SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE VULNERABILIDAD EN UNA RED DE ENERGÍA ELÉCTRICA ANTE UN EVENTO DISRUPTIVO

JOHN SEBASTIAN GUERRERO PINTO

DIRECTOR:

DARIN JAIRO MOSQUERA PALACIOS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD TECNOLÓGICA

TECNOLOGÍA EN SISTEMATIZACIÓN DE DATOS

BOGOTÁ D.C
2022

SISTEMA DE INFORMACIÓN WEB PARA LA GESTIÓN DE VULNERABILIDAD EN UNA RED DE ENERGÍA ELÉCTRICA ANTE UN EVENTO DISRUPTIVO

JOHN SEBASTIAN GUERRERO PINTO

DIRECTOR:

DARIN JAIRO MOSQUERA PALACIOS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD TECNOLÓGICA

TECNOLOGÍA EN SISTEMATIZACIÓN DE DATOS

BOGOTÁ D.C
2022

Nota de aceptación

_____

Tutor

_____

ING. DARÍN JAIRO MOSQUERA PALACIOS

Jurado

_____

ING. HECTOR ARTURO FLORES

Bogotá D.C 2022

CONTENIDO

1 FASE DE INICIO, PLANEACION Y ORGANIZACIÓN 11
1.1 TITULO 11
1.2 PLANTEAMIENTO DEL PROBLEMA 11
1.2.1 DESCRIPCIÓN 11
1.2.2 FORMULACIÓN DEL PROBLEMA 12
1.3 JUSTIFICACIÓN 12
1.4 OBJETIVOS 12
1.4.1 OBJETIVO GENERAL 12
1.4.2 OBJETIVOS ESPECÍFICOS 12
1.5 MARCO REFERENCIAL 12
1.5.1 MARCO HISTÓRICO 12
1.5.2 ESTADO DE ARTE 14
1.5.3 MARCO TEÓRICO 15
..... 18
1.5.4 MARCO CONCEPTUAL 22
1.5.4.1 INFRAESTRUCTURA CRITICA 22
1.5.4.2 SECTOR ENERGÉTICO 22
1.5.4.3 SISTEMA ELÉCTRICO 23
1.5.4.3.1 GENERACIÓN 23
1.5.4.3.2 TRANSPORTE 23
1.5.4.4 EVENTO DISRUPTIVO 24
1.5.4.5 VULNERABILIDAD 24
1.5.4.6 RESILIENCIA 24
1.5.4.7 INTERDICCIÓN EN SISTEMAS DE POTENCIA 24
1.5.4.8 API REST 25
1.5.4.8.1 MÉTODOS DE UNA API REST 25
1.5.4.8.2 CONTROL DE ERRORES EN LA API 26
1.5.4.9 MEAN STACK 26
1.5.4.10 MATLAB 27
1.5.4.10.1 MATPOWER 27

1.5.4.10.2 MATLAB COMPILER 27
1.5.4.10.3 MATLAB COMPILER SDK 28
1.5.4.10.4 MATLAB PRODUCTION SERVER (MPS) 29
1.5.4.10.4.1 PETICIÓN HTTP SINCRONA 30
1.5.4.10.4.2 PETICIÓN HTTP ASINCRONA 31
1.5.4.10.4.3 ADMINISTRADOR MPS 34
1.5.4.11 JSON 34
1.5.4.12 METAHEURÍSTICA 35
1.5.4.13 ALGORITMO GENÉTICO 35
1.6 MARCO METODOLÓGICO 35
1.6.1 FASE DE INICIO 35
1.6.2 FASE DE ELABORACIÓN 35
1.6.3 FASE DE CONSTRUCCIÓN 36
1.6.4 FASE DE TRANSICIÓN 36
1.7 ALCANCES 36
1.8 DELIMITACIONES 37
1.8.1 TÉCNICA 37
1.8.2 TEMÁTICA 37
1.8.3 GEOGRÁFICA 37
1.8.4 TEMPORAL 37
1.9 FACTIBILIDAD 37
1.9.1 FACTIBILIDAD TÉCNICA 39
1.9.2 FACTIBILIDAD OPERATIVA 40
1.9.3 FACTIBILIDAD LEGAL 40
1.10 CRONOGRAMA 40
2 FASE ELABORACIÓN 41
2.1 IDENTIFICACIÓN DE ACTORES 41
2.1.1 OPERADOR 42
2.1.2 MPS 42
2.1.3 BASE DE DATOS 42
2.2 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES 42
2.3 LISTA DE CASOS DE USO 44
2.4 MODELADO DE CASOS DE USO 45

2.4.1 CASOS DE USO GENERALES 46
2.4.1.1 DOCUMENTACIÓN CASOS DE USO GENERALES 46
2.4.1.1.1 CASO DE USO: INICIO DE SESIÓN 46
2.4.1.1.2 CASO DE USO: ABRIR MENÚ Y CERRAR MENÚ 48
2.4.2 CASO DE USO: ALGORITMO GENÉTICO 49
2.4.3 MODULO ANÁLISIS DE VULNERABILIDAD 50
2.4.4 CASO DE USO: FLUJO OPTIMO DE POTENCIA 51
2.4.5 CASO DE USO: REPORTES 51
2.4.6 MODULO ALGORITMO GENÉTICO 53
2.4.6.1 DOCUMENTACIÓN CASOS DE USO MODULO ALGORITMO GENÉTICO 54
2.4.6.1.1 CASO DE USO: GENERACIÓN 54
2.4.6.1.2 CASO DE USO: SELECCIONAR CASO 55
2.4.6.1.3 CASO DE USO: CASO DE DESARROLLO 56
2.4.6.1.4 CASO DE USO: RD 57
2.4.6.1.5 CASO DE USO: GENERAL 58
2.4.6.1.6 CASO DE USO: GENERAR VECTOR DE INTERDICCIÓN 59
2.4.6.1.7 CASO DE USO: GUARDAR VECTOR DE INTERDICCIÓN 61
2.4.6.1.8 CASO DE USO: RESULTADOS 62
2.4.7 MODULO ANÁLISIS DE VULNERABILIDAD 63
2.4.7.1 DOCUMENTACIÓN CASOS DE USO ANÁLISIS DE VULNERABILIDAD 64
2.4.7.1.1 CASO DE USO: ANÁLISIS 64
2.4.7.1.2 CASO DE USO: SELECCIONAR CASO 64
2.4.7.1.3 CASO DE USO: REALIZAR ANÁLISIS DE VULNERABILIDAD 66
2.4.7.1.4 CASO DE USO: GENERADORES MÁS ATACADOS 68
2.4.7.1.5 CASO DE USO: LINEAS MÁS ATACADAS 68
2.4.7.1.6 CASO DE USO: RECUPERACIÓN 69
2.4.7.1.7 CASO DE USO: ANTIGUO VECTOR 70
2.4.7.1.8 CASO DE USO: NUEVO VECTOR 71
2.4.7.1.9 CASO DE USO: GUARDAR RESULTADOS 72
2.4.8 MODULO FLUJO ÓPTIMO DE POTENCIA 74

2.4.8.1 DOCUMENTACIÓN CASOS DE USO MODULO FLUJO OPTIMO DE POTENCIA 74
2.4.8.1.1 CASO DE USO: BASE 74
2.4.8.1.2 CASO DE USO: SELECCIONAR CASO 75
2.4.8.1.3 CASO DE USO: RD 77
2.4.8.1.4 CASO DE USO: GD 78
2.4.8.1.5 CASO DE USO: RD Y GD 79
2.4.9 MODULO REPORTES 80
2.4.9.1 DOCUMENTACIÓN CASOS DE USO 80
2.4.9.1.1 CASO DE USO: SELECCIONAR CASO 80
2.5 MODELADO DEL NEGOCIO 81
2.5.1 MODULO ALGORITMO GENETICO 81
2.5.1.1 DIAGRAMAS DE ACTIVIDAD 81
2.5.1.2 DIAGRAMA DE SECUENCIA 84
2.5.2 MODULO ANÁLISIS DE VULNERABILIDAD 89
2.5.2.1 DIAGRAMAS DE ACTIVIDAD 89
2.5.2.2 DIAGRAMA DE SECUENCIA 91
2.5.3 MODULO FLUJO OPTIMO DE POTENCIA 94
2.5.3.1 DIAGRAMA DE ACTIVIDAD 94
2.5.3.2 DIAGRAMA DE SECUENCIA 95
2.5.4 MODULO REPORTES 96
2.5.4.1 DIAGRAMA DE ACTIVIDAD 96
2.5.4.2 DIAGRAMA DE SECUENCIA 97
2.6 DIAGRAMA DE CLASES ANGULAR 98
2.7 DIAGRAMA DE PAQUETES 99
2.7.1 CLIENTE 99
2.7.2 MPS 100
2.7.3 BACKEND 100
2.8 DIAGRAMA DE DESPLIEGUE 101
3 FASE DE CONSTRUCCIÓN 102
3.1 ARQUITECTURA DEL SISTEMA 102
3.1.1 CLIENTE 102
3.1.2 SERVIDORES 103

3.1.2.1 MATLAB PRODUCTION SERVER 103
3.1.2.2 BACKEND 104
3.1.2.2.1 RUTAS 104
3.1.2.2.2 ESQUEMAS 104
3.1.2.2.2.1 ESQUEMA PARA MODULO ALGORITMO GENETICO 105
3.1.2.2.2.2 ESQUEMA PARA MODULOS ANALISIS DE VULNERABILIDAD, FLUJO OPTIMO DE POTENCIA Y REPORTES 105
4 FASE DE TRANSICIÓN 106
4.1 MODULO ALGORITMO GENÉTICO 107
4.1.1 CASO DE DESARROLLO 108
4.1.2 RESPUESTA A LA DEMANDA Y GENERAL 109
4.1.3 GENERAR VECTOR DE INTERDICCIÓN 110
4.1.4 RESULTADOS 111
4.2 MODULO ANALISIS DE VULNERABILIDAD 113
4.2.1 ANÁLISIS 113
4.2.2 RECUPERACIÓN 115
4.3 MODULO FLUJO OPTIMO DE POTENCIA 115
4.3.1 BASE 116
4.3.2 GD 117
4.3.3 RD 117
4.3.4 RD Y GD 118
4.4 MODULO REPORTES 118
4.5 CONCLUSIONES 119
5 BIBLIOGRAFÍA 120

LISTA DE ILUSTRACIONES

Ilustración 1. Línea de tiempo de eventos disruptivos en redes eléctricas ---------- 13
Ilustración 2. Sistema de energía eléctrico ---------- 16
Ilustración 3. Interdependencia entre infraestructuras criticas ---------- 16
Ilustración 4. Peligros que afrontan las redes eléctricas ---------- 17
Ilustración 5. Problema de interdicción ---------- 18
Ilustración 6. Red eléctrica siendo atacada ---------- 19
Ilustración 7. Mejor plan de ataque para topología de red de 24 nodos ---------- 20
Ilustración 8. Recursos energéticos distribuidos (DER's) ---------- 21
Ilustración 9. Triangulo de resiliencia ---------- 22
Ilustración 10. Matlab Compiler ---------- 28
Ilustración 11. Matlab Compiler SDK ---------- 29
Ilustración 12. Patrón de integración en MPS ---------- 30
Ilustración 13. Petición sincrónica en MPS ---------- 31
Ilustración 14. Petición asíncrona en MPS ---------- 33
Ilustración 15. Fases OpenUP ---------- 36
Ilustración 16. Cronograma ---------- 41
Ilustración 17. Casos de uso generales ---------- 46
Ilustración 18. Casos de uso modulo algoritmo genético ---------- 53
Ilustración 19. Casos de uso modulo Análisis de vulnerabilidad ---------- 63
Ilustración 20. Casos de uso modulo flujo óptimo de potencia ---------- 74
Ilustración 21. Casos de uso modulo reportes ---------- 80
Ilustración 22. Actividad 1 Modulo algoritmo genético ---------- 82
Ilustración 23. Actividad 2 modulo Análisis de vulnerabilidad ---------- 83
Ilustración 24. Diagrama de secuencia Modulo algoritmo genético ---------- 84
Ilustración 25. Actividad 1 Modulo análisis de vulnerabilidad ---------- 89
Ilustración 26. Actividad 2 modulo Análisis de vulnerabilidad ---------- 90
Ilustración 27. Diagrama de secuencia modulo Análisis de vulnerabilidad ---------- 91
Ilustración 28. Diagrama de actividad modulo flujo óptimo de potencia ---------- 94
Ilustración 29. Diagrama de secuencia modulo Flujo óptimo de potencia ---------- 95
Ilustración 30. Diagrama de actividad modulo Reportes ---------- 96
Ilustración 31. Diagrama de secuencia modulo Reportes ---------- 97
Ilustración 32. Diagrama de clases Angular ---------- 98
Ilustración 33. Diagrama de paquetes de Angular ---------- 99
Ilustración 34. Diagrama de paquetes MPS ---------- 100
Ilustración 35. Diagrama de paquetes servidor base de datos ---------- 100
Ilustración 36. Diagrama de despliegue del sistema ---------- 101
Ilustración 37. Diagrama arquitectura del sistema ---------- 102
Ilustración 38. Instancias del servidor MPS ---------- 103
Ilustración 39. Rutas servidor base de datos (Backend) ---------- 104

Ilustración 40. Esquema base de datos para algoritmo genético ---------- 105
Ilustración 41. Esquema base de datos para módulos análisis de vulnerabilidad, ---------- 105
Ilustración 42. Inicio de sesión ---------- 106
Ilustración 43. Menú de la aplicación ---------- 107
Ilustración 44. Modulo algoritmo genético ---------- 107
Ilustración 45. Características caso de desarrollo ---------- 108
Ilustración 46. Parámetros respuesta a la demanda ---------- 109
Ilustración 47. Generar vector de interdicción ---------- 110
Ilustración 48. Resultados generación vector de interdicción ---------- 111
Ilustración 49. Guardar resultados ---------- 112
Ilustración 50. Modulo Análisis de vulnerabilidad ---------- 113
Ilustración 51. Guardar resultados ---------- 114
Ilustración 52. Métricas de recuperación ---------- 115
Ilustración 53. Escenario Base ---------- 116
Ilustración 54. Escenario GD ---------- 117
Ilustración 55. Escenario RD ---------- 117
Ilustración 56. Escenario RD y GD ---------- 118
Ilustración 57. Modulo reportes ---------- 118

1 FASE DE INICIO, PLANEACION Y ORGANIZACIÓN

1.1 TITULO

Sistema de información web para la gestión de vulnerabilidad en una red de energía eléctrica ante un evento disruptivo.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 DESCRIPCIÓN

Uno de los sectores con mayor vulnerabilidad al momento de presentarse un evento disruptivo es la infraestructura en el sector de la energía, cuya incapacidad o interrupción tiene efectos que debilitan la seguridad nacional y económica y el de la comunidad [1]. Un evento disruptivo tiene impactos en otras infraestructuras criticas como el transporte y las telecomunicaciones [2], por lo cual genera costos elevados, por ejemplo, en suiza 24 horas sin electricidad causa la pérdida de 2 a 4 billones de dólares sobrepasando el PIB.[3]

Las redes eléctricas están sujetas a diferentes eventos disruptivos; pueden ser involuntarios como eventos extremos naturales (huracanes, terremotos, inundaciones) o eventos deliberados como ciberataques o daños físicos a la red [4].Los eventos disruptivos extremos son eventos que pueden provocarse con una baja probabilidad y un alto impacto (HILP) que afectan significativamente el sistema de una red eléctrica, conllevando a que el fluido eléctrico sea interrumpido afectando a miles de usuarios.

