Flores, 2024

Por otro lado, al utilizar el código para obtener curvas de fragilidad mediante el método FEMA 440 alternativa C, utilizando una curva de capacidad con un periodo inicial de 0.27 segundos, se observó que no se generó una curva de fragilidad para el caso de daño completo, como se muestra en la Figura 63. Esto se debe a que no se identificaron puntos de excedencia suficientes para llevar a cabo el análisis correspondiente, tal como se evidencia en la Figura 62. El análisis fue realizado con un conjunto de 520 registros sísmicos, que aparentemente no son lo suficientemente representativos para capturar adecuadamente el comportamiento de dicha estructura ante escenarios de daño completo. Este resultado sugiere que podría ser necesario ampliar la base de datos sísmicos.

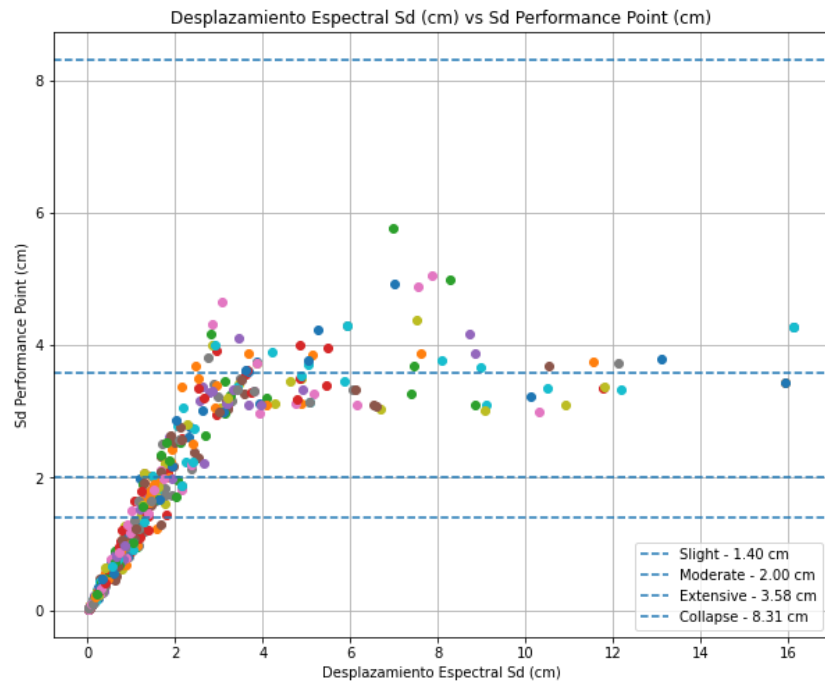


Figura 62 Gráfica para la obtención de curvas de Fragilidad para curva de capacidad con periodo de 0.27s. Fuente: Elaboración propia.

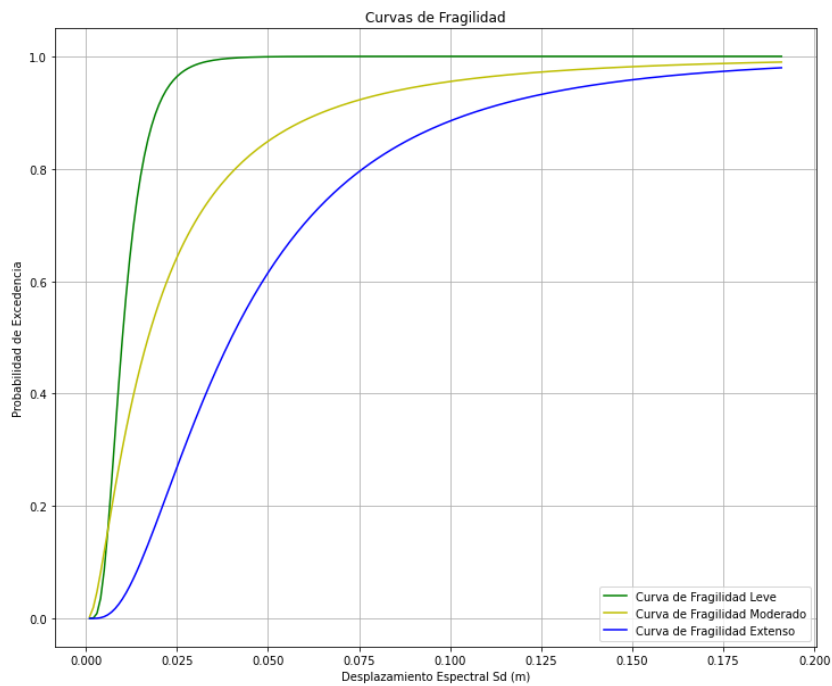


Figura 63 Curva de Fragilidad obtenida para la Curva de Capacidad con periodo de 0.27s. Fuente: Elaboración Propia.

4 CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Una vez presentadas las curvas de fragilidad empleando el código desarrollado, se concluyen los siguientes aspectos clave:

- Este código es aplicable exclusivamente a los resultados derivados de un análisis estático no lineal, comúnmente referido como Curva de Capacidad o análisis Pushover.
- La curva de capacidad debe representar fielmente el comportamiento global de la estructura, y así asegurar que los cálculos realizados por el código muestren con precisión la realidad estructural.
- Una vez obtenida la curva de capacidad, es importante seleccionar de manera adecuada una base de datos de registros sísmicos. Estos registros deben abarcar una amplia gama de magnitudes y características, con el fin de garantizar que el análisis sea representativo y capture adecuadamente el espectro de posibles demandas sísmicas que la estructura podría experimentar.
- Se deberá conocer y proporcionar los valores input mencionados en el desarrollo, estos son: participación modal (α), y factor de forma (PF), peso sísmico W (Ton), Número de pisos de la estructura estudiada, tipo de suelo.
- Al momento de utilizar el código para la obtención de curvas de fragilidad, el tiempo que tome en procesar los archivos es directamente proporcional a la cantidad de registros sísmicos utilizados para su análisis.

De acuerdo con los resultados obtenidos, se concluye que el método de coeficientes propuesto por el ASCE 41-17 para la determinación, de curvas de fragilidad muestra una tendencia lineal, lo cual limita su capacidad para proporcionar una estimación precisa de la probabilidad estadística. Por otro lado, el método propuesto por el FEMA 440 alternativa C, presenta una mayor dispersión en los datos, lo que resulta en una caracterización más realista de

la incertidumbre y una probabilidad estadística más representativa. Esta mayor dispersión permite un análisis más detallado y ajustado a la variabilidad de los registros sísmicos, mejorando la precisión en la estimación de la vulnerabilidad estructural.

Según la metodología empleada, el código desarrollado es aplicable a diferentes tipos de estructuras, cuando se utilice el método FEMA 440, el cual no presenta limitaciones significativas en cuanto al tipo de material o sistema estructural, permitiendo su uso en estructuras de acero, hormigón, madera, entre otros.

Por otro lado, el código utilizado para el método de coeficientes propuesto por el ASCE 41-17 presenta mayores restricciones, ya que requiere parámetros específicos que varían según el tipo de material estructural. Esto implica que, para materiales como hormigón o acero, se deben utilizar valores diferentes para los coeficientes involucrados en el cálculo de los puntos de desempeño. Sin embargo, para estructuras de madera, el método no menciona cómo obtener el punto de desempeño, lo que limita aún más su aplicabilidad. Esta dependencia en los materiales restringe el uso general del código, lo que obliga a realizar ajustes o modificaciones adicionales para adaptarse a estructuras que no sean de hormigón, disminuyendo así su versatilidad.

En estructuras con un periodo de vibración inicial menor a 0.5 segundos, se debe implementar un catálogo extenso de registros sísmicos con el fin de obtener resultados más representativos y precisos para cada nivel de daño. Esto se debe hacer para cualquiera de las dos metodologías empleadas, ya sea el Método de los coeficientes ASCE 41-17o el método FEMA440 Método C (gráfico).

Para finalizar, se debe tener en cuenta que la implementación de un código en Python para la realización de curvas de fragilidad para cualquier estructura es solo el inicio de una serie de estudios que se pueden realizar más adelante. Por lo tanto, este código está abierto a la manipulación, corrección y su respectivo mejoramiento.

4.2 Recomendaciones

Se recomienda utilizar el código para la obtención de curvas de fragilidad en diferentes tipos de estructuras con diversas tipologías estructurales. Esto permitirá evaluar la flexibilidad y adaptabilidad del código en una variedad de sistemas estructurales y materiales. Al aplicar el código a diferentes casos, se podrá identificar si es necesario realizar cambios o modificaciones en los parámetros o algoritmos, con el fin de optimizar el análisis para cada tipo de estructura y asegurar la precisión de los resultados.

Se sugiere llevar a cabo estudios que permitan la obtención de curvas de fragilidad en función de la variable "Drift" (desplazamiento relativo entre pisos), ya que a lo largo del desarrollo de este trabajo de titulación se ha observado que estas curvas también pueden presentarse en dicho formato. El uso de esta variable es especialmente relevante en el análisis del comportamiento estructural bajo cargas sísmicas, dado que es un parámetro directamente relacionado con el daño estructural. Incluir esta variable en el análisis proporcionaría una visión más completa y comparativa de la vulnerabilidad de las estructuras frente a diferentes demandas sísmicas.

Para concluir esta sección se recomienda modificar los códigos para que presenten los resultados finales en Aceleraciones espectrales S_a , no se realizó este apartado debido a que no es la finalidad del presente trabajo de titulación. Adicionalmente, se propone la generación de curvas de fragilidad para otras estructuras. En caso de que sea necesario realizar ajustes en el código, ya sea debido a la metodología seleccionada o a otros factores relevantes, se realicen las modificaciones correspondientes. Esto permitirá la mejora continua del código, facilitando su adaptación y aplicabilidad a diversas tipologías estructurales.

5 REFERENCIAS

- Agudelo C., J. A., & López, R. R. (2009). *Curvas de fragilidad para estructuras de pórticos de hormigón armado con paredes de mampostería. Caso de estudio: Estructuras de 1 y 2 niveles en Puerto Rico*. Revista Internacional de Desastres Naturales, Accidentes e Infraestructura Civil, 9(1-2), 163.
- American Society of Civil Engineers (ASCE). (2017). *Seismic Evaluation and Retrofit of Existing Buildings (ASCE/SEI 41-17)*. Reston, VA: American Society of Civil Engineers.
- Applied Technology Council. (1996). *Seismic Evaluation and Retrofit of Concrete Buildings (ATC-40 Report)*. Redwood City, CA: Author.
- Caicedo, C., Barbat, A. H., Canas, J. A., & Aguiar, R. (1994). *Vulnerabilidad sísmica de edificios (Monografía CIMNE IS-6)*. Centro Internacional de Métodos Numéricos en Ingeniería.
- Calderón, E. (2018). *Curvas de fragilidad según el método del espectro de capacidad para tipologías estructurales definidas en el Código Sísmico de Costa Rica*. Universidad de Costa Rica.
- Chávez, B. A. (2016). *Evaluación de la vulnerabilidad sísmica de las edificaciones de la ciudad de Quito - Ecuador y riesgo de pérdida (Tesis de maestría, Escuela de Posgrado en Ingeniería y Ciencias, Quito, Ecuador)*.
- Chiroiu, L., Goula, X., Roca, A., Susagna, T., Barbat, A., & Pujades, L. (2005). *El método del espectro de capacidad aplicado a la evaluación de daños sísmicos*. Presentado en el 2º Congreso Iberoamericano de Ingeniería Sísmica, Asociación Española de Ingeniería Sísmica (AEIS).
- Chopra, A. K. (2014). *Dynamics of structures: Theory and applications to earthquake engineering (4th ed.)*. Pearson.
- Coronel D., G., & López, O. A. (2013). Metodología para la estimación de daños por sismos en edificios escolares de Venezuela mediante curvas de fragilidad. *Revista de la Facultad de Ingeniería U.C.V*, 28(2), 65-86.
- Coronel D., G., López, O. A., & Betancourt, N. (2010). *Evaluación de niveles de riesgo sísmico en escuelas de Venezuela mediante curvas de*