Los operadores de red deben detectar los elementos más vulnerables de la red ante múltiples escenarios o eventos de ataque que pueden poner en riesgo la seguridad de la misma, es decir deben garantizar confiabilidad en la recuperación o resiliencia ante cualquier evento disruptivo y poder tomar decisiones. Para lo anterior el operador de red debe tener un sistema de información web que le permita

poder detectar daños en sistema eléctrico al momento de presentarse un evento disruptivo y recuperar el sistema en un tiempo optimo.

1.2.2 FORMULACIÓN DEL PROBLEMA

¿Cómo implementar un sistema de información web que permita al operador de red de distribución tomar decisiones al momento de presentarse un evento disruptivo en un sistema de energía eléctrica?

1.3 JUSTIFICACIÓN

Se justifica el hecho de hacer una plataforma web de análisis de vulnerabilidad en una red eléctrica porque cada vez dependemos más de este crucial servicio el cual es susceptible a tener impactos negativos y esto se da por emergentes amenazas que podrían tener devastadores efectos en otras infraestructuras importantes.

Es necesario por ende tener un sistema que permita al operador de red tomar las mejores decisiones, evaluar los costos y poder minimizar al máximo los mismos; es necesario mejorar la seguridad de una red eléctrica haciendo uso de herramientas y algoritmos que permitan detectar las vulnerabilidades para hacer frente a cualquier ataque y mejorar la resiliencia de una red. A razón, de que los sistemas de redes eléctricas necesitan la digitalización derivada de las tecnologías de la información porque están en constante proceso de transformación.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Realizar una plataforma web para la gestión y el análisis de vulnerabilidad en un sistema de energía eléctrica ante un evento disruptivo

1.4.2 OBJETIVOS ESPECÍFICOS

- Desarrollar los módulos que nos permitan visualizar el análisis de vulnerabilidad en una red eléctrica.
- Diseñar una base de datos no relacional que nos permita guardar y consultar toda la información generada de un sistema eléctrico ante un evento disruptivo.
- Generar por medio de herramientas de visualización la información del análisis de vulnerabilidad en la red eléctrica

1.5 MARCO REFERENCIAL

1.5.1 MARCO HISTÓRICO

Eventos disruptivos han marcado hitos históricos en las redes eléctricas; desastres naturales y causados por humanos han resultado en pérdidas totales de electricidad impactando negativamente la economía y la seguridad de las naciones. Los eventos extremos que ponen en riesgo las redes también vienen aumentando su fuerza y su frecuencia.

A la fecha innumerables casos que han afectado las redes eléctricas se han venido reportando, impactos a las redes por hechos naturales se han presentado con mayor regularidad en 2005 el huracán Katrina, en 2012 en huracán Sandy, en 2016 el huracán hermine, otros devastadores efectos a las redes se han dado con terremotos, en 2018 el terremoto de hokkiado etc.

Históricamente las redes también han sido blanco de ataques causados por el hombre; en california en 2013 un hombre ataco 17 transformadores dejando millonarias pérdidas, en 2015 se tiene el reporte de un ciberataque en el reino unido a una red eléctrica dejando miles de personas afectadas; el ciber ataque más reciente se dio en los estados unidos en 2019 [5], en la ilustración 1 podemos ver algunos de los impactos a infraestructuras eléctricas:

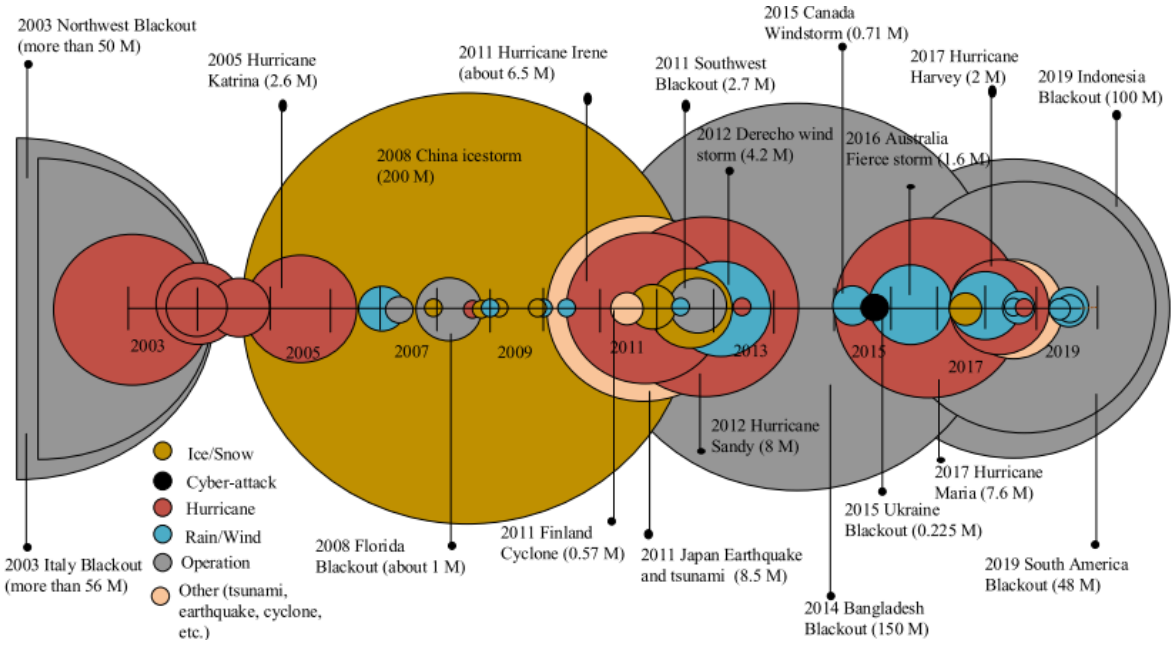


Ilustración 1. Línea de tiempo de eventos disruptivos en redes eléctricas

Fuente: https://www.researchgate.net/figure/Examples-of-high-impact-low-frequency-HILF-events-where-M-denotes-the-number-of_fig1_348891730

Las redes eléctricas han venido mejorando su seguridad y eficiencia, desde los años 70 se han implementado nuevas tecnologías [6], esto porque eventos anteriormente descritos las hacen muy vulnerables por tal motivo es necesario que las redes sean lo más seguras y resilientes posibles.

1.5.2 ESTADO DE ARTE

Los sistemas eléctricos no solo están sujetos a disrupciones eléctricas sino también ataques intencionales. Hay un creciente interés en fortalecer los sistemas eléctricos para que sean más resilientes ante estas disrupciones; hay modelos y algoritmos que predicen peligros, así como estrategias de restauración que han sido desarrollados, sin embargo, hay pocas herramientas Online que ayuden a los operadores a restaurar los sistemas en tiempo real.

El EPRI (Electric Power Research Institute) tiene herramientas de simulación integradas con bases de datos para tener planes de restauración y poder mostrar las consecuencias de un ataque. Sin embargo, a pesar de que estas herramientas están disponibles hay pocos sistemas computarizados web que han sido desarrollados e implementados [7].

Generalmente estos sistemas cuentas con herramientas de optimización no lineal para ayudar a los operadores de red a restaurar el sistema, como lo son métodos heurísticos. También se hace uso de sistemas de software especiales pero que son difíciles de mantener y poco imprácticos en la industria, otros sistemas se basan en redes neuronales y algoritmos genéticos.

En la actualidad los sistemas de energía eléctrica están afrontando una digitalización, la participación de la generación distribuida que utiliza los recursos energéticos distribuidos (DERs) y los modelos de programación binivel para aumentar la resiliencia del sistema son una fuente reciente de estudio y podrían ayudar a mejorar el panorama; de hecho, se ha demostrado que presentan las mejoras respuestas y resultados ante ataques deliberados. Con la presencia de la generación distribuida y modelos propuestos a partir de algoritmos genéticos híbridos permiten concluir que dan soluciones optimas de restauración [8].

Con la creciente tecnología, nuevas iniciativas con respecto la transición del sector energético se ha dado ,un ejemplo de esto lo podemos ver con las tecnologías "smart grid" y son utilizadas para crear plantas de energía virtuales o virtual power plants (VPP) lo cual beneficia al sector en términos de distribución, y manejo eficiente de la energía [9].

El uso de los DER ha puesto en discusión las tradicionales redes eléctricas ya que se presentan como alternativas por su eficiencia y reciente caso de estudio. Los

devastadores efectos naturales hacen cuestionar la resiliencia de estos sistemas tradicionales de energía. Con los DER, smart grid y algoritmos de machine learning es posible mejorar significativamente los eventos disruptivos [10].

Aunque las anteriores tecnologías son las más representativas también es evidente que con ayuda de la tecnología surjan sistemas modernos y potentes como los sistemas cyber-physical power system (CPP) que ayudan a controlar y moldear los ataques híbridos, como ataques cibernéticos a la integridad de la información como los (FDIA) y físicos como ataques terroristas, ambos coordinados y ejecutados [11].

La estabilidad de la red eléctrica no está lo suficientemente entendida aun, la transición y el estudio de las redes eléctricas con la tecnología, las fuentes renovables de energía imponen grandes retos; en general la topología de la red es muy compleja, sin embargo, métodos desarrollados por ingenieros y físicos orientados a incluir métodos no lineales y complejas teorías de redes están prometiendo un avance en las redes eléctricas [12]. Sin embargo, tal transición tecnológica en las redes eléctricas supone más vulnerabilidades porque tales cambios suponen riesgos, también impone retos en ciberseguridad ya que serán más usuales los ciberataques.

1.5.3 MARCO TEÓRICO

El punto de partida inicia en las infraestructuras criticas (CI) las cuales son sistemas de gran escala que nos ofrecen servicios importantes como el agua y la energía, así como transporte banco y la salud. La incapacidad o destrucción de alguna infraestructura tiene un fuerte impacto en la salud, la seguridad y la economía; tienen un impacto tan fuerte en los países que la destrucción de alguna infraestructura critica puede dañar la economía de un país [13]. Las redes eléctricas son las infraestructuras críticas más importantes y enmarcan la investigación y ejecución del proyecto.

En general las redes eléctricas tienen una estructura centralizada; las plantas generan grandes cantidades de electricidad, sus líneas de transmisión comprenden vastas zonas, lo cual por supuesto supone un gran camino donde finalizan en su consumo, en la ilustración 2 podemos ver un ejemplo de un sistema de electricidad desde su generación hasta distribución.

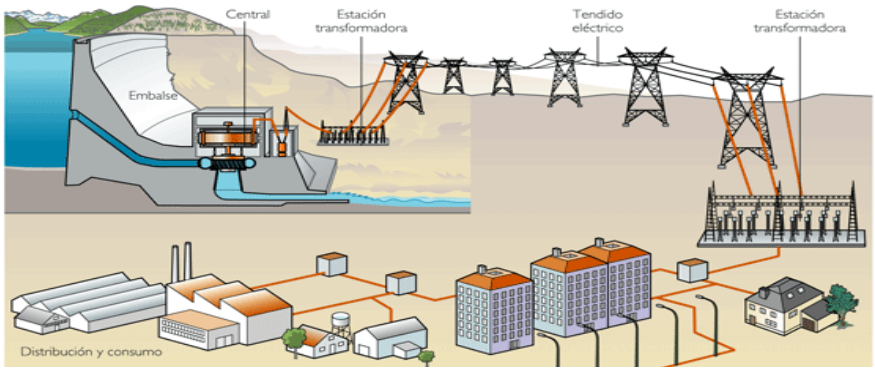


Ilustración 2. Sistema de energía eléctrico

Fuente: https://www.unielectrica.com/agentes-del-mercado/

Importantes infraestructuras criticas dependen de la fiabilidad y una estable provisión de electricidad para que funcionen, por ejemplo, el sector de las comunicaciones, salud transporte depende de la electricidad para; por esto es necesario una continua e ininterrumpida distribución de energía [14]. Esta interdependencia puede mejorar las infraestructuras críticas en términos de eficiencia y operabilidad, sin embargo, esto incrementa las vulnerabilidades de estas infraestructuras, en la ilustración 3 vemos la ligada dependencia que existe entre infraestructuras y como dependen de la electricidad.

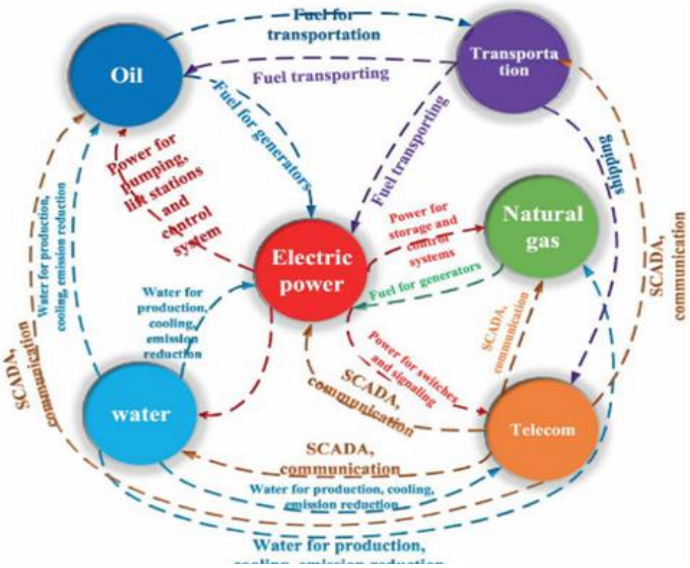


Ilustración 3. Interdependencia entre infraestructuras criticas

Fuente: https://www.researchgate.net/figure/Interdependence-among-critical-infrastructures_fig4_316025063

Las redes eléctricas son vulnerables a diferentes eventos disruptivos; esta vulnerabilidad en los sistemas de electricidad representa una disminución de su funcionalidad y puede conllevar a una total perdida del sistema. Los eventos disruptivos y peligros que afrontan las redes y sistemas eléctricos son:

- Fallas aleatorias: pueden afectar los componentes de la infraestructura pueden considerarse fallas del sistema, sabotajes, como como terrorismo, errores humanos y mal uso del sistema.

- Peligros naturales: este aspecto considera eventos como huracanes, terremotos, vientos etc, este tipo de peligros puede destruir por completo las infraestructuras, es decir puede dañar por ejemplo torres eléctricas.

- Un ataque intencional es un elemento crítico, las redes son atractivas para este tipo de ataques porque en general se distribuyen en largas áreas, elementos como subestaciones están muy concentrados y pueden ser foco de ataques.

En la ilustración 4 podemos evidenciar los peligros que afrontan las redes eléctricas y su probabilidad de ocurrencia:

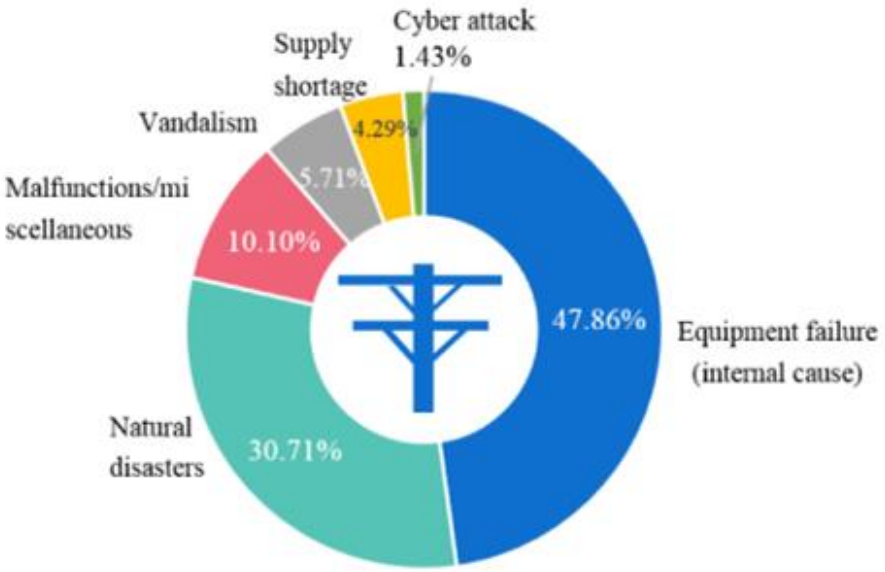


Ilustración 4. Peligros que afrontan las redes eléctricas

Fuente: https://www.researchgate.net/figure/Power-outage-causes-for-140-worldwide-outage-data-from-1965-to-2012_fig1_316025063

Ataques intencionales pueden ser físicos o cibernéticos. Los ataques cibernéticos inyectan datos falsos para burlar el operador del sistema, si un hacker conoce la topología de la red puede alterarla con diferentes flujos de potencia sin que el operador la detecte.

Entre el operador del sistema y el atacante existe un problema conocido como interdicción en la red eléctrica o electric grid interdiction problem (EGIP) está orientado en la interacción de un agente disruptivo y el operador del sistema; consiste en buscar las vulnerabilidades y estrategias de la red eléctrica basándose en restricciones basadas en flujos de potencia, este análisis sirve para que sea más resiliencia la red y que el operador pueda tomar decisiones asertivas al momento de ocurrir un ataque. El operador del sistema está situado en un nivel inferior de optimización del daño del sistema mientras que el agente disruptivo está en el nivel más alto, en la ilustración 5 es posible ilustrar lo anterior.

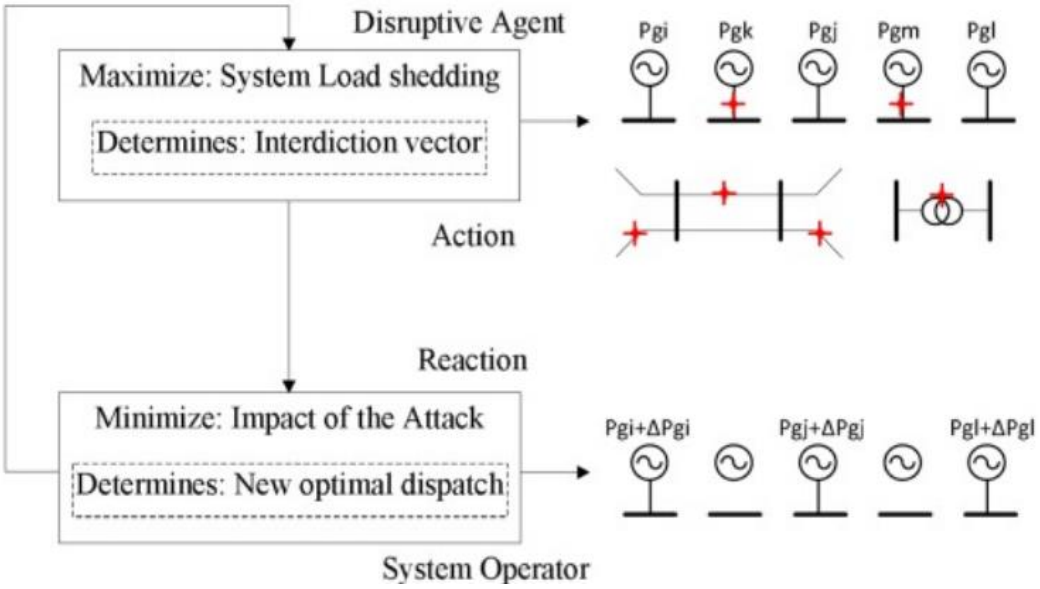


Ilustración 5. Problema de interdicción

Fuente: https://www.sciencedirect.com/science/article/abs/pii/S0378779616305326

En el problema de interdicción se suele usar un modelo DC para versiones simplificadas de red lo cual supone grandes ventajas desde el punto de vista de complejidad, pero no aporta aspectos como magnitudes de voltajes, sin embargo, es usado en la mayoría de los estudios de vulnerabilidad. En el modelo AC de la red

es más preciso, pero más complejo y se suele usar algoritmos metaheurísticos y genéticos para su uso.

Es posible optimizar el EGIP con técnicas heurísticas como los algoritmos genéticos los cuales se basan en métodos naturales, como adaptación del cambio y selección natural. Estas técnicas requieren un vector de interdicción que contiene información del ataque realizado; está compuesto por números que representan las líneas atacadas y que a su vez modifican la topología de una red en la ilustración 6 podemos ver una red siendo atacada con un vector de interdicción binario , los 0 representan las líneas que no están en servicio.

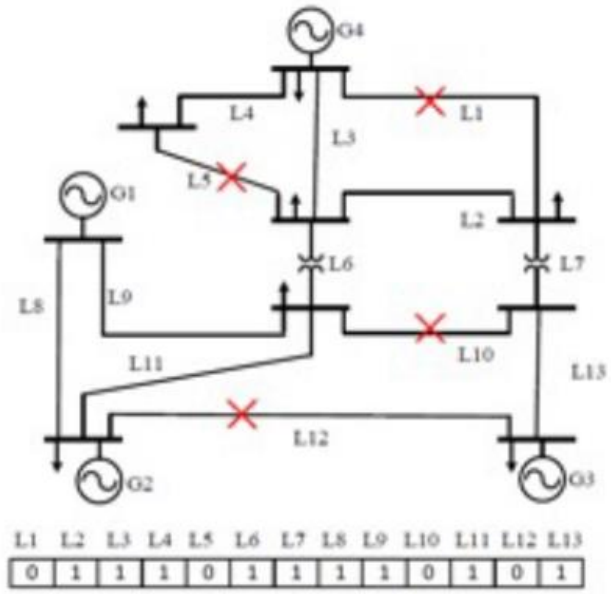


Ilustración 6. Red eléctrica siendo atacada

Fuente: https://www.scielo.cl/scielo.php?pid=S0718-07642018000200073&script=sci_arttext

En el problema, el atacante busca las líneas que mayor daño puede producir a la red y por lo tanto el costo aumenta, por otro lado, el operador del sistema de red debe buscar minimizar los daños producidos y buscar un despacho disponible de energía en general hablamos de un problema de optimización en el cual es posible que el operador del sistema pueda cambiar la topología de la red en pro de reducir costos y ataques intencionales fortaleciendo vulnerabilidades de la red [15].

Con respecto a los operadores del sistema eléctrico, deben modelar múltiples escenarios donde haya escenarios que asimilen la realidad de un evento disruptivo

para reducir la vulnerabilidad ante eventos impredecibles; esto es un importante factor para evaluar la resiliencia de un sistema eléctrico.

Es posible considerar los peores escenarios, es decir los mejores planes de ataque del agente disruptivo causando el mayor deslastre de carga; lo cual da una visión de los costos a considerar, los elementos más vulnerables de la red y la respuesta a la demanda que el sistema puede tener, en la ilustración 7 podemos evidenciar el mejor plan de ataque de un agente disruptor para con una topología inicial de 24 nodos en la cual se desconectaron varios nodos y hubo un impacto negativo de la red.

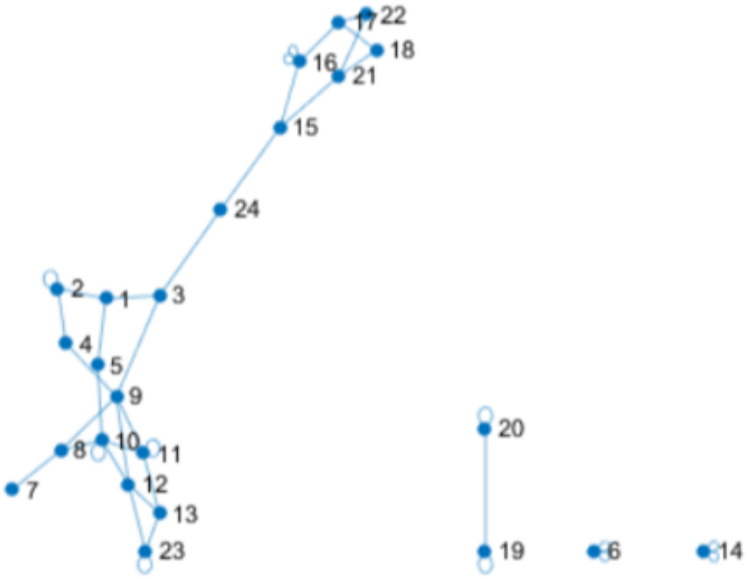


Ilustración 7. Mejor plan de ataque para topología de red de 24 nodos

Fuente: propia

Estos análisis de vulnerabilidad se hacen con el fin de mejorar la fiabilidad y resiliencia de la red. Por un lado, la fiabilidad de las redes eléctricas se da en la medida en que desempeñen correctamente su función en términos de suplir eficientemente la energía a los consumidores; es decir desde el punto de vista de utilidad debe haber suficiente generación, transmisión y distribución de los recursos. Es necesario que las redes mantengan un constante flujo de electricidad incluso ante perturbaciones a la misma red.

Es posible aumentar la fiabilidad del sistema de electricidad utilizando recursos energéticos distribuidos (DER's) como lo pueden ser generación distribuida y tecnologías nuevas como las smart grid para aumentar la fiabilidad del sistema

dando ininterrumpidas fuentes de electricidad para su consumo en la ilustración 8 se puede evidenciar un ejemplo de recurso energético distribuido como lo son paneles solares y recurso eólico para generar energía.



Ilustración 8. Recursos energéticos distribuidos (DER's)

Fuente: https://www.caloryfrio.com/energias-renovables/recursos-energeticos-distribuidos-plan-nacional-integrado-energia-clima.html

Por otro lado, la resiliencia es un término nuevo y largamente investigado; la resiliencia destaca en la habilidad de un sistema y componentes en su recuperación, adaptabilidad al cambio en condiciones extremas, es decir la posibilidad de una rápida recuperación ante un evento disruptivo o un evento de alto impacto poco probable (HILP).

Existen diferentes métodos para evaluar la resiliencia según un evento especifico, también es posible evaluar la resiliencia en múltiples eventos. Los métodos para evaluar la resiliencia de forma cuantitativa deben mostrar las consecuencias de un evento disruptivo incorporando estrategias y planes basados en simulación, probabilísticos y métodos estáticos con el fin de prevenir disrupciones y mantener un estándar de protección del sistema.