- fragilidad*. Ponencia presentada en el 10º Congreso Chileno de Sismología e Ingeniería Antisísmica, Valdivia, Chile.
- Del Salto, M. B. (2023). *Vulnerabilidad sociodemográfica ante amenazas naturales en el perfil costero norte del Ecuador*.
- Dunn, K. B. (2016). *Rehabilitación estructural de la Facultad de Ingeniería mediante aisladores sísmicos* (Trabajo de titulación, Universidad Católica de Santiago de Guayaquil, Facultad de Ingeniería Civil). Guayaquil, Ecuador.
- Federal Emergency Management Agency (FEMA). (2005). *Improvement of Nonlinear Static Seismic Analysis Procedures* (FEMA 440). Prepared by Applied Technology Council (ATC-55 Project). Washington, D.C.: Department of Homeland Security, Federal Emergency Management Agency.
- Flores, R. D., & Toro, C. M. (2024). *Cálculo de curvas de fragilidad de estructuras de hormigón armado de un piso alto con vigas banda y cubierta liviana en residencias ubicadas en vía a la Costa, Guayaquil – Ecuador*. Trabajo de titulación, Universidad Católica de Santiago de Guayaquil, Facultad de Ingeniería, Carrera de Ingeniería Civil.
- Herrera, R. J. (2000). Amenaza sísmica y el código sísmico de Costa Rica. *Ingeniería*, 10(1-2), 107-124.
- Lizarazo, O. M. (2023). *Determinación del nivel de desempeño de un edificio habitacional mediante la aplicación del método de espectro de capacidad, diseñado según la norma sismorresistente colombiana (NSR-10). Comparación métodos FEMA 440 y N2*.
- Medina, R., & Music, J. (2018). Determinación del nivel de desempeño de un edificio habitacional estructurado en base a muros de hormigón armado y diseñado según normativa chilena. *Obras y Proyectos*, 23, 63-77.
- Mena, U. (2002). *Evaluación del riesgo sísmico en zonas urbanas* (Tesis doctoral, Universidad Politécnica de Cataluña, Departamento de Ingeniería del Terreno, Cartográfica y Geofísica). Barcelona.
- Ministerio de Desarrollo Urbano y Vivienda. (2015). *Norma Ecuatoriana de la Construcción (NEC)*. Quito, Ecuador: Ministerio de Desarrollo Urbano y Vivienda.

- Moreno, R. (2006). *Evaluación del riesgo sísmico en edificios mediante análisis estático no lineal: Aplicación a diversos escenarios sísmicos de Barcelona* (Tesis doctoral). Universidad Politécnica de Cataluña, Escuela Técnica Superior de Ingenieros de Caminos, Canales y Puertos.
- Moreno, R., & Bairán, J. M. (2010). Curvas de fragilidad para evaluar el daño sísmico en edificios de concreto armado con losas reticulares. *Revista de la Facultad de Ingeniería U.C.V.*, 25(4), 61-71.
- Moreno, R., & Bairán, J. M. (2014). Estudio sísmico de los edificios agregados de obra de fábrica utilizando el método del espectro de capacidad. *Informes de la Construcción*, 66(533), e001.
- Moreno González, R., Bairán, J. M., & Rodas, W. A. (2011). Vulnerabilidad sísmica en edificios metálicos. *Hormigón y Acero*, 62(261), 69-80.
- Parra, H. A. (2016). *Desarrollos metodológicos y aplicaciones hacia el cálculo de la peligrosidad sísmica en el Ecuador continental y estudio de riesgo sísmico en la ciudad de Quito* (Tesis doctoral). Universidad Politécnica de Madrid, Madrid, España.
- Rojas, R. J. (2010). *Curvas de fragilidad sísmica para edificios aporricados de concreto reforzado de poca altura*. Trabajo especial de grado, Universidad Central de Venezuela, Caracas, Venezuela.
- Sáenz, U. H. (2019). Vulnerabilidad sísmica en edificaciones esenciales mediante curvas de fragilidad analíticas—edificio administrativo de la Universidad Nacional del Centro del Perú.
- Soria, G. H. (2010). *Curvas de fragilidad para puentes carreteros típicos del Pacífico Mexicano*. Instituto de Ingeniería, Universidad Nacional Autónoma de México.
- Vargas, Y. F., Pujades, L. G., Barbat, A. H., & Hurtado, J. E. (2013). Evaluación probabilista de la capacidad, fragilidad y daño sísmico de edificios de hormigón armado. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 29(2), 63-78.
- Vásquez, A. P. (2016). *Desarrollos metodológicos y aplicaciones hacia el cálculo de la peligrosidad sísmica en el Ecuador continental y estudio de riesgo sísmico en la ciudad de Quito*.

Yépez, F., Barbat, A. H., & Canas, J. A. (1995). *Riesgo, peligrosidad y vulnerabilidad sísmica de edificios de mampostería* (Monografía CIMNE IS-12). Centro Internacional de Métodos Numéricos en Ingeniería.

6 Anexos

6.1 Anexo 1 Curva de fragilidad usando metodología ASCE41-17

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.integrate import trapz
from concurrent.futures import ThreadPoolExecutor, as_completed
from numpy.polynomial.polynomial import Polynomial
from scipy.stats import norm, binom
from lmfit import minimize, Parameters
from scipy.interpolate import UnivariateSpline

# Función de fragilidad teórica
def fragilidad_teorica(Sd, theta, beta):
    return norm.cdf((np.log(Sd) - np.log(theta)) / beta)

# Función para calcular la probabilidad
def calcular_probabilidad(casos_excedencia, num_analisis,
    fragilidad_teorica):
    return binom.pmf(casos_excedencia, num_analisis, fragilidad_teorica)

# Función objetivo para la optimización usando lmfit
def funcion_objetivo(params, Sd, casos_excedencia, num_analisis):
    theta = params['theta']
    beta = params['beta']
    if theta < 0.01 or beta < 0.1:
        return np.full_like(Sd, np.inf) # Penalización por valores fuera
de los límites

    fragilidades_teoricas = fragilidad_teorica(Sd, theta, beta)

    ln_probabilidades = np.array([np.log(calcular_probabilidad(e,
num_analisis, ft) + 1e-10) for e, ft in zip(casos_excedencia,
fragilidades_teoricas)])

    penalizacion = np.abs(fragilidades_teoricas[-1] - 1) # Penalización
por no alcanzar un valor de fragilidad teórica cercano a 1
```

```
    return -ln_probabilidades + penalizacion # Negativo para maximizar,
sumando la penalización
```

```
# Función para optimizar  $\theta$  y  $\beta$  usando lmfit
```

```
def optimizar_theta_beta(Sd, casos_excedencia, num_analisis):
```

```
    params = Parameters()
```

```
    params.add('theta', value=0.05, min=0.01)
```

```
    params.add('beta', value=0.45, min=0.1)
```

```
    result = minimize(funcion_objetivo, params, args=(Sd,
casos_excedencia, num_analisis), method='leastsq')
```

```
    return result.params['theta'].value, result.params['beta'].value #
Retorna  $\theta$  y  $\beta$  optimizados
```

```
# Función para procesar archivos sísmicos
```

```
def procesar_archivo(nombre_archivo):
```

```
    try:
```

```
        datosacel = np.loadtxt(nombre_archivo)
```

```
    except Exception as e:
```

```
        print(f"Error leyendo el archivo {nombre_archivo}: {e}")
```

```
        return None
```

```
    ug = datosacel / 981 # Convertir de cm/s2 a g
```

```
    ts = 0.01 # Pasos del registro
```

```
    tg = np.arange(0, len(ug) * ts, ts) # Vector Tiempo
```

```
    m = 1 # masa
```

```
    zita = 0.05 # Amortiguamiento
```

```
    Tn = np.arange(0, 3.0, 0.04) # Periodo
```

```
    ESPECTD = []
```

```
    ESPECTA = []
```

```
    for T in Tn:
```

```
        if T == 0:
```

```
            continue # Evitar división por cero
```

```
            wn = 2 * np.pi / T # frecuencia del sistema
```

```
            k = wn**2 * m # Rigidez
```



```

c = 2 * zita * wn * m # amortiguamiento [N s/m]

uo = 0 # [m]
vo = 0 # [m/s]
ao1 = 0 # [m/s^2]
dt = ts
po = -m * ug

y = 1 / 2
be = 1 / 6 # Aceleración lineal

ao = (po[0] - c * vo - k * uo) / m
a1 = (m / (be * dt**2)) + (y / (be * dt)) * c
a2 = (m / (be * dt)) + ((y / be) - 1) * c
a3 = (1 / (2 * be) - 1) * m + dt * ((y / (2 * be)) - 1) * c
kk = k + a1

u = [uo]
u1 = [vo]
u2 = [ao1]

for i in range(len(tg) - 1):
    p1 = po[i + 1] + a1 * u[i] + a2 * u1[i] + a3 * u2[i]
    u_new = p1 / kk if kk != 0 else 0
    u1_new = (3 * (u_new - u[i]) / dt) - 2 * u1[i] - (dt * u2[i] /
2)
    u2_new = (6 * (u_new - u[i]) / (dt**2)) - 2 * u2[i] - (6 *
u1[i] / dt)

    u.append(u_new)
    u1.append(u1_new)
    u2.append(u2_new)

ESPECTD.append(max(u) * 10000) # Convertir de m a cm
ESPECTA.append(max(u) * wn**2)

```

```

# Ajustar las longitudes si es necesario
min_len = min(len(Tn), len(ESPECTA), len(ESPECTD))
Tn = Tn[:min_len]
ESPECTA = np.array(ESPECTA[:min_len])
ESPECTD = np.array(ESPECTD[:min_len])

return Tn, ESPECTA, ESPECTD

# Función para analizar archivos sísmicos
def analizar_sismos(carpeta):
    archivos = [f for f in os.listdir(carpeta) if f.endswith('.txt')]
    resultados = []

    with ThreadPoolExecutor() as executor:
        future_to_archivo = {executor.submit(procesar_archivo,
os.path.join(carpeta, archivo)): archivo for archivo in archivos}
        for future in as_completed(future_to_archivo):
            archivo = future_to_archivo[future]
            try:
                result = future.result()
                if result:
                    resultados.append((os.path.splitext(archivo)[0],
*result))
            except Exception as e:
                print(f"Error procesando el archivo {archivo}: {e}")

    return resultados

# Función para contar los registros exitosos
def contar_registros_exitosos(resultados):
    num_registros = len(resultados)
    print(f"Número de registros analizados: {num_registros}")
    return num_registros

# Función para graficar resultados
def plot_resultados(resultados):

```

```

plt.figure(figsize=(10, 8))

for archivo, Tn, ESPECTA in resultados:
    plt.plot(Tn, ESPECTA, label=archivo)

plt.xlabel('Periodo (s)')
plt.ylabel('Aceleración Espectral (g)')
plt.title('Aceleraciones Espectrales vs Periodo para cada Sismo')
plt.legend()
plt.grid(True)
plt.show()

# Función para leer datos de push over
def read_push_over_data(file_path):
    if not os.path.exists(file_path):
        print(f"Error: El archivo {file_path} no existe.")
        return None, None

    data = pd.read_csv(file_path, delim_whitespace=True, header=None,
names=['Displacement', 'Force'])
    return data['Displacement'].values, data['Force'].values

# Función para calcular el área bajo la curva
def calculate_area_under_curve(x, y, end_index):
    x_segment = np.concatenate(([0], x[:end_index + 1]))
    y_segment = np.concatenate(([0], y[:end_index + 1]))
    area = trapz(y_segment, x_segment)
    return area

# Función para calcular el área bilineal completa
def bilinear_area_complete(Dyi, Ayi, Dpi, Api):
    if Dyi == 0 or Dpi == 0:
        return 0

    m1 = Ayi / Dyi
    area1 = (m1 / 2) * Dyi**2
    m2 = (Api - Ayi) / (Dpi - Dyi) if (Dpi - Dyi) != 0 else 0
    b2 = Ayi - m2 * Dyi

```

```

    area2 = (m2 / 2) * (Dpi**2 - Dyi**2) + b2 * (Dpi - Dyi)
    total_area_under_curve = area1 + area2
    return total_area_under_curve

# Función para calcular el error
def calculate_error(displacements, forces, Dyi, Ayi, Dpi, Api):
    area_under_curve = calculate_area_under_curve(displacements, forces,
np.argmax(displacements == Dpi))
    area_bilineal = bilinear_area_complete(Dyi, Ayi, Dpi, Api)
    if area_under_curve == 0:
        return 0
    error = abs(area_under_curve - area_bilineal) / area_under_curve * 100
    return error

# Función para encontrar el punto de cambio de pendiente
def find_slope_change_point(displacements, forces):
    slopes = []
    for i in range(1, len(displacements)):
        if displacements[i] != displacements[i-1]:
            slope = (forces[i] - forces[i-1]) / (displacements[i] -
displacements[i-1])
            slopes.append(slope)
        else:
            slopes.append(np.nan)

    initial_slope = slopes[0]
    for i in range(1, len(slopes)):
        if not np.isnan(slopes[i]) and abs(slopes[i] - initial_slope) >
0.1 * abs(initial_slope):
            return i
    return -1

# Función para convertir a ADRS
def convert_to_adrs(displacements, forces, W, alpha, Pf):
    Sa = forces / (W * alpha)
    Sd = displacements / Pf
    Sd = np.maximum(Sd, 0) # Reemplazar valores negativos por 0

```

```

return Sd, Sa

# Función para interpolar el valor de Co
def interpolate_co(n_floors):
    co_values = {1: 1, 2: 1.2, 3: 1.3, 4: 1.35, 5: 1.4, 10: 1.5}
    if n_floors in co_values:
        return co_values[n_floors]
    elif 5 < n_floors < 10:
        return np.interp(n_floors, [5, 10], [1.4, 1.5])
    else:
        return np.nan

# Función para obtener los valores de Co y Cm. Si la estructura es mayor a
dos pisos revisar Tabla7-4 del ASCE41-17 para obtener valor de Cm
def obtener_valores(n_floors):
    Co = interpolate_co(n_floors)
    Cm = 1 if n_floors <= 2 else 0.9
    return Co, Cm

# Función para obtener el valor de a
def obtener_valor_a():
    conoce_suelo = input("¿Conoce el tipo de Suelo? (si/no):
").strip().lower()
    if conoce_suelo == "si":
        tipo_suelo = input("Ingrese Tipo de Suelo (A, B, C, D, E, F):
").strip().upper()
        if tipo_suelo in ["A", "B"]:
            return 130
        elif tipo_suelo == "C":
            return 90
        elif tipo_suelo in ["D", "E", "F"]:
            return 60
        else:
            print("Tipo de suelo no reconocido, usando valor por defecto
para suelo tipo C.")
            return 90
    else:

```

```

return 90

# Función para graficar Desplazamiento Espectral Sd vs Desplazamiento
Objetivo  $\delta t$ 

def
Desplazamiento_Espectral_Sd_vs_Desplazamiento_Objetivo_dt(puntos_desempeño
, Slight, Moderate, Extensive, Collapse, num_registros):
    plt.figure(figsize=(10, 8))
    plt.title('Desplazamiento Espectral Sd (cm) vs Desplazamiento Objetivo
 $\delta t$ ')

    for Sd, delta_t, archivo in puntos_desempeño:
        plt.plot(Sd, delta_t, 'o')

    # Definir los límites intercambiados
    limites = [Slight, Moderate, Extensive, Collapse]
    nombres_limites = ['Slight', 'Moderate', 'Extensive', 'Collapse']

    for limite, nombre in zip(limites, nombres_limites):
        plt.axhline(y=limite, linestyle='--', label=f'{nombre} -
{limite:.2f} cm')

    plt.xlabel('Desplazamiento Espectral Sd (cm)')
    plt.ylabel('Desplazamiento Objetivo  $\delta t$  (cm)')
    plt.grid(True)
    plt.legend()
    plt.show()

    # Clasificar y contar los casos de daño
    daño_leve = sum(1 for _, delta_t, _ in puntos_desempeño if Slight <
delta_t <= Moderate)
    daño_moderado = sum(1 for _, delta_t, _ in puntos_desempeño if
Moderate < delta_t <= Extensive)
    daño_extenso = sum(1 for _, delta_t, _ in puntos_desempeño if
Extensive < delta_t <= Collapse)
    daño_completo = sum(1 for _, delta_t, _ in puntos_desempeño if delta_t
> Collapse)

    print(f"Daño Leve: {daño_leve}")

```

```

print(f"Daño Moderado: {daño_moderado}")
print(f"Daño Extenso: {daño_extenso}")
print(f"Daño Completo: {daño_completo}")

# Pedir el valor del intervalo en X
intervalo = float(input("Insertar valor de intervalo: "))
if intervalo < 2:
    intervalo = 2

# Pedir el último valor de análisis
ultimo_valor = float(input("Inserte último valor de análisis: "))

num_intervals = int(np.ceil(ultimo_valor / intervalo))

promedios = []
excedencias_leve = []
excedencias_moderado = []
excedencias_extenso = []
excedencias_completo = []

total_excedencias_leve = 0
total_excedencias_moderado = 0
total_excedencias_extenso = 0
total_excedencias_completo = 0

for i in range(num_intervals):
    lower_bound = i * intervalo
    upper_bound = (i + 1) * intervalo
    interval_points = [Sd for Sd, delta_t, _ in puntos_desempeño if
lower_bound < Sd <= upper_bound]

    if interval_points:
        avg_Sd = np.mean(interval_points) / 100 # Convertir de cm a m
        promedios.append(avg_Sd)

# Casos de Excedencia Leve

```

```

        if i == 0:
            excedencia_points_leve = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Slight < delta_t
<= Moderate]

            excedencias_leve.append(len(excedencia_points_leve) - 1)

            total_excedencias_leve = excedencias_leve[-1]

        else:
            excedencia_points_leve = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Slight < delta_t
<= Moderate]

            if len(excedencia_points_leve) > 0:
                excedencias_leve.append(total_excedencias_leve +
len(excedencia_points_leve))

                total_excedencias_leve = excedencias_leve[-1]

            else:
                excedencias_leve.append(daño_leve)

# Casos de Excedencia Moderado

        if i == 0:
            excedencia_points_moderado = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Moderate < delta_t
<= Extensive]

excedencias_moderado.append(len(excedencia_points_moderado))

            total_excedencias_moderado = excedencias_moderado[-1]

        else:
            excedencia_points_moderado = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Moderate < delta_t
<= Extensive]

            if len(excedencia_points_moderado) > 0:
                excedencias_moderado.append(total_excedencias_moderado
+ len(excedencia_points_moderado))

                total_excedencias_moderado = excedencias_moderado[-1]

            else:
                excedencias_moderado.append(daño_moderado)

# Casos de Excedencia Extenso

        if i == 0:
            excedencia_points_extenso = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Extensive <
delta_t <= Collapse]

```



```

        excedencias_extenso.append(len(excedencia_points_extenso))
        total_excedencias_extenso = excedencias_extenso[-1]
    else:
        excedencia_points_extenso = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Extensive <
delta_t <= Collapse]
        if len(excedencia_points_extenso) > 0:
            excedencias_extenso.append(total_excedencias_extenso +
len(excedencia_points_extenso))
            total_excedencias_extenso = excedencias_extenso[-1]
        else:
            excedencias_extenso.append(daño_extenso)

# Casos de Excedencia Completo
if i == 0:
    excedencias_completo.append(0)
else:
    excedencia_points_completo = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and delta_t >
Collapse]
    if len(excedencia_points_completo) > 0:
        total_excedencias_completo +=
len(excedencia_points_completo)
excedencias_completo.append(total_excedencias_completo)
    else:
        if i == 1:
            excedencias_completo.append(0)
        elif i == num_intervals - 1:
            excedencias_completo.append(num_registros)
        else:
            excedencias_completo.append(excedencias_completo[-
1])

    else:
        promedios.append((lower_bound + upper_bound) / 200) #
Promedio en el intervalo
        excedencias_leve.append(daño_leve)
        excedencias_moderado.append(daño_moderado)

```

```

    excedencias_extenso.append(daño_extenso)
    if i == num_intervals - 1:
        excedencias_completo.append(num_registros)
    else:
        if i == 1:
            excedencias_completo.append(0)
        else:
            excedencias_completo.append(excedencias_completo[-1])

    print(f"Promedio Intervalo {i + 1} = {promedios[-1]:.4f} m")

# Ajustar todas las listas a la misma longitud
max_length = max(len(promedios), len(excedencias_leve),
len(excedencias_moderado), len(excedencias_extenso),
len(excedencias_completo))

while len(promedios) < max_length:
    promedios.append(np.nan)
while len(excedencias_leve) < max_length:
    excedencias_leve.append(np.nan)
while len(excedencias_moderado) < max_length:
    excedencias_moderado.append(np.nan)
while len(excedencias_extenso) < max_length:
    excedencias_extenso.append(np.nan)
while len(excedencias_completo) < max_length:
    excedencias_completo.append(np.nan)

return promedios, excedencias_leve, excedencias_moderado,
excedencias_extenso, excedencias_completo, daño_leve, daño_moderado,
daño_extenso, daño_completo

# Función para ajustar un polinomio
def fit_polynomial(Tn, ESPECT, degree=3):
    p = Polynomial.fit(Tn, ESPECT, degree)
    return p.convert()

# Función para estimar valores utilizando interpolación spline

```

```

def estimate_values(Tn, ESPECTA, ESPECTD, Te):
    # Crear funciones de interpolación spline
    spline_interp_Sa = UnivariateSpline(Tn, ESPECTA, s=0)
    spline_interp_Sd = UnivariateSpline(Tn, ESPECTD, s=0)

    # Interpolar valores
    Sa_Te = spline_interp_Sa(Te)
    Sd_Te = spline_interp_Sd(Te) / 10 # Convertir de mm a cm

    return Sa_Te, Sd_Te

# Función para crear tablas
def crear_tablas(promedios, excedencias_leve, excedencias_moderado,
excedencias_extenso, excedencias_completo, niveles_dano):
    tablas = {}

    for nivel, casos in niveles_dano.items():
        if casos == 0:
            continue # Omitir nivel de daño si no hay casos
        if nivel == 'Leve':
            theta_beta_opt = optimizar_theta_beta(promedios,
excedencias_leve, casos)
            fragilidades_teoricas = [fragilidad_teorica(Sd,
theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]
            probabilidades = [calcular_probabilidad(e, casos, ft) for e,
ft in zip(excedencias_leve, fragilidades_teoricas)]
            ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
probabilidades]
            data = {
                'Sd (m)': promedios,
                'Número de Análisis': [casos] * len(promedios),
                'Casos de Excedencia': excedencias_leve,
                'Fracción que causa el colapso': [e / casos for e in
excedencias_leve],
                'Función de fragilidad teórica': fragilidades_teoricas,
                'Probabilidad': probabilidades,
                'LN Probabilidad': ln_probabilidades,
                'θ': theta_beta_opt[0],

```