En general antes de que un evento disruptivo ocurra el sistema se mantiene en un estado estable, cuando un evento ocurre el sistema entra en un estado de degradación pero intenta mantener la funcionalidad y es donde la resiliencia operativa y de restauración comienza; comienza una operación adaptativa y coordinada de recuperación en la cual se adapta al cambio y comienza una restauración del sistema ante el evento y debe ser tan rápida como sea posible interviniendo la resiliencia de la infraestructura para tener por ultimo un estado operativo funcional , lo anterior lo podemos ver en la ilustración 9.

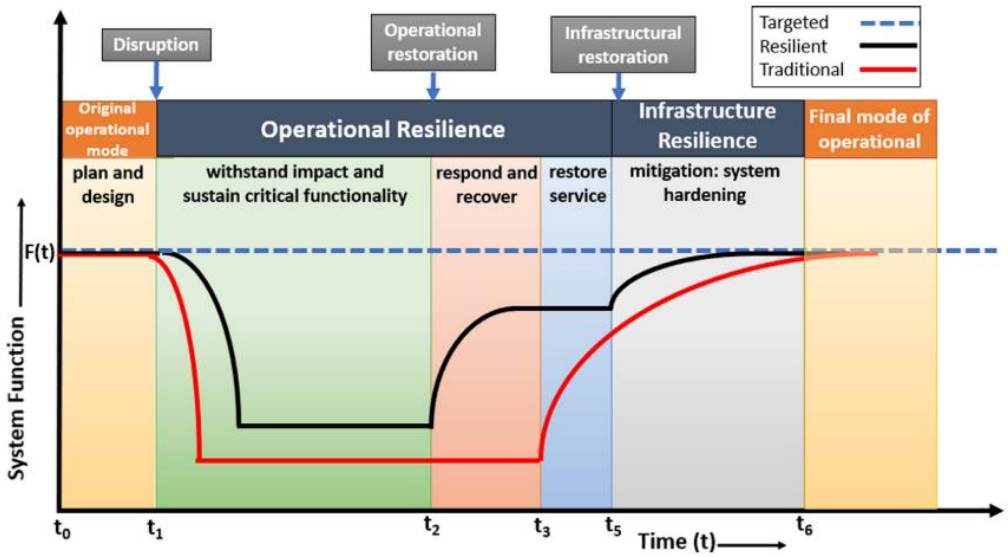


Ilustración 9. Triangulo de resiliencia

Fuente: https://arxiv.org/ftp/arxiv/papers/1910/1910.01234.pdf

1.5.4 MARCO CONCEPTUAL

1.5.4.1 INFRAESTRUCTURA CRITICA

Infraestructura critica son sistemas hechos por el hombre, de gran escala que ofrecen servicios esenciales. Son considerados críticos porque su falla o destrucción tiene impactos considerables en la seguridad, economía y la salud. Dentro de la lista de infraestructuras críticas se encuentran la agricultura, sistemas de comida, bases de defensa, sistemas de energía, salud pública, bancos, comercio, sistemas de emergencia, sistemas nucleares, tecnología, telecomunicaciones y transporte [16]. La falla o destrucción de alguna infraestructura critica puede afectar la funcionalidad de otra ya que están usualmente están relacionadas [17].

1.5.4.2 SECTOR ENERGÉTICO

El sector de la energía es un recurso crucial para la economía y las actividades sociales, es uno de los sectores con mayor desarrollo en los países. Este sector compromete la producción de electricidad a pequeñas y grandes escalas, así como del transporte y otros servicios. Sus usuarios finales incluyen la industria, comercio, actividades de hogar, salud entre otros. La energía puede ser producida a partir de combustibles fósiles, electricidad, y otras fuentes incluyendo tecnologías

renovables. Este sector se encarga de el acceso de fuentes de electricidad, así como de su transporte y distribución efectiva, también se encarga de la producción de energía renovable [18].

1.5.4.3 SISTEMA ELÉCTRICO

El sistema eléctrico es el conjunto de aparatos, barras y líneas que constituyen un circuito. Un sistema eléctrico tiene como finalidad la producción de energía eléctrica a partir de centrales térmicas e hidráulicas hasta su consumo final ya sea en ciudades o centros turísticos. Para tal propósito es necesario tener una correcta capacidad de generación de tal manera que llegue al consumidor.

1.5.4.3.1 GENERACIÓN

La energía se genera en los alternadores con tensiones de entre 3 a 35 kV con corriente alterna en las centrales de generación. Entre las centrales de generadoras podemos encontrar a las hidráulicas, térmicas nucleares, energía solar, eólica, etc [19]. No es posible almacenar la energía eléctrica en grandes cantidades (si en pequeñas como pilas, baterías) esta se debe consumir en el momento en el que se genera por lo cual es necesario recurrir a potencias elevadas para su consumo y demanda.

1.5.4.3.2 TRANSPORTE

La red de transporte se encarga de conectar las redes de distribución con las subestaciones de transformación, de esta manera se asegura el transporte de la electricidad entre puntos muy lejanos y con las menores perdidas. Son las redes eléctricas las encargadas de suministrar el servicio a la mayoría de los habitantes. Estas redes también se les conoce como "redes de alta tensión" y pueden ser subterráneas o aéreas. Las redes de distribución están especialmente diseñadas para que cuando haya una interrupción del suministro sea posible recibir energía procedente de otro punto de producción [20].

Según la tensión que soporte la red eléctrica se puede dividir en tres tipos:

• Líneas de alta tensión (AT): son las que transportan la energía eléctrica con tensiones de 400.000 hasta 30.000 voltios, con el único fin de reducir perdidas de energía en su transporte de largas distancias

• Líneas de media tensión (MT): Son las redes de electricidad que llevan tensiones que van desde los 30.000 hasta los 1000 voltios

• Líneas de baja tensión (BT): llevan la energía a su correspondiente destino para llegue al consumidor final. La tensión de este tipo de red es menor a los 1000 voltios porque los equipos del hogar funcionan con voltajes que están entre los 220 voltios a 380 voltios.

1.5.4.4 EVENTO DISRUPTIVO

Un evento disruptivo tiene potenciales consecuencias drásticas. Los ejemplos más comunes son ataques causados por el hombre, desastres naturales, huracanes, inundaciones etc. En general por eventos naturales extremos. Estos eventos tienen un incremento en su frecuencia, intensidad y duración [21]. Sin embargo, la perdida de información podría ser también un evento disruptivo para una empresa, crisis en la moneda, caída del mercado en general los eventos disruptivos afectan drásticamente el mercado e industria [22].

1.5.4.5 VULNERABILIDAD

la vulnerabilidad es una característica de un sistema la cual causa la pérdida o reducción de la capacidad de realizar una misión o un trabajo como resultado de algún evento definido sea causado por el hombre o por climas extremos o hostiles. En general la vulnerabilidad es definida como la indefensa ante un peligro [23].

1.5.4.6 RESILIENCIA

La resiliencia es la habilidad de volver o recobrar a una forma determinada o posición después de haber sido afectada. En el contexto de sistemas de potencia, la resiliencia refleja la habilidad de resistir los impactos poco probables de alto impacto de una manera eficiente asegurando la menor interrupción posible de electricidad.

La resiliencia de los sistemas se puede conceptualizar teniendo los siguientes aspectos:

• Robustez: es la fuerza y la resistencia para soportar demandas externas sin perder funcionalidad o degradarse.

• Redundancia: propiedad que tiene un sistema para alternar opciones o sustituciones cuando está bajo estrés.

• Rapidez: la velocidad con la que una disrupción se supera y se restaura

1.5.4.7 INTERDICCIÓN EN SISTEMAS DE POTENCIA

Este problema se refiere a la interacción entre un agente o evento disruptivo el cual tiene como objetivo causar el máximo daño al sistema; tiene recursos limitados y debe tener una combinación de elementos que al atacar tengan un daño al sistema, por otro lado, el operador de red debe minimizar estos daños. [24].

1.5.4.8 API REST

Las API son un conjunto de protocolos que se utilizan para diseñar e integrar el software, se pueden implementar de muchas formas. La API REST son los mediadores entre los clientes y los servicios que se desean obtener. Cuando un cliente envía una solicitud por medio de una API REST esta se le transfiere por medio del protocolo de transmisión de la información HTTP en formatos como JSON.[1] Para que una API se considere como REST deberá cumplir los siguientes parámetros:

• Arquitectura cliente-servidor: El servidor controla todos los datos y el cliente controla las interacciones del usuario. Todo se da mediante solicitudes a través de HTTP. La arquitectura cliente servidor está débilmente acoplada ya que el cliente no conoce las implementaciones del servidor y el servidor se encarga de dar acceso a los datos. Lo anterior es un claro ejemplo de separación de funciones basadas en servicios lo que permite la integridad del código y escalabilidad de las aplicaciones.

• Cacheable: debe admitir un sistema de almacenamiento cache lo cual evitará repetir conexiones entre el servidor y el cliente.

Interfaz uniforme: se define una interfaz genérica para administrar cada interacción

que se dé entre el cliente y el servidor esto permite separar la arquitectura.

• uso de URIS: las uris son un identificador uniforme de recursos, el propósito en la API REST es poder interactuar con los recursos que dispone el servidor.

1.5.4.8.1 MÉTODOS DE UNA API REST

Hay métodos importantes que nos permiten interactuar y manipular los datos de una API REST; están soportados por el protocolo HTTP y son usados por la API REST entre los más importantes son:

• **GET**: Usando para obtener un recurso

• **POST**: Crear un recurso nuevo u obtener un recurso

• **PUT**: Modificar un recurso

[1]https://www.redhat.com/es/topics/api/what-is-a-rest-api

• **PATCH**: Modificar un atributo de un recurso

1.5.4.8.2 CONTROL DE ERRORES EN LA API

Las API REST deben manejar los errores que se produzcan deben informar los errores al cliente sobre los recursos que no pudieron ser accedidos. Los errores que son controlados en las API son:

• **200 OK:** Cuando la petición del cliente fue exitosa.

• **201 CREATED:** La petición fue exitosa y se creó un recurso.

• **202 ACEPTED:** Se está procesando la petición, pero no se ha completado.

• **400 BAD REQUEST:** Cuando la sintaxis para hacer la petición no es correcta.

• **403 FORBIDDEN:** Esta respuesta se da al cliente cuando no tiene los permisos para realizar la solicitud.

• **404 NOT FOUND:** Este error ocurre cuando el servidor no encuentra el recurso solicitado.

• **500 INTERNAL SERVER ERROR:** Es un error que se presenta cuando el servidor no tiene los recursos para procesar la solicitud o no está disponible.

1.5.4.9 MEAN STACK

El acrónimo MEAN stack es un reemplazo moderno de LAMP (Linux, Apache,Mysql, PHP /Python) que fue muy popular en anteriores décadas. La particularidad del mean stack es que las aplicaciones son completamente en JavaScript y usan el formato JSON, para crear aplicaciones web dinámicas [2] .

El mean stack se puede resumir de la siguiente manera:

M = MONGODB/Mongoose.js es una base de datos elegante y popular que interactúa muy bien con el formato JSON

E = Express.js es un framework liviano para crear aplicaciones web

A = Angular.js es un framework robusto para crear páginas web HTML5 dinámicas.

[2] https://www.mongodb.com/blog/post/building-your-first-application-mongodb-creating-rest-api-using-mean-stack-part-1

N = Node.js es un intérprete de JavaScript para el lado del servidor

1.5.4.10 MATLAB

MATLAB es un lenguaje de alto nivel que integra computación, visualización y programación en una manera intuitiva donde los problemas y soluciones son expresados en el lenguaje de las matemáticas. Hay muchos proyectos en el mundo que son desarrollados por científicos e ingenieros en Matlab[3].

Matlab tiene una lista interesante de herramientas como lo son:

• Sistemas de control: para diseño test e implementación de sistemas

• Machine learning: para desarrollar algoritmos y modelos

• Deep Learning: para hacer algoritmos con redes neuronales, preparación de datos

• Para redes y comunicaciones: se diseñan test para la verificación de comunicaciones y sistemas

• Procesamiento de imágenes y computación visual: para analizar imágenes mediante diseño y desarrollo de algoritmos

• Robótica: para convertir sistemas autónomos basados en la vida real

1.5.4.10.1 MATPOWER

Matpower es un sistema de energía de simulación basado en Matlab que provee un set de herramientas de alto nivel como flujo de energía, flujo óptimo de potencia y otras herramientas que usan estudiantes e investigadores ya que es una herramienta que permite modelar y formular problemas [4].

1.5.4.10.2 MATLAB COMPILER

Matlab compiler permite utilizar los programas de Matlab como web apps (las aplicaciones se pueden alojar en un host) también es posible empaquetarlos como imágenes Docker. Se pueden crear instaladores para esos programas, usarlos como librerías para otros lenguajes de programación (con sus respectivos formatos sean .py .jar etc), desplegarlos y poder ser usados para integración en otros programas[5]. Los archivos que se generan cuando se empaquetan y se usan para un servicio web tienen el prefijo o extensión. CTF (Component Technology File) el cual se divide en dos partes: un archivo binario que es el main y los demas archivos

[3]https://www.mathworks.com/products/matlab.html
[4]https://matpower.org/
[5]https://www.mathworks.com/products/compiler.html

o librerías que sean necesarios utilizar[6]. En la siguiente ilustración se muestra un ejemplo del aplicativo donde en apps se pueden compilar aplicaciones como escritorio o web, también librerías para empaquetarlas.



Ilustración 10. Matlab Compiler

Fuente: https://la.mathworks.com/products/compiler.html

1.5.4.10.3 MATLAB COMPILER SDK

MATLAB Compiler SDK extiende la funcionalidad de MATLAB Compiler. Incluye herramientas de testing antes de lanzar las aplicaciones en escritorio o web. Las aplicaciones echas en Matlab se pueden compartir gratis con usuarios que no usan Matlab (instalado o como lenguaje de programación). Lo anterior se logra con **Matlab runtime** que es un set de librerías de Matlab que permiten la ejecución de aplicaciones compiladas[7]. Matlab Runtime permite crear y distribuir numerosos aplicativos de software. Los desarrolladores que quieran distribuir su software deben recurrir a Matlab Runtime[8]. Esto para que se pueda compartir y utilizar en internet.