```

        'β': theta_beta_opt[1]
    }
    elif nivel == 'Moderado':
        theta_beta_opt = optimizar_theta_beta(promedios,
        excedencias_moderado, casos)

        fragilidades_teoricas = [fragilidad_teorica(Sd,
        theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]

        probabilidades = [calcular_probabilidad(e, casos, ft) for e,
        ft in zip(excedencias_moderado, fragilidades_teoricas)]

        ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
        probabilidades]

        data = {
            'Sd (m)': promedios,
            'Número de Análisis': [casos] * len(promedios),
            'Casos de Excedencia': excedencias_moderado,
            'Fracción que causa el colapso': [e / casos for e in
            excedencias_moderado],
            'Función de fragilidad teórica': fragilidades_teoricas,
            'Probabilidad': probabilidades,
            'LN Probabilidad': ln_probabilidades,
            'θ': theta_beta_opt[0],
            'β': theta_beta_opt[1]
        }

    elif nivel == 'Extenso':
        theta_beta_opt = optimizar_theta_beta(promedios,
        excedencias_extenso, casos)

        fragilidades_teoricas = [fragilidad_teorica(Sd,
        theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]

        probabilidades = [calcular_probabilidad(e, casos, ft) for e,
        ft in zip(excedencias_extenso, fragilidades_teoricas)]

        ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
        probabilidades]

        data = {
            'Sd (m)': promedios,
            'Número de Análisis': [casos] * len(promedios),
            'Casos de Excedencia': excedencias_extenso,
            'Fracción que causa el colapso': [e / casos for e in
            excedencias_extenso],
            'Función de fragilidad teórica': fragilidades_teoricas,

```

```

        'Probabilidad': probabilidades,
        'LN Probabilidad': ln_probabilidades,
        'θ': theta_beta_opt[0],
        'β': theta_beta_opt[1]
    }
    elif nivel == 'Completo':
        theta_beta_opt = optimizar_theta_beta(promedios,
        excedencias_completo, casos)
        fragilidades_teoricas = [fragilidad_teorica(Sd,
        theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]
        probabilidades = [calcular_probabilidad(e, casos, ft) for e,
        ft in zip(excedencias_completo, fragilidades_teoricas)]
        ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
        probabilidades]
        data = {
            'Sd (m)': promedios,
            'Número de Análisis': [casos] * len(promedios),
            'Casos de Excedencia': excedencias_completo,
            'Fracción que causa el colapso': [e / casos for e in
            excedencias_completo],
            'Función de fragilidad teórica': fragilidades_teoricas,
            'Probabilidad': probabilidades,
            'LN Probabilidad': ln_probabilidades,
            'θ': theta_beta_opt[0],
            'β': theta_beta_opt[1]
        }
        tablas[nivel] = pd.DataFrame(data)

    # Ajustar último valor de "Casos de Excedencia" para "Daño
    Completo" igual al número de análisis
    if nivel == 'Completo' and len(tablas[nivel]) > 0:
        tablas[nivel].at[len(tablas[nivel]) - 1, 'Casos de
        Excedencia'] = casos
        tablas[nivel].at[len(tablas[nivel]) - 1, 'Fracción que causa
        el colapso'] = 1.0
        tablas[nivel].at[len(tablas[nivel]) - 1, 'Probabilidad'] =
        calcular_probabilidad(casos, casos, fragilidades_teoricas[-1])
        tablas[nivel].at[len(tablas[nivel]) - 1, 'LN Probabilidad'] =
        np.log(calcular_probabilidad(casos, casos, fragilidades_teoricas[-1]))

```

```

# Calcular la sumatoria de LN Probabilidad
suma_ln_probabilidad = tablas[nivel]['LN Probabilidad'].sum()

# Agregar fila de sumatoria al final de la tabla
suma_fila = {
    'Sd (m)': 'Σ=',
    'Número de Análisis': '',
    'Casos de Excedencia': '',
    'Fracción que causa el colapso': '',
    'Función de fragilidad teórica': '',
    'Probabilidad': '',
    'LN Probabilidad': suma_ln_probabilidad
}

tablas[nivel] = pd.concat([tablas[nivel],
pd.DataFrame([suma_fila])], ignore_index=True)

return tablas

# Función para exportar tablas a Excel
def exportar_a_excel(tablas, nombre_archivo):
    with pd.ExcelWriter(nombre_archivo) as writer:
        for nivel, tabla in tablas.items():
            tabla.to_excel(writer, sheet_name=nivel, index=False)

# Función para graficar curvas de fragilidad
def graficar_curvas_fragilidad(tablas, niveles_dano, max_Sd):
    max_x = (1.75 * max_Sd) # Determinar el valor máximo de x
    x = np.arange(0.001, max_x, 0.001) # Rango para la curva de
distribución normal

    plt.figure(figsize=(12, 10))
    plt.title('Curvas de Fragilidad')
    plt.xlabel('Desplazamiento Espectral Sd (m)')
    plt.ylabel('Probabilidad de Excedencia')
    colores = ['g', 'y', 'b', 'r'] # Colores para las curvas de
fragilidad

```

```

for nivel, tabla in tablas.items():
    if nivel not in niveles_dano:
        continue

    theta = tabla['θ'][0]
    beta = tabla['β'][0]

    # Graficar curva de fragilidad con distribución normal
    plt.plot(x, fragilidad_teorica(x, theta, beta), '-', label=f'Curva
de Fragilidad {nivel}', color=colores.pop(0))

plt.legend()
plt.grid(True)
plt.show()

# Función principal
def main():
    carpeta = 'Registros Sismicos'
    resultados = analizar_sismos(carpeta)

    # Contar y mostrar el número de registros analizados
    num_registros = contar_registros_exitosos(resultados)

    file_path = 'DatosPushOver/PushOver.txt'
    displacements, forces = read_push_over_data(file_path)
    if displacements is None or forces is None:
        return

    plt.figure(figsize=(8, 6))
    plt.plot(displacements, forces, marker='o', linestyle='-', color='r',
label='Curva Pushover')
    plt.xlabel('Desplazamiento (cm)')
    plt.ylabel('Cortante V (Tonf)')
    plt.title('Curva de Capacidad')
    plt.legend()

```

```

plt.grid(True)
plt.show()

pos = int(input("Ingrese la posición del punto Dui-Aui para la curva
pushover: ")) - 1

dy_ay = input("Ingrese las coordenadas iniciales del punto A para la
curva pushover (Dyi, Ayi): ")
Dyi, Ayi = map(float, dy_ay.split(','))

if 0 <= pos < len(displacements):
    area_OBD = calculate_area_under_curve(displacements, forces, pos)
    print(f"Área bajo la curva Pushover: {area_OBD:.2f}")

Dui = displacements[pos]
Aui = forces[pos]
initial_bilinear_area = bilinear_area_complete(Dyi, Ayi, Dui, Aui)
print(f"Área bajo la curva de la Representación bilineal inicial:
{initial_bilinear_area:.2f}")

error = calculate_error(displacements, forces, Dyi, Ayi, Dui, Aui)
print(f"Error de la representación bilineal inicial: {error:.2f}%")

calcular_nuevas_coordenadas = input("¿Calcular nuevas coordenadas para
el punto A? (si/no): ").strip().lower()

if calcular_nuevas_coordenadas == "si":
    while error > 1:
        Dyi *= 0.99
        Ayi *= 0.99
        error = calculate_error(displacements, forces, Dyi, Ayi, Dui,
Aui)

    print(f"Nuevas coordenadas del punto A: Dyi = {Dyi:.2f}, Ayi =
{Ayi:.2f}")
    print(f"Área bajo la curva de la Representación bilineal con el
punto A encontrado: {bilinear_area_complete(Dyi, Ayi, Dui, Aui): .2f}")
    print(f"Error de la representación bilineal: {error:.2f}%")

```



```

# Calcular el nuevo punto con 60% de las coordenadas de A
Dx_new = 0.6 * Dy_i
Ay_new = 0.6 * Ay_i

# Calcular la rigidez lateral efectiva Ke
Ke = Ay_new / Dx_new if Dx_new != 0 else 0
print(f"Rigidez lateral efectiva Ke: {Ke:.2f} Tonf/cm")
slope_change_point_index = find_slope_change_point(displacements,
forces)
if slope_change_point_index != -1:
    Dx_change = displacements[slope_change_point_index]
    Ay_change = forces[slope_change_point_index]
else:
    Dx_change, Ay_change = 0, 0

# Calcular la rigidez inicial Ki
Ki = Ay_change / Dx_change if Dx_change != 0 else 0
print(f"Rigidez inicial Ki: {Ki:.2f} Tonf/cm")

# Solicitar valores adicionales al usuario
W = float(input("Ingresar Peso Efectivo sísmico W (Ton): "))

conoce_factores = input("¿Conoce los valores de factor de
participación modal ( $\alpha$ ) y factor de forma de la estructura (Pf)? (si/no):
").strip().lower()

if conoce_factores == "si":
    alpha = float(input("Ingrese factor de participación modal ( $\alpha$ ):
"))
    Pf = float(input("Ingrese factor de forma de la estructura (Pf):
"))
else:
    alpha = 0.8
    Pf = 1.4

print(f"Factor de participación modal ( $\alpha$ ): {alpha}")
print(f"Factor de forma de la estructura (Pf): {Pf}")

```

```

plt.figure(figsize=(8, 6))

plt.plot(displacements, forces, marker='o', linestyle='-', color='r',
label='Curva Pushover')

plt.plot(0, 0, marker='o', color='black')
plt.text(0, 0, '0', ha='right', color='black')

if 0 <= pos < len(displacements):
    plt.plot(displacements[pos], forces[pos], marker='x',
markersize=8, color='blue')
    plt.plot([displacements[pos], displacements[pos]], [0,
forces[pos]], 'b--')
    plt.plot([0, displacements[pos]], [forces[pos], forces[pos]], 'b--')
    plt.text(displacements[pos], 0, 'Dui', ha='center', color='blue',
va='bottom')
    plt.text(0, forces[pos], 'Aui', ha='right', color='blue',
va='center')
    plt.plot(displacements[pos], forces[pos], marker='o',
markersize=8, color='red')
    plt.text(displacements[pos], forces[pos], 'B', ha='left',
color='red', va='bottom')

plt.plot(Dyi, Ayi, marker='o', markersize=8, color='green')
plt.text(Dyi, Ayi, 'A', ha='left', color='green', va='bottom')
plt.plot([0, Dyi], [0, Ayi], 'g--')
plt.plot(Dx_new, Ay_new, marker='o', markersize=8, color='purple')
plt.text(Dx_new, Ay_new, 'Pke', ha='right', color='purple', va='top')
if slope_change_point_index != -1:
    plt.plot(Dx_change, Ay_change, marker='o', markersize=8,
color='orange')
    plt.text(Dx_change, Ay_change, 'Pki', ha='right', color='orange',
va='top')
if 0 <= pos < len(displacements):
    plt.plot([Dyi, displacements[pos]], [Ayi, forces[pos]], 'g--')

plt.plot([0, Dyi, displacements[pos]], [0, Ayi, forces[pos]], 'g--',
label='Representación Bilineal')

plt.xlabel('Desplazamiento (cm)')

```

```

plt.ylabel('Cortante V (Tonf)')
plt.title('Representación Bilineal ASCE41-17 y Curva de Capacidad')
plt.legend()
plt.grid(True)
plt.show()

# Convertir puntos de curva Pushover y representación bilineal a ADRS
Sd_push, Sa_push = convert_to_adrs(displacements, forces, W, alpha,
Pf)
Sd_bilineal, Sa_bilineal = convert_to_adrs(np.array([0, Dyi, Dui]),
np.array([0, Ayi, Aui]), W, alpha, Pf)

plt.figure(figsize=(8, 6))
plt.plot(Sd_push, Sa_push, marker='o', linestyle='-', color='r',
label='Curva Pushover ADRS')
plt.plot(Sd_bilineal, Sa_bilineal, marker='x', linestyle='--',
color='b', label='Representación Bilineal ADRS')
plt.xlabel('Desplazamiento Espectral Sd (cm)')
plt.ylabel('Aceleración Espectral Sa (g)')
plt.title('Representación Bilineal ASCE41-17 ADRS y Curva de Capacidad
ADRS')
plt.legend()
plt.grid(True)
plt.show()

# Cálculo del período fundamental elástico Ti a partir de las
coordenadas espectrales del punto A
if len(Sd_bilineal) > 1:
    Sd_A = Sd_bilineal[1] # Coordenada Sd del punto A
    Sa_A = Sa_bilineal[1] # Coordenada Sa del punto A
    if Sa_A != 0:
        Ti = 2 * np.pi * np.sqrt(Sd_A / (Sa_A * 981))
        print(f"Período fundamental elástico Ti: {Ti:.2f} segundos")
    else:
        Ti = 0
        print("No se puede calcular el período fundamental elástico Ti
debido a que Sa_A es cero.")

```

```

# Solicitar el número de pisos
n_floors = int(input("Ingrese el número de pisos del edificio: "))

# Obtener valores de Co y Cm
Co, Cm = obtener_valores(n_floors)
print(f"Valor de Co: {Co}")
print(f"Valor de Cm: {Cm}")

# Obtener valor de a basado en el tipo de suelo
a = obtener_valor_a()
print(f"Valor de a: {a}")

# Cálculo del período fundamental efectivo Te
Te = Ti * np.sqrt(Ki / Ke) if Ke != 0 else 0
print(f"Período fundamental efectivo Te: {Te:.2f} segundos")

# Solicitar al usuario si desea ver las gráficas individuales de
puntos de desempeño para cada sismo
ver_graficas_individuales = input("¿Desea ver la gráfica de puntos de
desempeño correspondiente a cada sismo? (si/no): ").strip().lower()

# Lista para almacenar puntos de desempeño
puntos_desempeño = []

# Crear una lista para almacenar los resultados para exportación a
Excel
export_data = []

# Encontrar o interpolar Sa y Sd para el valor de Te para cada sismo
max_Sd_Te = 0 # Para determinar el valor máximo de Sd_Te

for archivo, Tn, ESPECTA, ESPECTD in resultados:
    if len(Tn) != len(ESPECTA) or len(Tn) != len(ESPECTD):
        print(f"Longitudes inconsistentes en el archivo {archivo}:
Tn={len(Tn)}, ESPECTA={len(ESPECTA)}, ESPECTD={len(ESPECTD)}")
        continue

```

```

# Usar ajuste de curvas polinomiales
Sa_Te, Sd_Te = estimate_values(Tn, ESPECTA, ESPECTD, Te)

# Reemplazar valores negativos por 0
Sd_Te = max(Sd_Te, 0)

# Calcular  $\mu$ strength
mu_strength = ((Sa_Te) / (Ayi / W)) * Cm

# Determinar C1 y C2 con condiciones
if Te > 1:
    C1 = 1
else:
    C1 = 1 + ((mu_strength - 1) / (a * Te**2))

if Te > 0.7:
    C2 = 1
else:
    C2 = 1 + (1 / 800) * ((mu_strength - 1) / Te)**2

# Calcular desplazamiento objetivo ( $\delta_t$ )
delta_t = Co * C1 * C2 * (Sa_Te) * (Te**2 / (4 * np.pi**2)) * 981

# Asegurar que delta_t no exceda el último desplazamiento de la
curva de capacidad
if delta_t > displacements[-1]:
    delta_t = displacements[-1]

# Interpolación en la curva Pushover
if delta_t > displacements[-1]:
    Sd_obj = displacements[-1]
    Sa_obj = forces[-1]
elif delta_t == 0:
    Sd_obj = 0
    Sa_obj = 0
else:

```

```

Sa_obj = np.interp(delta_t, displacements, forces)
Sd_obj = delta_t

if ver_graficas_individuales == "si":
    # Graficar punto de desempeño individual
    plt.figure(figsize=(10, 8))
    plt.plot(displacements, forces, marker='o', linestyle='-',
color='r', label='Curva de Capacidad')
    plt.plot(Sd_obj, Sa_obj, 'go', label=f'Punto de Desempeño
( $\delta t={\text{delta\_t:.2f}}$  cm)')
    plt.xlabel('Desplazamiento (cm)')
    plt.ylabel('Cortante (T)')
    plt.title(f'Punto de Desempeño para el sismo {archivo}')
    plt.legend()
    plt.grid(True)
    plt.show()

# Agregar punto de desempeño a la lista
puntos_desempeño.append((Sd_Te, delta_t, archivo))

# Actualizar el valor máximo de Sd_Te
if Sd_Te > max_Sd_Te:
    max_Sd_Te = 2 * Sd_Te

# Agregar datos a la lista de exportación
export_data.append([
    archivo, Sa_Te, Sd_Te, Co, C1, C2, Co * C1 * C2, delta_t
])

# Crear un DataFrame con los resultados para exportar a Excel
df_export = pd.DataFrame(export_data, columns=[
    'Nombre del Registro', 'Sa (Te) g', 'Sd (Te) cm', 'Co', 'C1',
'C2', 'Co*C1*C2', 'Desplazamiento Objetivo  $\delta t$  cm'
])

# Exportar el DataFrame a un archivo Excel
df_export.to_excel('resultados_sismos.xlsx', index=False)

```

```

print("Los resultados se han exportado a 'resultados_sismos.xlsx'.")

# Definir los umbrales de daño para Dyi, Dui
Slight = 0.7 * Dyi
Moderate = Dyi
Extensive = Dyi + 0.25 * (Dui - Dyi)
Collapse = Dui

# Graficar nueva gráfica "Desplazamiento Espectral Sd vs
Desplazamiento Objetivo δt"
promedios, excedencias_leve, excedencias_moderado,
excedencias_extenso, excedencias_completo, daño_leve, daño_moderado,
daño_extenso, daño_completo =
Desplazamiento_Espectral_Sd_vs_Desplazamiento_Objetivo_dt(puntos_desempeño
, Slight, Moderate, Extensive, Collapse, num_registros)

# Definir los casos totales por nivel de daño
niveles_dano = {
    'Leve': daño_leve,
    'Moderado': daño_moderado,
    'Extenso': daño_extenso,
    'Completo': daño_completo
}

tablas = crear_tablas(promedios, excedencias_leve,
excedencias_moderado, excedencias_extenso, excedencias_completo,
niveles_dano)

exportar_a_excel(tablas, 'Curvas de Fragilidad.xlsx')
print("Las tablas se han exportado a 'Curvas de Fragilidad.xlsx'.")

# Graficar curvas de fragilidad
graficar_curvas_fragilidad(tablas, niveles_dano, max(promedios))

if __name__ == "__main__":
    main()

```

6.2 Anexo 2 Curva de fragilidad usando metodología FEMA440 alternativa C

```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.integrate import trapz
from scipy.stats import norm, binom
from lmfit import minimize, Parameters
import matplotlib.cm as cm

def procesar_archivo(nombre_archivo):
    try:
        datosacel = np.loadtxt(nombre_archivo)
        ug = datosacel / 981 # Convertir de cm/s^2 a g
        ts = 0.01 # Pasos del registro
        tg = np.arange(0, len(ug) * ts, ts) # Vector Tiempo

        m = 1 # masa
        zita = 0.05 # Amortiguamiento
        Tn = np.arange(0, 3.0, 0.04) # Periodo

        ESPECTD = []
        ESPECTA = []

        for T in Tn:
            wn = 2 * np.pi / T # frecuencia del sistema
            k = wn**2 * m # Rigidez
            c = 2 * zita * wn * m # amortiguamiento [N s/m]

            uo = 0 # [m]
            vo = 0 # [m/s]
            ao1 = 0 # [m/s^2]
            dt = ts
            po = -m * ug
```



```

y = 1 / 2
be = 1 / 6 # Aceleración lineal

ao = (po[0] - c * vo - k * uo) / m
a1 = (m / (be * dt**2)) + (y / (be * dt)) * c
a2 = (m / (be * dt)) + ((y / be) - 1) * c
a3 = (1 / (2 * be) - 1) * m + dt * ((y / (2 * be)) - 1) * c
kk = k + a1

u = [uo]
u1 = [vo]
u2 = [ao1]

for i in range(len(tg) - 1):
    p1 = po[i + 1] + a1 * u[i] + a2 * u1[i] + a3 * u2[i]
    u_new = p1 / kk
    u1_new = (3 * (u_new - u[i]) / dt) - 2 * u1[i] - (dt *
u2[i] / 2)
    u2_new = (6 * (u_new - u[i]) / (dt**2)) - 2 * u2[i] - (6 *
u1[i] / dt)

    u.append(u_new)
    u1.append(u1_new)
    u2.append(u2_new)

ESPECTD.append(max(u))
ESPECTA.append(max(u) * wn**2)

ESPECTD = np.array(ESPECTD)
ESPECTA = np.array(ESPECTA)

return tg, ug, Tn, ESPECTD, ESPECTA

except Exception as e:
    print(f"Error procesando archivo {nombre_archivo}: {e}")

```

```

        return None, None, None, None, None

def read_push_over_data(file_path):
    if not os.path.exists(file_path):
        print(f"Error: El archivo {file_path} no existe.")
        return None, None

    data = pd.read_csv(file_path, delim_whitespace=True, header=None,
names=['Displacement', 'Force'])
    return data['Displacement'].values, data['Force'].values

def calculate_area_under_curve(x, y, end_index):
    x_segment = np.concatenate(([0], x[:end_index + 1]))
    y_segment = np.concatenate(([0], y[:end_index + 1]))
    area = trapz(y_segment, x_segment)
    return area

def bilinear_area_complete(Dyi, Ayi, Du, Au):
    m1 = Ayi / Dyi
    area1 = (m1 / 2) * Dyi**2
    m2 = (Au - Ayi) / (Du - Dyi)
    b2 = Ayi - m2 * Dyi
    area2 = (m2 / 2) * (Du**2 - Dyi**2) + b2 * (Du - Dyi)
    total_area_under_curve = area1 + area2
    return total_area_under_curve

def calculate_error(displacements, forces, Dyi, Ayi, Du, Au):
    area_under_curve = calculate_area_under_curve(displacements, forces,
np.argmax(displacements == Du))
    area_bilineal = bilinear_area_complete(Dyi, Ayi, Du, Au)
    error = abs(area_under_curve - area_bilineal) / area_under_curve * 100
    return error

def convert_to_adrs(displacements, forces, mass, alpha, Pf):
    Sa = forces / (mass * alpha)
    Sd = displacements / Pf
    return Sd, Sa

```

```

def calculate_post_elastic_rigidity_adrs(Dy, Ay, Du, Au):
    alpha_post_elastic = ((Au - Ay) / (Du - Dy)) / (Ay / Dy)
    return alpha_post_elastic

def calculate_ductility(Sd):
    Dy = Sd[1]
    Du = Sd[-1]
    return Du / Dy

def calculate_initial_period(Sd, Sa):
    Sd_2 = Sd[1]
    Sa_2 = Sa[1]
    To = 2 * np.pi * np.sqrt(Sd_2 / (Sa_2 * 981))
    return To

def calculate_teff(mu, To):
    if 1.0 < mu < 4.0:
        Teff = (0.20 * (mu - 1)**2 - 0.038 * (mu - 1)**3 + 1) * To
    elif 4.0 <= mu <= 6.5:
        Teff = (0.28 + 0.13 * (mu - 1) + 1) * To
    elif mu > 6.5:
        Teff = (0.89 * (np.sqrt((mu - 1) / (1 + 0.05 * (mu - 2))) - 1) +
1) * To
    else:
        Teff = To
    return Teff

def calculate_beff(mu, zita, Teff, To):
    if 1.0 <= mu < 4.0:
        Beff = 4.9 * (mu - 1)**2 - 1.1 * (mu - 1)**3 + 5
    elif 4.0 <= mu <= 6.5:
        Beff = 14 + 0.32 * (mu - 1) + 5
    elif mu > 6.5:
        factor = 0.64 * (mu - 1)
        Beff = 19 * ((factor - 1) / (factor**2)) * (Teff / To)**2 + 5

```