Dentro de Matlab compiler sdk se pueden crear librerías para:

• Aplicaciones escritas en C / C+

• Aplicaciones .NET y COM

• Aplicaciones en Java

[6]https://www.mathworks.com/matlabcentral/answers/93769-how-can-i-distribute-an-application-that-is-developed-using-matlab
[7]https://www.mathworks.com/products/matlab-compiler-sdk.html
[8]https://www.mathworks.com/help/compiler/matlab-runtime.html

• Aplicaciones en Python

• Aplicaciones para desarrolladores que usen servicios WSDL, SOAP , XML, JavaScript, HTML , HTTP services.

Como en la ilustración n el archivo main o función principal (que puede necesitar más archivos para su ejecución) puede ser empaquetado al final en un archivo con lenguajes que usen aplicaciones de escritorio o para servidores.

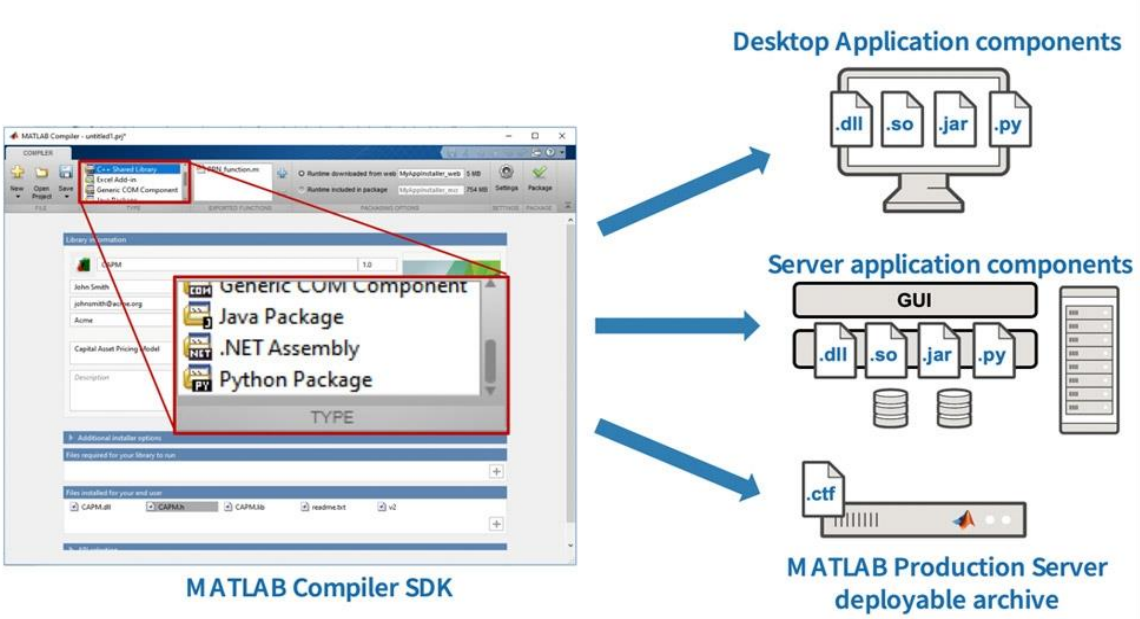


Ilustración 11. Matlab Compiler SDK

Fuente : https://la.mathworks.com/products/matlab-compiler-sdk.html

1.5.4.10.4 MATLAB PRODUCTION SERVER (MPS)

MATLAB PRODUCTION SERVER es una aplicación de servidor el cual publica los algoritmos y aplicaciones como API. Esto permite la interacción con los servicios web RESTful lo cual lo hace útil para aplicaciones empresariales, móvil (iOS, Android, Phonegap, Cordova) y web que pueden integrar: HTML/JavaScript, LAMP, Angular, React, ASP.NET, JSP, JSF y Python.

Dentro de las funcionalidades que extienden a MATLAB PRODUCTION SERVER (MPS) se encuentra que:

• Se le puede dar acceso a los algoritmos a usuarios

• Se puede actualizar fácilmente las versiones de los algoritmos

• Se puede proteger la propiedad intelectual del código

• Se pueden hacer miles de peticiones a los algoritmos

Se ilustra en la figura n que MATLAB PRODUCTION SERVER inicia su ciclo con el programador el cual empaqueta su algoritmo en el Compiler SDK para generar un archivo (.ctf) que sistemas empresariales transaccionales o sistemas basados en REST pueden hacer llamadas HTTP al servidor y puedan usarse como cualquier tipo de aplicación.

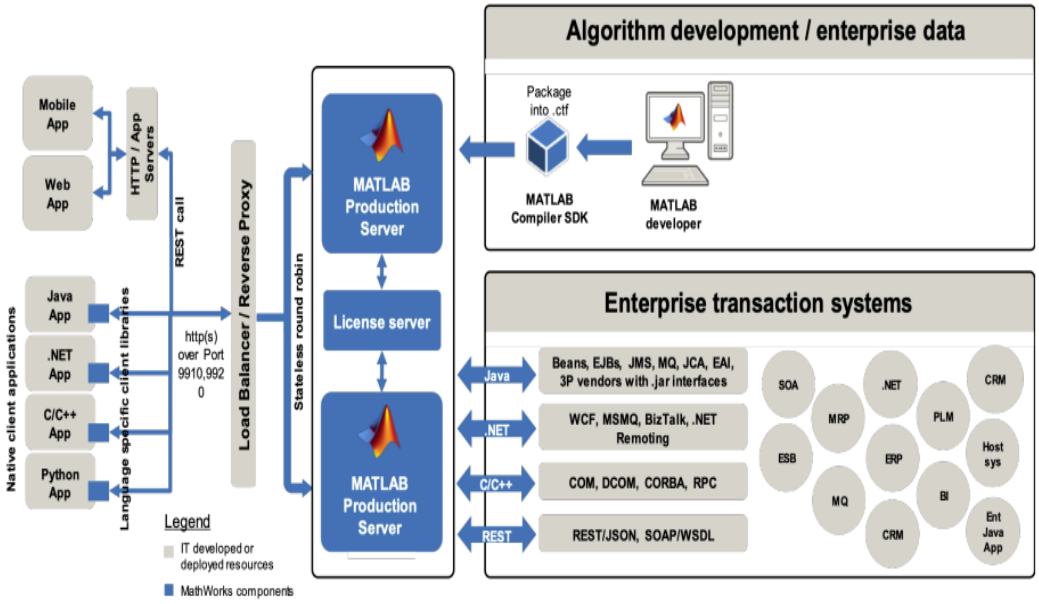


Ilustración 12. Patrón de integración en MPS

Fuente: https://it.mathworks.com/content/dam/mathworks/guide-or-book/matlab-production-server-guide.pdf ,Pag.26

1.5.4.10.4.1 PETICIÓN HTTP SINCRONA

Después de que un cliente hace una petición síncrona se bloquean las demás peticiones hasta que el proceso se haya completado en la petición original. En este tipo de petición automáticamente se envía una respuesta al cliente y se usa por el método HTTP POST. La petición síncrona se realiza usando un cuerpo o body especial, él está formado por los siguientes elementos:

- **nargout:** Número de salidas que el cliente requiere de una función desplegada. Las funciones desplegadas en Matlab pueden retornar uno o más parámetros, por lo tanto, esas salidas se definen en el nargout.

- **rhs:** Son las entradas o parámetros que recibe la función de Matlab, se definen las entradas dentro de un array separado por comas.

- **outputFormat:** especifica si la función debe retornar un JSON en notación larga o pequeña, pero también define si esa respuesta la retorna en un objeto o un string.

-**lhs** : Objeto que retorna MPS al ejecutar una función

En la ilustración n se observa que al hacer una petición síncrona en Matlab producción server a la función mymagic, en su cuerpo espera un resultado y solamente tiene un parámetro , el modo del cuerpo es de tipo corto y de tipo string. La respuesta de mps es un objeto lhs tipo matriz.

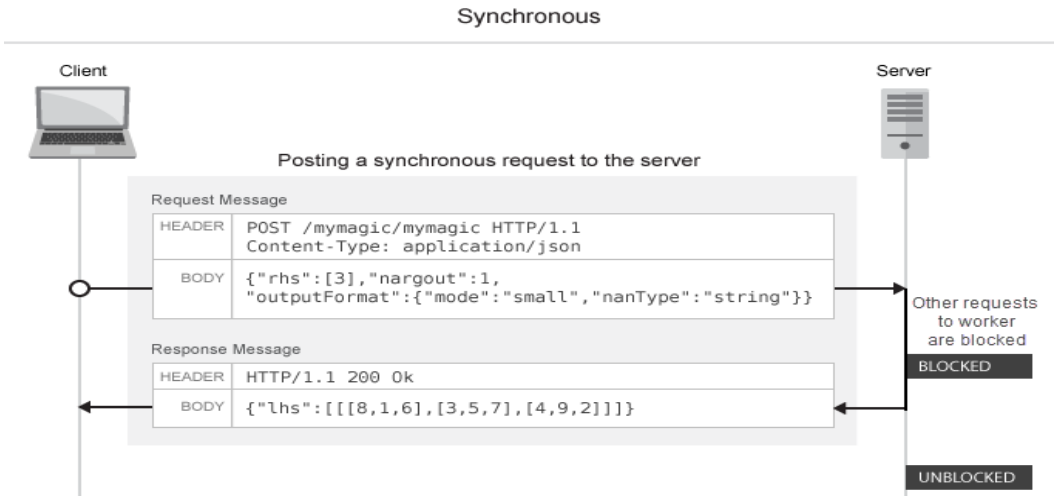


Ilustración 13. Petición sincrónica en MPS

Fuente: https://la.mathworks.com/help/mps/restfuljson/restful-api.html

1.5.4.10.4.2 PETICIÓN HTTP ASINCRONA

La petición asíncrona mientras se procesa deja libre al navegador para que pueda hacer otras peticiones. Las peticiones asíncronas son las más utilizadas en un sitio

web y permiten disminuir la carga en un sitio web.[9] Para realizar una petición asíncrona a Matlab producción server se usa el método HTTP POST.

Durante su ejecución es el primer proceso que se realiza. Así como la petición síncrona la petición asíncrona para Matlab producción server permite definir el cuerpo con rhs, nargout y outputformat.

Sin embargo, realizar una petición asíncrona en mps toma un poco más de trabajo; esta consta de dos procesos.

Cuando se realiza la petición POST la respuesta que recibe el cliente contiene un objeto con los siguientes parámetros los cuales podemos utilizar para hacer otras llamadas al servidor mps[10]:

- **id:** que es un identificador para la petición (con ella se puede cancelar las peticiones al servidor).

- **self:** es la uri de una petición particular con ella se puede retornar el estado de una petición o el resultado que se está buscando de la función desplegada.

- **up:** es una URI que permite ver la colección de peticiones de un cliente.

- lastModifiedSeq: número que indica cuando una petición representada por self fue modificada por última vez

- **state:** estado de una petición

- **client** id del cliente que está realizando la petición se define en la URI como modo opcional

Para obtener el resultado de la función desplegada de Matlab es necesario hacer otra petición HTTP de tipo GET esta vez usando el parámetro self que nos retornó la primera llamada.

Con self podemos consultar el estado de la petición, el resultado y también es posible eliminarla.

Las anteriores acciones siempre nos retornan un JSON tal como se muestra en la figura:

[9]https://cevichejs.com/6-xhr.html
[10]https://www.mathworks.com/help/mps/restfuljson/postasynchronousrequest.html

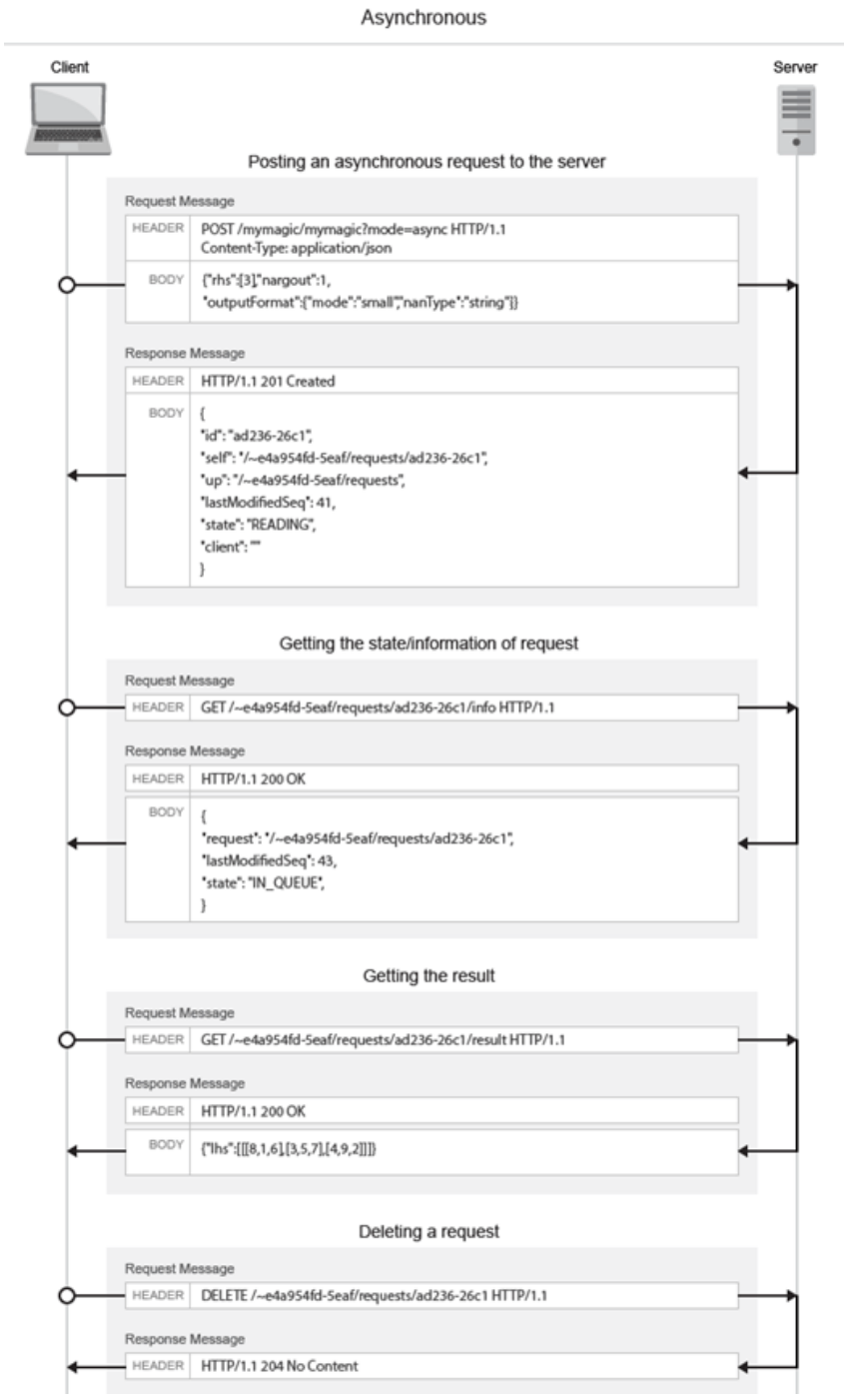


Ilustración 14. Petición asíncrona en MPS

Fuente: https://la.mathworks.com/help/mps/restfuljson/restful-api.html

1.5.4.10.4.3 ADMINISTRADOR MPS

Es el encargado de generar las URI's de la API rest de Matlab mediante las funciones desplegadas en Matlab por medio de Matlab Compiler SDK.