```

else:
    Beff = zita
return Beff

def calculate_tsec(To, mu, alpha_post_elastic):
    Tsec = np.sqrt(((To**2) * mu) / (1 + alpha_post_elastic * (mu - 1)))
    return Tsec

def calculate_B_Beff(Beff):
    return 4 / (5.6 - np.log(Beff))

def calculate_M(Teff, Tsec):
    return (Teff / Tsec) ** 2

def calculate_Beff_curve(Sd, Sa, Beff):
    B_Beff = calculate_B_Beff(Beff)
    Sa_Beff = Sa / B_Beff
    return Sa_Beff

def calculate_MADRS_curve(Sa_Beff, M):
    Sa_MADRS = Sa_Beff * M
    return Sa_MADRS

def find_intersection(Sd, Sa_MADRS, Sd_Tsec, Sa_Tsec):
    def interpolate(x, xp, fp):
        return np.interp(x, xp, fp)

    def binary_search(Sd, Sa_MADRS, Sd_Tsec, Sa_Tsec):
        left, right = 0, len(Sd) - 1
        while left <= right:
            mid = (left + right) // 2
            if interpolate(Sd[mid], Sd_Tsec, Sa_Tsec) < Sa_MADRS[mid]:
                left = mid + 1
            else:
                right = mid - 1

```

```

        return Sd[left], Sa_MADRS[left]

    Sd_int, Sa_int = binary_search(Sd, Sa_MADRS, Sd_Tsec, Sa_Tsec)
    return Sd_int, Sa_int

# Función de fragilidad teórica
def fragilidad_teorica(Sd, theta, beta):
    return norm.cdf((np.log(Sd) - np.log(theta)) / beta)

# Función para calcular la probabilidad
def calcular_probabilidad(casos_excedencia, num_analisis,
    fragilidad_teorica):
    return binom.pmf(casos_excedencia, num_analisis, fragilidad_teorica)

# Función objetivo para la optimización usando lmfit
def funcion_objetivo(params, Sd, casos_excedencia, num_analisis):
    theta = params['theta']
    beta = params['beta']
    if theta < 0.01 or beta < 0.1:
        return np.full_like(Sd, np.inf) # Penalización por valores fuera
de los límites

    fragilidades_teoricas = fragilidad_teorica(Sd, theta, beta)

    ln_probabilidades = np.array([np.log(calcular_probabilidad(e,
num_analisis, ft) + 1e-10) for e, ft in zip(casos_excedencia,
fragilidades_teoricas)])

    penalizacion = np.abs(fragilidades_teoricas[-1] - 1) # Penalización
por no alcanzar un valor de fragilidad teórica cercano a 1

    return -ln_probabilidades + penalizacion # Negativo para maximizar,
sumando la penalización

# Función para optimizar  $\theta$  y  $\beta$  usando lmfit
def optimizar_theta_beta(Sd, casos_excedencia, num_analisis):
    params = Parameters()
    params.add('theta', value=0.05, min=0.01)
    params.add('beta', value=0.45, min=0.1)

    result = minimize(funcion_objetivo, params, args=(Sd,
casos_excedencia, num_analisis), method='leastsq')

```

```

    return result.params['theta'].value, result.params['beta'].value #
Retorna  $\theta$  y  $\beta$  optimizados

def
Desplazamiento_Espectral_Sd_vs_Desplazamiento_Objetivo_dt(puntos_desempeño
, Slight, Moderate, Extensive, Collapse, num_registros):

    plt.figure(figsize=(10, 8))

    plt.title('Desplazamiento Espectral Sd (cm) vs Sd Performance Point
(cm)')

    for Sd, delta_t, archivo in puntos_desempeño:
        plt.plot(Sd, delta_t, 'o')

    # Definir los límites intercambiados
    limites = [Slight, Moderate, Extensive, Collapse]
    nombres_limites = ['Slight', 'Moderate', 'Extensive', 'Collapse']

    for limite, nombre in zip(limites, nombres_limites):
        plt.axhline(y=limite, linestyle='--', label=f'{nombre} -
{limite:.2f} cm')

    plt.xlabel('Desplazamiento Espectral Sd (cm)')
    plt.ylabel('Sd Performance Point (cm)')
    plt.grid(True)
    plt.legend()
    plt.show()

    # Clasificar y contar los casos de daño
    daño_leve = sum(1 for _, delta_t, _ in puntos_desempeño if Slight <
delta_t <= Moderate)

    daño_moderado = sum(1 for _, delta_t, _ in puntos_desempeño if
Moderate < delta_t <= Extensive)

    daño_extenso = sum(1 for _, delta_t, _ in puntos_desempeño if
Extensive < delta_t <= Collapse)

    daño_completo = sum(1 for _, delta_t, _ in puntos_desempeño if delta_t
> Collapse)

    print(f"Daño Leve: {daño_leve}")
    print(f"Daño Moderado: {daño_moderado}")

```

```

print(f"Daño Extenso: {daño_extenso}")
print(f"Daño Completo: {daño_completo}")

# Pedir el valor del intervalo en X
intervalo = float(input("Insertar valor de intervalo: "))
if intervalo < 2:
    intervalo = 2

# Pedir el último valor de análisis
ultimo_valor = float(input("Inserte último valor de análisis: "))

num_intervals = int(np.ceil(ultimo_valor / intervalo))

promedios = []
excedencias_leve = []
excedencias_moderado = []
excedencias_extenso = []
excedencias_completo = []

total_excedencias_leve = 0
total_excedencias_moderado = 0
total_excedencias_extenso = 0
total_excedencias_completo = 0

for i in range(num_intervals):
    lower_bound = i * intervalo
    upper_bound = (i + 1) * intervalo
    interval_points = [Sd for Sd, delta_t, _ in puntos_desempeño if
lower_bound < Sd <= upper_bound]

    if interval_points:
        avg_Sd = np.mean(interval_points) / 100 # Convertir de cm a m
        promedios.append(avg_Sd)

# Casos de Excedencia Leve
if i == 0:

```

```

        excedencia_points_leve = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Slight < delta_t
<= Moderate]

        excedencias_leve.append(len(excedencia_points_leve) - 1)
        total_excedencias_leve = excedencias_leve[-1]
    else:
        excedencia_points_leve = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Slight < delta_t
<= Moderate]

        if len(excedencia_points_leve) > 0:
            excedencias_leve.append(total_excedencias_leve +
len(excedencia_points_leve))
            total_excedencias_leve = excedencias_leve[-1]
        else:
            excedencias_leve.append(daño_leve)

# Casos de Excedencia Moderado
if i == 0:
    excedencia_points_moderado = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Moderate < delta_t
<= Extensive]

excedencias_moderado.append(len(excedencia_points_moderado))
    total_excedencias_moderado = excedencias_moderado[-1]
else:
    excedencia_points_moderado = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Moderate < delta_t
<= Extensive]

    if len(excedencia_points_moderado) > 0:
        excedencias_moderado.append(total_excedencias_moderado
+ len(excedencia_points_moderado))
        total_excedencias_moderado = excedencias_moderado[-1]
    else:
        excedencias_moderado.append(daño_moderado)

# Casos de Excedencia Extenso
if i == 0:
    excedencia_points_extenso = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Extensive <
delta_t <= Collapse]

    excedencias_extenso.append(len(excedencia_points_extenso))

```



```

        total_excedencias_extenso = excedencias_extenso[-1]
    else:
        excedencia_points_extenso = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and Extensive <
delta_t <= Collapse]
        if len(excedencia_points_extenso) > 0:
            excedencias_extenso.append(total_excedencias_extenso +
len(excedencia_points_extenso))
            total_excedencias_extenso = excedencias_extenso[-1]
        else:
            excedencias_extenso.append(daño_extenso)

# Casos de Excedencia Completo
if i == 0:
    excedencias_completo.append(0)
else:
    excedencia_points_completo = [Sd for Sd, delta_t, _ in
puntos_desempeño if lower_bound < Sd <= upper_bound and delta_t >
Collapse]
    if len(excedencia_points_completo) > 0:
        total_excedencias_completo +=
len(excedencia_points_completo)
excedencias_completo.append(total_excedencias_completo)
    else:
        if i == 1:
            excedencias_completo.append(0)
        elif i == num_intervals - 1:
            excedencias_completo.append(num_registros)
        else:
            excedencias_completo.append(excedencias_completo[-
1])

    else:
        promedios.append((lower_bound + upper_bound) / 200) #
Promedio en el intervalo
        excedencias_leve.append(daño_leve)
        excedencias_moderado.append(daño_moderado)
        excedencias_extenso.append(daño_extenso)

```

```

        if i == num_intervals - 1:
            excedencias_completo.append(num_registros)
        else:
            if i == 1:
                excedencias_completo.append(0)
            else:
                excedencias_completo.append(excedencias_completo[-1])

    print(f"Promedio Intervalo {i + 1} = {promedios[-1]:.4f} m")

# Ajustar todas las listas a la misma longitud
max_length = max(len(promedios), len(excedencias_leve),
len(excedencias_moderado), len(excedencias_extenso),
len(excedencias_completo))

while len(promedios) < max_length:
    promedios.append(np.nan)
while len(excedencias_leve) < max_length:
    excedencias_leve.append(np.nan)
while len(excedencias_moderado) < max_length:
    excedencias_moderado.append(np.nan)
while len(excedencias_extenso) < max_length:
    excedencias_extenso.append(np.nan)
while len(excedencias_completo) < max_length:
    excedencias_completo.append(np.nan)

return promedios, excedencias_leve, excedencias_moderado,
excedencias_extenso, excedencias_completo, daño_leve, daño_moderado,
daño_extenso, daño_completo

def crear_tablas(promedios, excedencias_leve, excedencias_moderado,
excedencias_extenso, excedencias_completo, niveles_dano):
    tablas = {}

    for nivel, casos in niveles_dano.items():
        if casos == 0:
            continue

```

```

    if nivel == 'Leve':
        theta_beta_opt = optimizar_theta_beta(promedios,
        excedencias_leve, casos)
        fragilidades_teoricas = [fragilidad_teorica(Sd,
        theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]
        probabilidades = [calcular_probabilidad(e, casos, ft) for e,
        ft in zip(excedencias_leve, fragilidades_teoricas)]
        ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
        probabilidades]
        data = {
            'Sd (m)': promedios,
            'Número de Análisis': [casos] * len(promedios),
            'Casos de Excedencia': excedencias_leve,
            'Fracción que causa el colapso': [e / casos for e in
            excedencias_leve],
            'Función de fragilidad teórica': fragilidades_teoricas,
            'Probabilidad': probabilidades,
            'LN Probabilidad': ln_probabilidades,
            'θ': theta_beta_opt[0],
            'β': theta_beta_opt[1]
        }
    elif nivel == 'Moderado':
        theta_beta_opt = optimizar_theta_beta(promedios,
        excedencias_moderado, casos)
        fragilidades_teoricas = [fragilidad_teorica(Sd,
        theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]
        probabilidades = [calcular_probabilidad(e, casos, ft) for e,
        ft in zip(excedencias_moderado, fragilidades_teoricas)]
        ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
        probabilidades]
        data = {
            'Sd (m)': promedios,
            'Número de Análisis': [casos] * len(promedios),
            'Casos de Excedencia': excedencias_moderado,
            'Fracción que causa el colapso': [e / casos for e in
            excedencias_moderado],
            'Función de fragilidad teórica': fragilidades_teoricas,
            'Probabilidad': probabilidades,

```