Puede configurar los parámetros del backend necesarios para el desarrollo de la aplicación mediante un dashboard[11]; es el encargado de mantener el sistema de rutas de las aplicaciones, al mismo tiempo gestiona los algoritmos desarrollados por terceros o por el mismo.

Dentro de las funcionalidades que puede hacer el administrador de Matlab producción server son:

- Ingreso al dashboard[12].
- Crear instancias del servidor[13].
- Detener instancias del servidor.
- Reiniciar instancias del servidor.
- Eliminar instancias del servidor
- Visualizar instancias del servidor[14].
- Hacer deploy a las funciones de Matlab[15].
- Hacer undeploy a las funciones de Matlab
- Visualizar logs y errores de las funciones desplegadas16.
- Configurar parámetros del servidor[17].

1.5.4.11 JSON

El acrónimo JSON (JavaScript Object Notation) es un formato liviano de intercambio de datos es muy fácil de usar y de generar. Está formado por dos estructuras, colección de nombre y parejas de valores, se puede usar como un objeto, una

---

[11]Plataforma de información web de monitoreo

[12] mediante un gestor de bases de datos (PostgreSQL) los datos de ingreso al dashboard del administrador son guardados por default para su posterior autenticación y sesión en la plataforma.

[13]Se crean instancias del servidor virtuales (no fisicas)

[14]Se visualizan el estado del procesador, memoria ram y de las peticiones sincronas y asincronas en cola que haga un cliente.

[15]Son desplegadas o lanzadas las funciones y de matlab que contienen los algorimos, como recursos unicos identificables (URI's)

[16]Cuando ocurre un error sea por la función de Matlab desplegada o por sintaxis indecuada para hacer la petición HTTP a la función se visualizan los errores.

[17]Es posible configurar parametros de la instancia del servidor como numero de puerto, licencia , seguridad , CORS etc

estructura, diccionario, o como array; todos los lenguajes de programación soportan estas estructuras de datos [18].

1.5.4.12 METAHEURÍSTICA

Una metaheurística es un método que provee las pautas y estrategias para desarrollar una optimización a un problema [25].

1.5.4.13 ALGORITMO GENÉTICO

Los algoritmos genéticos son una familia computacional de modelos inspirados por la evolución. Tienen potenciales soluciones dentro de problemas específicos, son vistos como optimizadores dentro de una gran gama de problemas. Muchos algoritmos genéticos son introducidos por investigadores que trabajan de una forma experimental y usan estos algoritmos como herramientas de optimización [26].

1.6 MARCO METODOLÓGICO

La metodología que se implementara en el proyecto de es la OpenUP porque es una metodología que da una prioridad inicial a la arquitectura que va a tener la aplicación para minimizar riesgos futuros. Es una metodología que es conveniente ya que gestiona la configuración del cambio, permite el desarrollo de pruebas; utiliza como forma de desarrollo las iteraciones para lograr un producto de calidad controlando la financiación, el riesgo, y los resultados esperados.

Existen 4 fases dentro de la metodología OpenUP[19]:

1.6.1 FASE DE INICIO

se toman en cuenta los objetivos del proyecto para confirmar lo que se debe hacer; se capturan las necesidades obteniendo la mayor información posible.

1.6.2 FASE DE ELABORACIÓN

Se realizan análisis de dominio y definición de la arquitectura se debe tener un prototipo ejecutable de esta, se elaboran planes estableciendo requisitos, uso de las herramientas e infraestructura necesaria de la aplicación.

[18]https://www.json.org/json-en.html
[19]http://openupeaojmp.blogspot.com/2013/09/metodologia-open-up.html

1.6.3 FASE DE CONSTRUCCIÓN

todos los componentes y funcionalidades del sistema se implementan es decir se integran generando y ejecutando resultados rápidamente sin dejar a un lado la calidad del desarrollo.

1.6.4 FASE DE TRANSICIÓN

Esta fase es la introducción de la aplicación cuando esta completada, capacita a los usuarios finales y encargados del mantenimiento del sistema. Permite añadir más funcionalidades o cambios en las entregas finales.

En la ilustración v se ilustra las fases o ciclos de la metodología OpenUP:

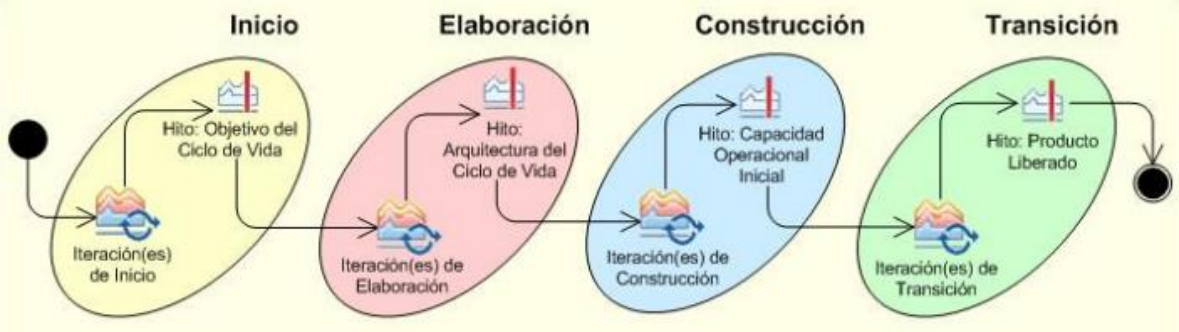


Ilustración 15. Fases OpenUP

Fuente: http://repositorio.espe.edu.ec/bitstream/21000/6316/1/AC-SISTEMAS-ESPE-047042.pdf

1.7 ALCANCES

La finalidad del proyecto es hacer una plataforma web que implemente análisis de vulnerabilidad utilizando el problema de interdicción en las redes eléctricas (EGIP) para evaluar los mejores planes de ataque de un agente disruptivo. Con esto el operador puede tomar decisiones, evaluar costos generar estrategias o planes para gestionar mejor la red eléctrica y en general suministrarle información importante con respecto al nuevo sistema generado ayudando a mejorar la resiliencia del sistema.

1.8 DELIMITACIONES

1.8.1 TÉCNICA

Inicialmente se piensa que la plataforma sea trabajada en MEAN usando el stack completo de JavaScript junto con Matlab. Sin embargo y de acuerdo con la metodología es posible cambiar de arquitectura y de herramientas como java o Python buscando el menor riesgo y tener una factibilidad técnica al finalizar la aplicación.

1.8.2 TEMÁTICA

El desarrollo del proyecto se desarrolla con bibliografía y fuentes científicas que respaldan la temática del proyecto. El semillero de investigación facilita los recursos temáticos que hacen posible una profunda investigación y aporte del tema.

1.8.3 GEOGRÁFICA

El desarrollo de la aplicación se desarrollará de manera estable en instalaciones que no afectan el desarrollo ni la calidad futura del producto.

1.8.4 TEMPORAL

El tiempo para el desarrollo de la aplicación está comprendido entre agosto y diciembre de 2021.

1.9 FACTIBILIDAD

Tabla 1 Factibilidad Económica Recursos Humanos

Tipo	Descripción	Valor-Hora	Cantidad	Total
Tutor 1	**DARIN JAIRO MOSQUERA PALACIOS**	**$ 0**	**6 a 8 horas semanales**	**$ 0**
Desarrolladores	**JOHN SEBASTIAN GUERRERO PINTO**	**$ 0**	**8 horas semanales**	**$ 0**

Total, Recursos Humanos	**$ 0**

En esta tabla se presentan las asesorías que se tendrán y los gastos de los desarrolladores.

En la Tabla 2 se presentarán los gastos y recursos que se necesitan en el desarrollo del proyecto.

Tabla 2 Factibilidad Económica Recursos Técnicos

Recurso	Descripción	Valor Unitario	Cantidad	Total
Computador 1	**Samsung** **MotherBoard A320 Asus** **Ram 8 GB DDR4** **Monitor 22 Pulgadas** **DD 1 TB** **SSD 240 GB**	**$ 1.450.000**	**1**	**$ 1.450.000**
Software	**Matlab**	**$0**	**1**	**$0**
Software	**Visual Studio**	**$0**	**1**	**$0**

Total, Recursos Técnicos	**$ 1.450.000**

Adicionalmente se muestran los gastos adicionales en la Tabla 3 que serán solventados por desarrolladores del proyecto.

Tabla 3 Factibilidad Económica Costo Total

Recurso	Valor
Total, Recursos Humanos	**$0**
Total, Recursos Técnicos	**$1.450.000**
Total, Otros recursos	**$ 50.000**
Costos imprevistos (10%)	**$ 200.000**
TOTAL, COSTO	**$ 1.700.000**

1.9.1 FACTIBILIDAD TÉCNICA

Para el desarrollo de la plataforma web es necesario el uso de las siguientes herramientas:

- Visual Studio
- MongoDB

• Matlab

1.9.2 FACTIBILIDAD OPERATIVA

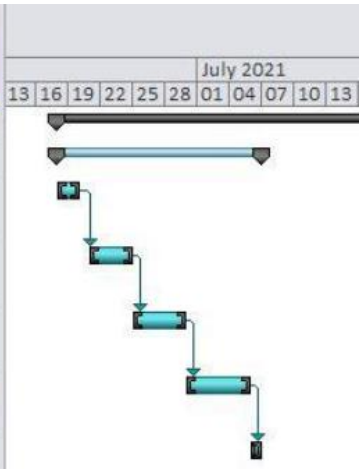
La factibilidad operativa de la plataforma web grande ya que se usarán formatos estables como el JSON herramientas de licencia libre dada por la universidad como Matlab y visual estudio que permite interpretar o compilar cualquier lenguaje de programación; la facilidad de manejar herramientas es grande, por un lado, Matlab es un lenguaje de alto nivel, así como la mayoría de los lenguajes, en el caso del proyecto es JavaScript.

1.9.3 FACTIBILIDAD LEGAL

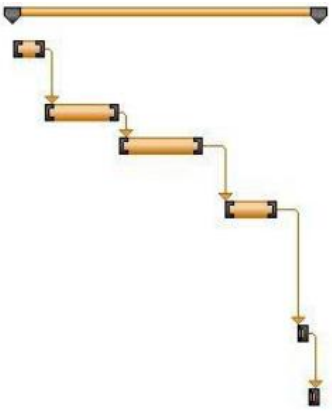
La aplicación se desarrollará con herramientas de libre distribución por lo cual su uso no supone ningún obstáculo en el momento del desarrollo de la aplicación.