```

        'LN Probabilidad': ln_probabilidades,
        'θ': theta_beta_opt[0],
        'β': theta_beta_opt[1]
    }
elif nivel == 'Extenso':
    theta_beta_opt = optimizar_theta_beta(promedios,
excedencias_extenso, casos)
    fragilidades_teoricas = [fragilidad_teorica(Sd,
theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]
    probabilidades = [calcular_probabilidad(e, casos, ft) for e,
ft in zip(excedencias_extenso, fragilidades_teoricas)]
    ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
probabilidades]
    data = {
        'Sd (m)': promedios,
        'Número de Análisis': [casos] * len(promedios),
        'Casos de Excedencia': excedencias_extenso,
        'Fracción que causa el colapso': [e / casos for e in
excedencias_extenso],
        'Función de fragilidad teórica': fragilidades_teoricas,
        'Probabilidad': probabilidades,
        'LN Probabilidad': ln_probabilidades,
        'θ': theta_beta_opt[0],
        'β': theta_beta_opt[1]
    }
elif nivel == 'Completo':
    theta_beta_opt = optimizar_theta_beta(promedios,
excedencias_completo, casos)
    fragilidades_teoricas = [fragilidad_teorica(Sd,
theta_beta_opt[0], theta_beta_opt[1]) for Sd in promedios]
    probabilidades = [calcular_probabilidad(e, casos, ft) for e,
ft in zip(excedencias_completo, fragilidades_teoricas)]
    ln_probabilidades = [np.log(p) if p > 0 else np.nan for p in
probabilidades]
    data = {
        'Sd (m)': promedios,
        'Número de Análisis': [casos] * len(promedios),
        'Casos de Excedencia': excedencias_completo,

```

```

        'Fracción que causa el colapso': [e / casos for e in
excedencias_completo],
        'Función de fragilidad teórica': fragilidades_teoricas,
        'Probabilidad': probabilidades,
        'LN Probabilidad': ln_probabilidades,
        'θ': theta_beta_opt[0],
        'β': theta_beta_opt[1]
    }
    tablas[nivel] = pd.DataFrame(data)

    # Ajustar último valor de "Casos de Excedencia" para "Daño
Completo" igual al número de análisis
    if nivel == 'Completo' and len(tablas[nivel]) > 0:
        tablas[nivel].at[len(tablas[nivel]) - 1, 'Casos de
Excedencia'] = casos
        tablas[nivel].at[len(tablas[nivel]) - 1, 'Fracción que causa
el colapso'] = 1.0
        tablas[nivel].at[len(tablas[nivel]) - 1, 'Probabilidad'] =
calcular_probabilidad(casos, casos, fragilidades_teoricas[-1])
        tablas[nivel].at[len(tablas[nivel]) - 1, 'LN Probabilidad'] =
np.log(calcular_probabilidad(casos, casos, fragilidades_teoricas[-1]))

    # Calcular la sumatoria de LN Probabilidad
    suma_ln_probabilidad = tablas[nivel]['LN Probabilidad'].sum()

    # Agregar fila de sumatoria al final de la tabla
    suma_fila = {
        'Sd (m)': 'Σ=',
        'Número de Análisis': '',
        'Casos de Excedencia': '',
        'Fracción que causa el colapso': '',
        'Función de fragilidad teórica': '',
        'Probabilidad': '',
        'LN Probabilidad': suma_ln_probabilidad
    }

    tablas[nivel] = pd.concat([tablas[nivel],
pd.DataFrame([suma_fila]), ignore_index=True)

```

```

return tablas

def exportar_a_excel(tablas, nombre_archivo):
    with pd.ExcelWriter(nombre_archivo) as writer:
        for nivel, tabla in tablas.items():
            tabla.to_excel(writer, sheet_name=nivel, index=False)

def graficar_curvas_fragilidad(tablas, niveles_dano, max_Sd):
    max_x = (1.75 * max_Sd)
    x = np.arange(0.001, max_x, 0.001)

    plt.figure(figsize=(12, 10))
    plt.title('Curvas de Fragilidad')
    plt.xlabel('Desplazamiento Espectral Sd (m)')
    plt.ylabel('Probabilidad de Excedencia')
    colores = ['g', 'y', 'b', 'r']

    for nivel, tabla in tablas.items():
        if nivel not in niveles_dano:
            continue

        theta = tabla[' $\theta$ '][0]
        beta = tabla[' $\beta$ '][0]

        plt.plot(x, fragilidad_teorica(x, theta, beta), '-', label=f'Curva
de Fragilidad {nivel}', color=colores.pop(0))

    plt.legend()
    plt.grid(True)
    plt.show()

def main():
    file_path = 'DatosPushOver/PushOver.txt'
    displacements, forces = read_push_over_data(file_path)
    if displacements is None or forces is None:
        return

```

```

plt.figure(figsize=(8, 6))
plt.plot(displacements, forces, marker='o', linestyle='-', color='r',
label='Curva Pushover')
plt.xlabel('Desplazamiento (cm)')
plt.ylabel('Cortante V (Tonf)')
plt.title('Curva Pushover')
plt.legend()
plt.grid(True)
plt.show()

pos = int(input("Ingrese la posición del punto DU-AU: ")) - 1
dy_ay = input("Ingrese las coordenadas iniciales del punto A para la
curva pushover (Dy, Ay): ")
Dy, Ay = map(float, dy_ay.split(', '))

if 0 <= pos < len(displacements):
    area_OBD = calculate_area_under_curve(displacements, forces, pos)
    print(f"Área bajo la curva Pushover: {area_OBD:.2f}")

Du = displacements[pos]
Au = forces[pos]
initial_bilinear_area = bilinear_area_complete(Dy, Ay, Du, Au)
print(f"Área bajo la curva de la Representación bilineal inicial:
{initial_bilinear_area:.2f}")

error = calculate_error(displacements, forces, Dy, Ay, Du, Au)
print(f"Error de la representación bilineal inicial: {error:.2f}%")

calcular_nuevas_coordenadas = input("¿Calcular nuevas coordenadas para
el punto A? (si/no): ").strip().lower()

if calcular_nuevas_coordenadas == "si":
    while error > 1:
        Dy *= 0.99
        Ay *= 0.99
        error = calculate_error(displacements, forces, Dy, Ay, Du, Au)

```

```

        print(f"Nuevas coordenadas del punto A: Dy = {Dy:.2f}, Ay =
{Ay:.2f}")

        print(f"Área bajo la curva de la Representación bilineal con el
punto A encontrado: {bilinear_area_complete(Dy, Ay, Du, Au): .2f}")

        print(f"Error de la representación bilineal: {error:.2f}%")

plt.figure(figsize=(8, 6))

plt.plot(displacements, forces, marker='o', linestyle='-', color='r',
label='Curva Pushover')

plt.plot(0, 0, marker='o', color='black')
plt.text(0, 0, '0', ha='right', color='black')

if 0 <= pos < len(displacements):
    plt.plot(displacements[pos], forces[pos], marker='x',
markersize=8, color='blue')

    plt.plot([displacements[pos], displacements[pos]], [0,
forces[pos]], 'b--')

    plt.plot([0, displacements[pos]], [forces[pos], forces[pos]], 'b--
')

    plt.text(displacements[pos], 0, 'Du', ha='center', color='blue',
va='bottom')

    plt.text(0, forces[pos], 'Au', ha='right', color='blue',
va='center')

    plt.plot(displacements[pos], forces[pos], marker='o',
markersize=8, color='red')

    plt.text(displacements[pos], forces[pos], 'B', ha='left',
color='red', va='bottom')

plt.plot(Dy, Ay, marker='o', markersize=8, color='green')
plt.text(Dy, Ay, 'A', ha='left', color='green', va='bottom')
plt.plot([0, Dy], [0, Ay], 'g--')

if 0 <= pos < len(displacements):
    plt.plot([Dy, displacements[pos]], [Ay, forces[pos]], 'g--')

plt.plot([0, Dy, displacements[pos]], [0, Ay, forces[pos]], 'g--',
label='Representación Bilineal')

plt.xlabel('Desplazamiento (cm)')

```



```

plt.ylabel('Cortante V (Tonf)')
plt.title('Curva Pushover')
plt.legend()
plt.grid(True)
plt.show()

mass = float(input("Ingrese la masa del sistema (Ton): "))
alpha = float(input("Ingrese el factor de participación modal ( $\alpha$ ): "))
Pf = float(input("Ingrese el factor de forma de la estructura (Pf):
"))
zita = 0.05 # Amortiguamiento

Sd, Sa = convert_to_adrs(displacements, forces, mass, alpha, Pf)
Sd_bilinear, Sa_bilinear = convert_to_adrs(np.array([0, Dy, Du]),
np.array([0, Ay, Au]), mass, alpha, Pf)

plt.figure(figsize=(8, 6))
plt.plot(Sd, Sa, marker='o', linestyle='-', color='b', label='Curva
Capacidad ADRS')
plt.plot(Sd_bilinear, Sa_bilinear, marker='x', linestyle='--',
color='r', label='Representación Bilineal ADRS')

plt.xlabel('Desplazamiento Espectral Sd (cm)')
plt.ylabel('Aceleración Espectral Sa (g)')
plt.title('Representación Bilineal ADRS')
plt.legend()
plt.grid(True)
plt.show()

alpha_post_elastic =
calculate_post_elastic_rigidity_adrs(Sd_bilinear[1], Sa_bilinear[1],
Sd_bilinear[-1], Sa_bilinear[-1])
print(f"Rigidez post-elástica: {alpha_post_elastic:.2f}")
ductility = calculate_ductility(Sd_bilinear)
print(f"Ductilidad última esperada ( $\mu$ ): {ductility:.2f}")

To = calculate_initial_period(Sd_bilinear, Sa_bilinear)
print(f"Periodo Inicial ( $T_0$ ): {To:.2f}")

```

```

resultados = {
    'Ductilidad  $\mu$ ': [],
    'Periodo Efectivo  $T_{eff}$ ': [],
    'Amortiguamiento Efectivo  $\beta_{eff}$ ': [],
    'Coeficiente M': [],
    'Periodo Secante  $T_{sec}$ ': [],
    'B': [],
    'Rigidez Post-Elástica': [],
    'Du': [],
    'Au': [],
    'Reduction(M/B)': []
}

ductilidades = np.arange(1, ductility + 0.5, 0.5)
if ductilidades[-1] > ductility:
    ductilidades[-1] = ductility

for mu in ductilidades:
    Teff = calculate_teff(mu, To)
    Beff = calculate_beff(mu, zita, Teff, To)
    Tsec = calculate_tsec(To, mu, alpha_post_elastic)
    M = calculate_M(Teff, Tsec)
    B = calculate_B_Beff(Beff)

    resultados['Ductilidad  $\mu$ '].append(mu)
    resultados['Periodo Efectivo  $T_{eff}$ '].append(Teff)
    resultados['Amortiguamiento Efectivo  $\beta_{eff}$ '].append(Beff)
    resultados['Coeficiente M'].append(M)
    resultados['Periodo Secante  $T_{sec}$ '].append(Tsec)
    resultados['B'].append(B)

    Du_temp = mu * Sd_bilineal[1]
    Au_temp = np.interp(Du_temp, Sd_bilineal, Sa_bilineal)

```

```

    if mu == 1:
        rigidez_post_elastica = alpha_post_elastic
    else:
        rigidez_post_elastica =
calculate_post_elastic_rigidity_adrs(Sd_bilineal[1], Sa_bilineal[1],
Du_temp, Au_temp)
        if rigidez_post_elastica < 0:
            rigidez_post_elastica = 0

resultados['Rigidez Post-Elástica'].append(rigidez_post_elastica)
resultados['Du'].append(Du_temp)
resultados['Au'].append(Au_temp)
resultados['Reduction(M/B)'].append(M / B)

df_resultados = pd.DataFrame(resultados)
print(df_resultados)

df_resultados.to_excel('Resultados_Ductilidad.xlsx', index=False)

ver_graficas = input("¿Desea ver las gráficas de puntos de desempeño
para cada registro sísmico? (si/no): ").strip().lower()

Sd_values = []
Sd_performance_points = []
puntos_desempeño = []

if ver_graficas == "si":
    carpeta = "Registros Sismicos"
    archivos = [f for f in os.listdir(carpeta) if f.endswith('.txt')]

    colors = cm.rainbow(np.linspace(0, 1, len(ductilidades))) #
Asignar colores únicos

for archivo in archivos:
    try:
        plt.figure(figsize=(8, 6))
        ruta_archivo = os.path.join(carpeta, archivo)

```

```

        tg, ug, Tn, ESPECTD, ESPECTA =
procesar_archivo(ruta_archivo)

        if tg is None:
            continue

        Sd_intersections = []
        Sa_intersections = []
        intersection_found = False
        Sd_star, Sa_star = None, None

        for color, mu, M, Tsec, reduction in zip(colors,
ductilidades, df_resultados['Coeficiente M'], df_resultados['Periodo
Secante Tsec'], df_resultados['Reduction(M/B)']):
            Teff = calculate_teff(mu, To)
            Beff = calculate_beff(mu, zita, Teff, To)
            Sa_MADRS = ESPECTA * reduction

            Sd_MADRS_limited = np.clip(ESPECTD * 1000, 0, 10)
            Sa_MADRS_limited = np.clip(Sa_MADRS, 0, 0.5)

            plt.plot(Sd_MADRS_limited, Sa_MADRS_limited,
label=f'Espectro MADRS Reducido ( $\mu$ ={mu}, M={M:.2f})', color=color)

            Sd_Tsec = np.linspace(0, 10, 100)
            Sa_Tsec = ((4 * np.pi**2 / Tsec**2) * Sd_Tsec) / 1000
            Sa_Tsec_limited = np.clip(Sa_Tsec, 0, 0.5)
            plt.plot(Sd_Tsec, Sa_Tsec_limited, linestyle='--',
color=color, label=f'Línea Tsec ( $\mu$ ={mu}, Tsec={Tsec:.2f})')

            # Encontrar la intersección
            Sd_int, Sa_int = find_intersection(Sd_MADRS_limited,
Sa_MADRS_limited, Sd_Tsec, Sa_Tsec)

            if Sd_int is not None and Sa_int is not None:
                Sd_intersections.append(Sd_int)
                Sa_intersections.append(Sa_int)
                plt.plot(Sd_int, Sa_int, 'ko') # Punto de
intersección

```

```

        if not intersection_found and Sd_int <= Sd.max()
and Sa_int <= Sa.max():
            Sd_star, Sa_star = Sd_int, Sa_int
            intersection_found = True

# Dibujar la línea que une los puntos de intersección
if Sd_intersections and Sa_intersections:
    plt.plot(Sd_intersections, Sa_intersections, 'k-',
label='Function for Performance')

Sd_limited = np.clip(Sd, 0, 10)
Sa_limited = np.clip(Sa, 0, 0.5)

plt.plot(Sd_limited, Sa_limited, marker='o', linestyle='-',
, color='b', label='Curva Capacidad ADRS')

if Sd_star is not None and Sa_star is not None:
    plt.plot(Sd_star, Sa_star, marker='*', markersize=15,
color='orange', label='Intersección Principal')
else:
    Sd_star = Sd[-1]
    Sa_star = Sa[-1]
    plt.plot(Sd_star, Sa_star, marker='*', markersize=15,
color='orange', label='Asumido en el último punto')

plt.xlim(0, 10)
plt.ylim(0, 0.5)

plt.xlabel('Sd [cm]')
plt.ylabel('Sa [g]')
plt.title(f'Punto de desempeño - {archivo[:-4]}')
plt.grid(True)
plt.tight_layout()
plt.show()

Sd_T0 = np.interp(To, Tn, ESPECTD * 1000)
Sd_values.append(Sd_T0)

```

```

        Sd_performance_points.append(Sd_star)
        puntos_desempeño.append((Sd_T0, Sd_star, archivo))

    except Exception as e:
        print(f"Error procesando archivo {archivo}: {e}")

else:
    carpeta = "Registros Sismicos"
    archivos = [f for f in os.listdir(carpeta) if f.endswith('.txt')]

    for archivo in archivos:
        try:
            ruta_archivo = os.path.join(carpeta, archivo)
            tg, ug, Tn, ESPECTD, ESPECTA =
procesar_archivo(ruta_archivo)

            if tg is None:
                continue

            Sd_intersections = []
            Sa_intersections = []
            intersection_found = False
            Sd_star, Sa_star = None, None

            for mu, M, Tsec, reduction in zip(ductilidades,
df_resultados['Coeficiente M'], df_resultados['Periodo Secante Tsec'],
df_resultados['Reduction(M/B)']):
                Teff = calculate_teff(mu, To)
                Beff = calculate_beff(mu, zita, Teff, To)
                Sa_MADRS = ESPECTA * reduction

                Sd_MADRS_limited = np.clip(ESPECTD * 1000, 0, 10)
                Sa_MADRS_limited = np.clip(Sa_MADRS, 0, 0.5)

                Sd_Tsec = np.linspace(0, 10, 100)
                Sa_Tsec = ((4 * np.pi**2 / Tsec**2) * Sd_Tsec) / 1000
                Sa_Tsec_limited = np.clip(Sa_Tsec, 0, 0.5)

```

```

        # Encontrar la intersección
        Sd_int, Sa_int = find_intersection(Sd_MADRS_limited,
Sa_MADRS_limited, Sd_Tsec, Sa_Tsec)
        if Sd_int is not None and Sa_int is not None:
            Sd_intersections.append(Sd_int)
            Sa_intersections.append(Sa_int)
            if not intersection_found and Sd_int <= Sd.max()
and Sa_int <= Sa.max():
                Sd_star, Sa_star = Sd_int, Sa_int
                intersection_found = True

        if not intersection_found:
            Sd_star = Sd[-1]
            Sa_star = Sa[-1]

        Sd_T0 = np.interp(To, Tn, ESPECTD * 1000)
        Sd_values.append(Sd_T0)
        Sd_performance_points.append(Sd_star)
        puntos_desempeño.append((Sd_T0, Sd_star, archivo))

    except Exception as e:
        print(f"Error procesando archivo {archivo}: {e}")

    if Sd_values and Sd_performance_points:
        plt.figure(figsize=(8, 6))
        plt.plot(Sd_values, Sd_performance_points, 'bo', linestyle='None',
marker='o')
        plt.xlabel('Desplazamiento Espectral Sd (cm)')
        plt.ylabel('Sd Performance Point (cm)')
        plt.title('Desplazamiento Espectral Sd (cm) vs Sd Performance
Point (cm)')
        plt.grid(True)
        plt.tight_layout()
        plt.show()

# Calcular umbrales de daño usando Dy y Au

```

```

Slight = 0.7 * Dy
Moderate = Dy
Extensive = Dy + 0.25 * (Du - Dy)
Collapse = Du

# Graficar nueva gráfica "Desplazamiento Espectral Sd vs Sd
Performance Point"

promedios, excedencias_leve, excedencias_moderado,
excedencias_extenso, excedencias_completo, daño_leve, daño_moderado,
daño_extenso, daño_completo =
Desplazamiento_Espectral_Sd_vs_Desplazamiento_Objetivo_dt(puntos_desempeño
, Slight, Moderate, Extensive, Collapse, len(puntos_desempeño))

# Definir los casos totales por nivel de daño
niveles_dano = {
    'Leve': daño_leve,
    'Moderado': daño_moderado,
    'Extenso': daño_extenso,
    'Completo': daño_completo
}

tablas = crear_tablas(promedios, excedencias_leve,
excedencias_moderado, excedencias_extenso, excedencias_completo,
niveles_dano)

exportar_a_excel(tablas, 'Curvas de Fragilidad.xlsx')
print("Las tablas se han exportado a 'Curvas de Fragilidad.xlsx'.")

# Graficar curvas de fragilidad
graficar_curvas_fragilidad(tablas, niveles_dano, max(promedios))

if __name__ == "__main__":
    main(

```




DECLARACIÓN Y AUTORIZACIÓN

Nosotros, **Ajila Ajila Mateo Sebastián** con C.C: # **0705766459** y **Molina Arias Carlos Alfredo** con C.C: # **0941439051** autores del trabajo de titulación: **Cálculo de curvas de fragilidad mediante código de Python, utilizando el Método de Espectro de Capacidad (CSM)** previo a la obtención del título de **Ingeniero civil** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, a los 3 días del mes de septiembre del año 2024

1. **Ajila Ajila Mateo Sebastián**

f. **Molina Arias Carlos Alfredo**



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TEMA Y SUBTEMA:	Cálculo de curvas de fragilidad mediante código de Python, utilizando el Método de Espectro de Capacidad (CSM)		
AUTORES	Ajila Ajila, Mateo Sebastián y Molina Arias, Carlos Alfredo		
REVISOR/TUTOR	Ing. Hernández Barredo, Jaime Francisco, M. Sc		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Ingeniería		
CARRERA:	Ingeniería Civil		
TITULO OBTENIDO:	Ingeniero Civil		
FECHA DE PUBLICACIÓN:	03 de septiembre de 2024	No. DE PÁGINAS:	156
ÁREAS TEMÁTICAS:	Análisis estructural, Análisis sismorresistente, Ingeniería Forense		
PALABRAS CLAVES/ KEYWORDS:	Riesgo sísmico, punto de desempeño, curva de capacidad, representación bilineal, curvas de fragilidad, umbrales de daño.		
<p>RESUMEN: En el presente trabajo de titulación se desarrolló la obtención de curvas de fragilidad mediante dos códigos Python, utilizando dos metodologías, una propuesta por ASCE 41-17 conocida como el Método de los coeficientes y la otra propuesta por el FEMA 440, alternativa C, también conocido como "Método del espectro de Capacidad". Ambos métodos son usados para la obtención de puntos de desempeño de una determinada estructura que se desee analizar.</p> <p>En el desarrollo del proyecto se identificó que existe variación en el apartado necesario para la obtención de las curvas de fragilidad, ya que el método de los coeficientes muestra una tendencia lineal creciente, consecuentemente para el método propuesto por el FEMA 440, en el mismo apartado, presenta una mejor dispersión de datos, logrando así, una probabilidad estadística más representativa para cada caso de estudio.</p> <p>La implementación de un código elaborado en Python para la obtención de curvas de fragilidad es una herramienta significativa para el desarrollo y clasificación por demanda sísmica a una determinada curva de capacidad obtenida por su respectiva tipología estructural, por lo tanto, ambos códigos están abiertos a la manipulación, corrección y su respectivo mejoramiento.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/>	SI	<input type="checkbox"/> NO
CONTACTO CON AUTOR/ES:	Teléfono: +593 989528682 +593 989543454		E-mail: mateoajila0705@gmail.com molinaalfredo11@gmail.com
CONTACTO CON LA INSTITUCIÓN (COORDINADOR DEL PROCESO UTE):	Nombre: Clara Catalina Glas Cevallos		
	Teléfono: +593 98 461 6792		
	E-mail: clara.glas@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			