1.10 CRONOGRAMA

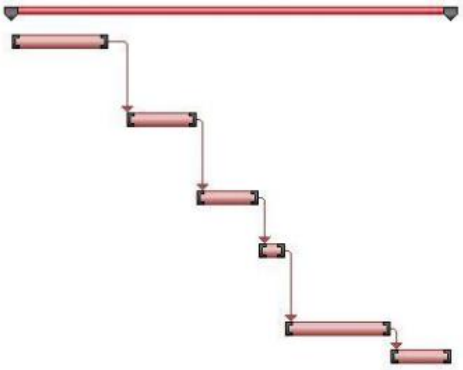
Task Name	Duratic	Start	Finish
Proyecto	**100 days**	**Fri 18/06/21**	**Thu 04/11/21**
Fase de inicio	**13 days**	**Fri 18/06/21**	**Tue 06/07/21**
Identificacion de roles en el desarrollo de la aplicación	2 days	Fri 18/06/21	Sat 19/06/21
Determinacion de requerimientos de la	4 days	Mon 21/06/21	Thu 24/06/21
Entendimiento global de la aplicacion	3 days	Fri 25/06/21	Tue 29/06/21
Identificación de las herramientas de desarrollo	4 days	Wed 30/06/21	Mon 05/07/21
Identificacion de frameworks	1 day	Tue 06/07/21	Tue 06/07/21



Fase de elaboración	**21 days**	**Tue 06/07/21**	**Tue 03/08/21**
Determinación de los posibles modulos de la	3 days	Tue 06/07/21	Thu 08/07/21
Realizar pruebas con Matlab	5 days	Fri 09/07/21	Thu 15/07/21
Realizar pruebas iniciales con arquitectura MEAN	6 days	Fri 16/07/21	Fri 23/07/21
Seleccionar diferentes opciones de arquitectura para la aplicación para minimizar el riesgo	5 days	Mon 26/07/21	Fri 30/07/21
Realizar diagramas de casos de uso	1 day	Mon 02/08/21	Mon 02/08/21
Realizar diagrama de clases	1 day	Tue 03/08/21	Tue 03/08/21



	Fase de construcción	36 days	Tue 03/08/21	Tue 21/09/21
	Tomar como referencia la arquitectura definida y desarrollar los modulos	9 days	Tue 03/08/21	Fri 13/08/21
	Creacion de la base de datos y esquemas necesarios para guardar información	6 days	Mon 16/08/21	Mon 23/08/21
	Creacion de rutas para hacer peticiones	5 days	Tue 24/08/21	Mon 30/08/21
	Completar el desarrollo funcional del servidor con componentes y frameworks	3 days	Tue 31/08/21	Thu 02/09/21
	Uso de Angular para las vistas	8 days	Fri 03/09/21	Tue 14/09/21
	Completar la aplicacion de acuerdo al modelo iterativo	5 days	Wed 15/09/21	Tue 21/09/21



	Fase de transición	32 days	Wed 22/09/21	Thu 04/11/21
	Mejoras visuales de informaci	4 days	Wed 22/09/21	Mon 27/09/21
	Realizar pruebas de funcionamiento	4 days	Tue 28/09/21	Fri 01/10/21
	Correcion de fallas y errores	2 days	Mon 04/10/21	Tue 05/10/21
	Documentar el desarrollo de la aplicación	8 days	Wed 06/10/21	Fri 15/10/21
	Despliegue de la aplicación	4 days	Mon 18/10/21	Thu 21/10/21
	Desarrollo de manual de usuario e introducion a la	4 days	Fri 22/10/21	Wed 27/10/21
	Proponer futuras mejoras y agrandamiento de la	6 days	Thu 28/10/21	Thu 04/11/21

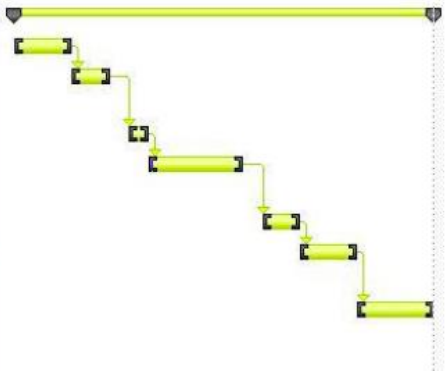


Ilustración 16. Cronograma

Fuente: propia

2 FASE ELABORACIÓN

A continuación, se implementará el modelado de los casos de uso y diagramas de casos de uso que facilitan el levantamiento de requerimientos y análisis de todo el entorno del sistema.

2.1 IDENTIFICACIÓN DE ACTORES

Un actor se entiende como el rol que un usuario y/o sistema realiza en la aplicación. En la aplicación se identifican tres actores que ayudan a entender el sistema por medio de actividades que ejecutan mediante su rol (Dependiendo el caso de uso) y que son relevantes para el sistema[20].

[20] https://es.wikipedia.org/wiki/Actor_(UML)

2.1.1 OPERADOR

Es el actor principal de la aplicación que realiza acciones en la plataforma.

2.1.2 MPS

El rol de Matlab Production Server es ejecutar las funciones y/o algoritmos mediante instancias del servidor dependiendo de los casos de uso en la aplicación.

2.1.3 BASE DE DATOS

La base de datos realiza actividades importantes como guardar y consultar datos dependiendo el caso de uso. Estas actividades permiten que el rol de la base de datos se use como actor externo de la aplicación.

2.2 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Dentro de la tabla se encuentran los requerimientos que pueden ser funcionales o no funcionales; de ella es posible posteriormente hacer una lista de casos de uso que nos ayudan a definir los requerimientos funcionales con los que va a contar la aplicación.

CÓDIGO	REQUERIMIENTO	FUNCIONAL	NO FUNCIONAL	DESCRIPCIÓN
RC01	El sistema debe tener una interfaz gráfica amigable que contenga los módulos de la aplicación.	x		Con angular se espera una plataforma responsiva y adaptable al móvil; que administre las páginas e información para que los operadores hagan uso correcto del sistema.

RC02	El sistema aporta seguridad en la plataforma		x	Se plantea que el sistema tenga autenticación e inicio de sesión para los operadores, sin embargo, no es de vital importancia para el sistema ni sus objetivos.
RC03	El sistema debe tener arquitectura cliente servidor.	x		Se debe implementar esta arquitectura para lograr desacoplar al máximo la aplicación, esto permitirá futuras implementaciones y variación en los lenguajes utilizados.
RC04	El sistema de información permite la ejecución de las funciones de Matlab.	x		Debido a que el sistema usa librerías para el flujo óptimo de potencia propias de Matlab con el uso de Matpower, el sistema deberá poder ejecutar funciones que contienen estos algoritmos.
RC05	El sistema de información permite guardar y consultar los datos.	x		Los datos generados en Matlab deben ser guardados en una base de datos NoSQL Para luego ser consultados, lo que permite ver resultados e información.

RC06	El sistema de información permite visualizar gráficas.	x		Con la ayuda de Angular y datos generados por Matlab se esperan obtener resultados gráficos de los análisis de vulnerabilidad.
RC07	El sistema realiza análisis de vulnerabilidad para todas las topologías de red.		x	Se tienen en cuenta casos iniciales del sistema, se implementarán todos los casos en versiones posteriores del sistema.
RC08	El sistema tiene gran capacidad para ejecutar algoritmos en tiempos óptimos		x	Pueden existir algoritmos que sean lentos en ejecución pero que sus resultados sean útiles. Lo anterior implica que no es indispensable que sean rápidos y que se ejecuten en tiempos óptimos pero que no afecten los análisis de vulnerabilidad.

2.3 LISTA DE CASOS DE USO

ID	CASO DE USO
1	iniciar sesión
2	Abrir menú/cerrar
3	Algoritmo genético
4	Generación

5	Seleccionar caso[21]
6	Caso de desarrollo
7	RD
8	General
9	Generar vector de interdicción
10	Guardar vector de interdicción
11	Resultados
12	Análisis de vulnerabilidad
13	Análisis
14	Realizar análisis de vulnerabilidad
15	Generadores más atacados
16	Líneas más atacadas
17	Recuperación
18	Antiguo vector
19	Nuevo vector
20	Guardar resultados
21	Flujo óptimo de potencia
22	Base
23	RD
24	GD
25	RD Y GD
26	Reportes
27	Cerrar sesión

2.4 MODELADO DE CASOS DE USO

Se definieron 4 módulos principales de la aplicación Algoritmo genético, Análisis de vulnerabilidad , Flujo óptimo de potencia y Reportes ; los casos de uso se utilizan para moldear la lista de casos de uso 2.3 requisitos funcionales , en los módulos principales de la aplicación.

[21] Este caso de uso se reutiliza en todos los módulos y corresponde a una clase de angular que se detallará posteriormente.

2.4.1 CASOS DE USO GENERALES

Se definen como actividades y o acciones básicas que el operador debe realizar para manejar de forma general la aplicación como se muestra en la ilustración 17.

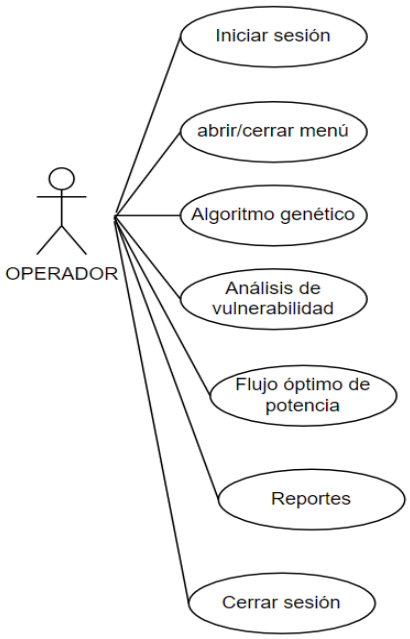


Ilustración 17. Casos de uso generales

Fuente: propia

2.4.1.1 DOCUMENTACIÓN CASOS DE USO GENERALES

2.4.1.1.1 CASO DE USO: INICIO DE SESIÓN

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**1**
ID. Requerimiento(s)	RC02
Nombre Caso de Uso:	Inicio de sesión
Autor:	Desarrollador
Fecha:	Noviembre 2021
Descripción:	Permite al usuario poder iniciar sesión en la plataforma autenticándose con los servicios de Google que ofrece Angular.

Actores:	
1	Operador

Precondiciones:	
1	Tener cuenta de Gmail
2	La cuenta tuvo que haber sido registrada en Google API[22] por un administrador

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón iniciar sesión	Desplegar formulario para inicio de sesión	Dato de Entrada: correo y Contraseña
2		Se validan los datos por medio de una petición a Google API para verificar si la cuenta tiene los permisos. Posteriormente autentica a la cuenta asociada.	
6		Ingreso a la plataforma.	El dato de salida correspond e a un objecto que contiene al

[22]

			token[23], información básica del usuario como foto, nombres y apellidos de usuario y correo.

2.4.1.1.2 CASO DE USO: ABRIR MENÚ Y CERRAR MENÚ

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**2**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Abrir y cerrar menú
Autor:	Desarrollador
Fecha:	Noviembre 2021
Descripción:	Este caso de uso expande los módulos de la aplicación (muestra sus nombres) y/o puede volver a su estado original.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema

[23] Un tipo de serial que confirma la autenticación del usuario

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón para abrir y cerrar menú	Despliega los nombres de los módulos de la aplicación expandiendo la barra lateral o sidenav. Caso contrario la contrae, ocultando los nombres, pero dejando visible los módulos.	

2.4.1.1.3 CASO DE USO: ALGORITMO GENÉTICO

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**3**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Algoritmo genético
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Permite al operador acceder al módulo de algoritmo genético donde podrá realizar operaciones asociadas a esa función específica de Matlab.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1			

	Pulsar botón “Algoritmo genético” de la barra de navegación lateral	Se redireccionar al módulo de algoritmo genético donde se visualiza y se pueden realizar acciones posteriores.	

2.4.1.1.4 MODULO ANÁLISIS DE VULNERABILIDAD

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**12**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Análisis de vulnerabilidad
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde al segundo módulo de la aplicación y al segundo botón del sidebar donde se accede al módulo.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón “Análisis de vulnerabilidad”	Redirecciona a la página del módulo análisis de vulnerabilidad	

2.4.1.1.5 CASO DE USO: FLUJO OPTIMO DE POTENCIA

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**21**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Flujo óptimo de potencia
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde al botón del tercer módulo de la aplicación se ubica en la sidebar o barra lateral
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón “Flujo óptimo de potencia”	Redirecciona al módulo Flujo óptimo de potencia	

2.4.1.1.6 CASO DE USO: REPORTES

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**24**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Reportes

Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde al botón del cuarto módulo de la aplicación Se ubica en la sidebar.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón reportes	Se redirecciona al módulo Reportes	

2.4.1.1.7 CASO DE USO: CERRAR SESIÓN

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**27**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Cerrar sesión
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde al botón para salir sesión
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón salir sesión.	El sistema elimina al usuario autenticado del caché[24]	

2.4.2 MODULO ALGORITMO GENÉTICO

En la ilustración 18 se observan los casos de uso que tiene el módulo de algoritmo genético.

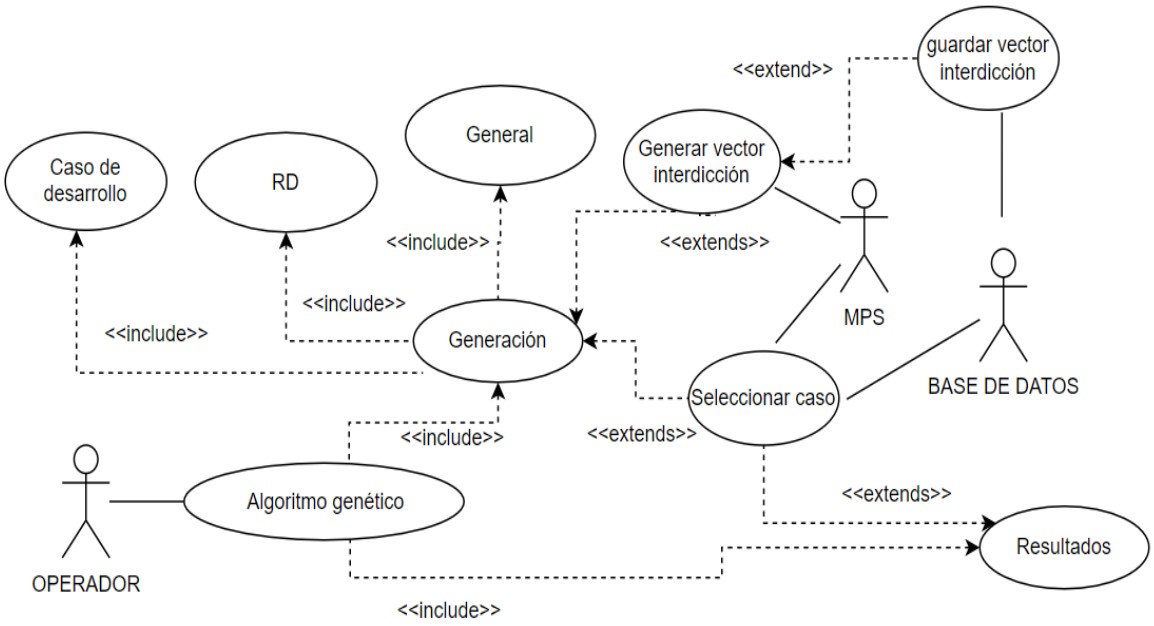


Ilustración 18. Casos de uso modulo algoritmo genético

Fuente: propia

[24] La sesión del usuario se guarda en el caché del sistema de Angular

2.4.2.1 DOCUMENTACIÓN CASOS DE USO MODULO ALGORITMO GENÉTICO

2.4.2.1.1 CASO DE USO: GENERACIÓN

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**4**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Generación
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	El módulo algoritmo genético se divide en dos pestañas generación y resultados. En la pestaña generación se pueden realizar acciones asociadas al módulo.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón de la pestaña “Generación”	Esta pestaña se encuentra en el módulo algoritmo genético	

2.4.2.1.2 CASO DE USO: SELECCIONAR CASO

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**5**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Seleccionar caso
Autor:	Desarrollador
Fecha:	Noviembre 2021 / actualizado diciembre 2022
Descripción:	En el módulo algoritmo se tiene un botón desplegable llamado “Seleccionar caso” que tiene los casos[25] , ejecuta y retorna los datos de una función de Matlab (Características) de acuerdo con cada caso seleccionado, los datos son características del sistema de red eléctrica.
Actores:	
1	Operador
2	MPS

Precondiciones:	
1	Estar autenticado en el sistema
2	Estar en el módulo de algoritmo genético

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador presiona el botón desplegable.	El sistema despliega todos los casos disponibles (topologías de red)	La lista de casos corresponde a un vector de enteros correspondientes al número de nodos de la topología de red.
2	El operador selecciona la topología de red de n nodos.	El sistema realiza una petición POST[26].	El dato de entrada es la topología de red de n nodos.

[25] Topologías de red de 5,6 nodos etc.
[26] Realiza una petición síncrona al servidor MPS 1.5.4.10.4.1

3	MPS ejecuta la función “Características”	El sistema retorna los datos de la función ejecutada donde es posible visualizarlos.	El dato de salida es un objecto [27]que contiene las características de la topología de red (líneas, nodos, cargas) recursos del atacante, respuesta a la demanda.

2.4.2.1.3 CASO DE USO: CASO DE DESARROLLO

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**6**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Caso de desarrollo
Autor:	Desarrollador
Fecha:	Noviembre 2021
Descripción:	Corresponde a un panel desplegable donde se visualiza información del caso, es decir de la topología de red y su configuración.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo algoritmo genético y a la pestaña “Generación”
3	Seleccionar algún caso o topología de red

Flujo Básico o Normal		
Acción del actor	**Acción del sistema**	**Datos de entrada/ Datos de salida**

[27] MPS siempre retorna un objeto llamado “lhs” que contiene los datos que retorna la función

1	Pulsar panel desplegable “Caso de desarrollo”	Despliega panel y muestra información del sistema de red seleccionado, utilizando datos del objeto retornado “lhs” cuando se selecciona caso	

2.4.2.1.4 CASO DE USO: RD

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**7**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	RD
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un panel desplegable donde se visualiza información de valores correspondientes a respuesta a la demanda que se utilizan para cálculos posteriores.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo algoritmo genético y a la pestaña “Generación”
3	Seleccionar algún caso o topología de red

Flujo Básico o Normal		
Acción del actor	**Acción del sistema**	**Datos de entrada/ Datos de salida**

1	Pulsa panel desplegable “RD”	Despliega panel y muestra información de los datos de respuesta a la demanda, utilizando datos del objeto retornado “lhs” cuando se selecciona caso.	

2.4.2.1.5 CASO DE USO: GENERAL

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**8**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	General
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un panel desplegable donde se visualiza información general del sistema de red correspondiente a costos
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo algoritmo genético y a la pestaña “Generación”
3	Seleccionar algún caso o topología de red

Flujo Básico o Normal		
Acción del actor	**Acción del sistema**	**Datos de entrada/ Datos de salida**

1	Pulsar botón desplegable General	Desplegar información del sistema de red :costo de las líneas, costo de los generadores y los recursos que tiene un posible atacante para hacer un ataque.	

2.4.2.1.6 CASO DE USO: GENERAR VECTOR DE INTERDICCIÓN

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**9**
ID. Requerimiento(s)	RC01 Y RC04
Nombre Caso de Uso:	Generar vector de interdicción
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un botón donde dado según la topología de red se genera un vector de interdicción que corresponde a la topología de red dañada por un agente disruptor y que tiene los máximos costos para el sistema
Actores:	
1	Operador
2	MPS

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo algoritmo genético y a la pestaña “Generación”
3	Seleccionar algún caso o topología de red

Flujo Básico o Normal		
Acción del actor	**Acción del sistema**	**Datos de entrada/**

			Datos de salida
1	El operador pulsa el botón de generar vector de interdicción.	La acción del sistema es cambiarse a la pestaña resultados y mientras se obtienen los resultados se muestra en la pantalla un spinner cargador. Internamente el sistema realiza la petición POST.	Los datos de entrada son los que resultan de seleccionar caso es decir los datos que se pueden visualizar en los paneles de los casos de uso “caso de desarrollo”, “RD” y “General”.
2	MPS ejecuta la función de algoritmo genético con los datos de entrada que el sistema ingreso y retorna un objeto	El sistema recibe el objeto “lhs” y los guarda en variables correspondientes para ser visualizados en la pestaña “resultados”. Detiene el spinner y muestra los resultados.	El dato de salida contiene una matriz donde cada fila corresponde a un vector correspondiente a una topología de red dañada. El ultimo vector corresponde al vector de interdicción.

Flujo alternativo			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador pulsa el botón de generar vector de interdicción.	Consulta a la base de datos si ya se ha generado vector de interdicción.	Dato de entrada caso n
2		Si ya se generó un vector de interdicción, sale una ventaja emergente preguntando si quiere ver resultados.	
3	Pulsar ver resultados	El sistema cambia a la pestaña resultados, con los datos.	Objeto que contiene población de

			interdicción y vector de interdicción.

2.4.2.1.7 CASO DE USO: GUARDAR VECTOR DE INTERDICCIÓN

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**10**
ID. Requerimiento(s)	RC01 y RC05
Nombre Caso de Uso:	Guardar vector de interdicción
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a una ventana donde se le pregunta al usuario si desea guardar el vector de interdicción generado
Actores:	
1	Operador Base de datos

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo algoritmo genético y a la pestaña “Generación”
3	Haber seleccionado algún caso o topología de red.
4	Haber ejecutado el algoritmo genético.

Flujo Básico o Normal		
Acción del actor	**Acción del sistema**	**Datos de entrada/**

			Datos de salida
1		Desplegar ventana modal para preguntar si desea guardar el vector de interdicción	
2	Pulsar botón de confirmación	Se realiza una petición POST al backend y se comunica con la base de datos	El dato de entrada es el vector de interdicción y la población de interdicción
	La base de datos guarda en un objeto la población de interdicción y el vector de interdicción.		

2.4.2.1.8 CASO DE USO: RESULTADOS

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**11**
ID. Requerimiento(s)	RC01 Y RC04
Nombre Caso de Uso:	General
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a una pestaña (Tab) donde dé se pueden visualizar resultados obtenidos de vectores y poblaciones de interdicción cuando se genera vector de interdicción.
Actores:	
1	Operador
2	

Precondiciones:	
1	Estar autenticado en el sistema

2	Entrar al módulo algoritmo genético

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador ejecuta botón de generar vector de interdicción.	Cuando el sistema proceso la petición y retorna los datos se cambia la pestaña are	Dato de salida es el vector donde se encuentran las topologías con n número de nodos.

2.4.3 MODULO ANÁLISIS DE VULNERABILIDAD

En el módulo análisis de vulnerabilidad se encuentra los casos de uso asociados como se observa en la ilustración 19.

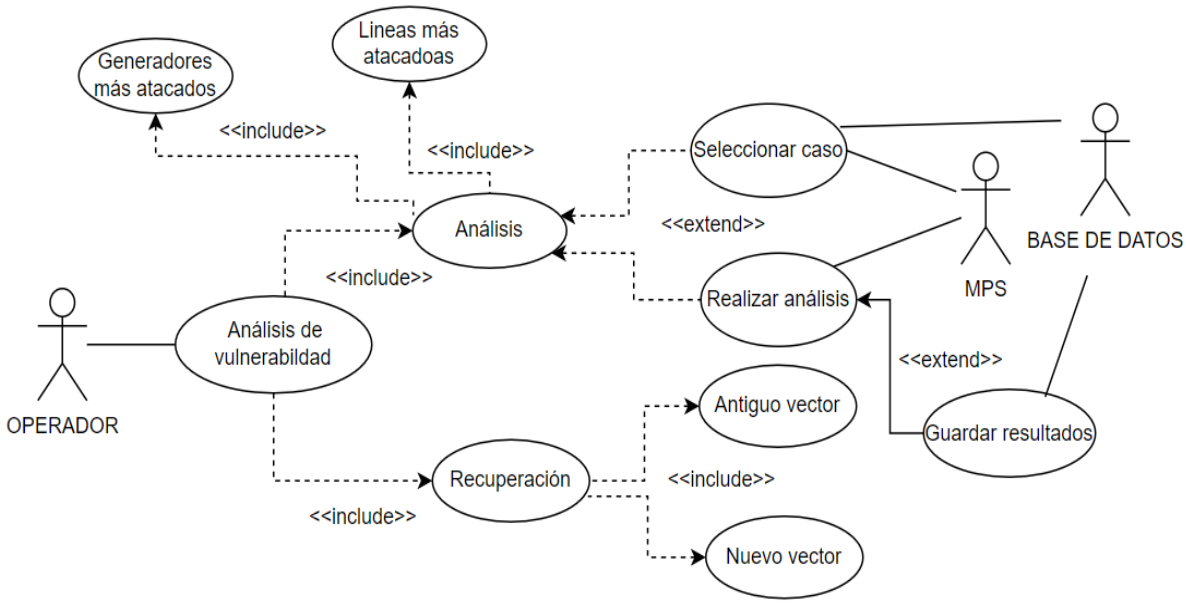


Ilustración 19. Casos de uso modulo Análisis de vulnerabilidad

Fuente: propia

2.4.3.1 DOCUMENTACIÓN CASOS DE USO ANÁLISIS DE VULNERABILIDAD

2.4.3.1.1 CASO DE USO: ANÁLISIS

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**13**
ID. Requerimiento(s)	RC01
Nombre Caso de Uso:	Análisis
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a una pestaña (Tab) del módulo análisis de vulnerabilidad donde se pueden realizar análisis y visualizar resultados.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar botón “análisis”	Muestra ventana principal del modulo	

2.4.3.1.2 CASO DE USO: SELECCIONAR CASO

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**5**
ID. Requerimiento(s)	RC01 Y RC04

Nombre Caso de Uso:	Seleccionar caso
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Cuando se selecciona el caso básicamente se traen datos tanto de MPS como de la base de datos para poder realizar un análisis de vulnerabilidad.

Actores:	
1	Operador
2	MPS
3	Base de datos

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador presiona el botón desplegable “seleccionar caso”	El sistema despliega todos los casos disponibles (topologías de red)	La lista de casos corresponde a un vector de enteros correspondientes al número de nodos de la topología de red.
2	El operador selecciona la topología de red de n nodos.	El sistema realiza una petición POST a MPS	El dato de entrada es la topología de red de n nodos.
3	MPS ejecuta función características en una instancia del servidor y retorna el objeto	El sistema recibe el objeto y lo que contiene se guarda en variables.	El objeto de salida es un “lhs”
3		El sistema realiza petición al servidor de la base de datos con el dato	

		de entrada de la acción 2.	
4	La base de datos consulta por los datos de vector de interdicción para el caso seleccionado.	El sistema recibe los datos y los guarda en variables	El dato de salida es el objeto que retorno el servidor de base de datos.

Flujo alternativo			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador pulsa el botón de generar vector de interdicción.	Consulta a la base de datos si ya se ha generado vector de interdicción.	Dato de entrada caso n
2		Si no se generó un vector de interdicción, sale una ventaja emergente alertando que no existe vector de interdicción para el caso seleccionado.	
3		El sistema pregunta si desea generar vector de interdicción.	
4	El operador desea generar vector de interdicción	El sistema redirecciona al módulo “Algoritmo genético”	

2.4.3.1.3 CASO DE USO: REALIZAR ANÁLISIS DE VULNERABILIDAD

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**14**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Realizar análisis de vulnerabilidad

Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Es un botón que ejecuta la función principal de Matlab donde realiza el análisis de vulnerabilidad de acuerdo con el vector de interdicción seleccionado.

Actores:	
1	Operador
2	MPS

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Seleccionar algún caso
4	Haber ejecutado algún algoritmo genético para la red seleccionada.

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El actor pulsa el botón realizar análisis de vulnerabilidad.	Espera la confirmación para realizar análisis. En caso de ser afirmativa el sistema realiza una petición POST al servidor MPS	Los datos de entrada corresponden al vector de interdicción y población de interdicción
2	MPS ejecuta la función y retorna datos	El sistema recibe los datos y pregunta al usuario si desea guardar resultados del análisis generado.	Los datos de salida es un objeto que contiene todos los resultados del análisis y métricas de recuperación de la red.

2.4.3.1.4 CASO DE USO: GENERADORES MÁS ATACADOS

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**15**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Generadores más atacados
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un panel desplegable que muestra el resultado de los generadores susceptibles de ser más atacados con su porcentaje.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Haber realizado un análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar panel desplegable de generadores más atacados	Despliega el panel y muestra los resultados	El dato corresponde a un string con la concatenación de los generadores más atacados.

2.4.3.1.5 CASO DE USO: LINEAS MÁS ATACADAS

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**16**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Líneas más atacadas
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un panel desplegable que muestra el resultado de las líneas susceptibles de ser más atacados con su porcentaje.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Haber realizado un análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar panel desplegable de líneas más atacados	Despliega el panel y muestra los resultados	El dato corresponde a un string con la concatenación de las líneas más atacados.

2.4.3.1.6 CASO DE USO: RECUPERACIÓN

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**17**

ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Recuperación
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a una pestaña del módulo de análisis de vulnerabilidad. En ella se pueden ver los resultados de recuperación de la red afectada.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Haber realizado un análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar pestaña Recuperación	Muestra los resultados de una red cuando se redistribuye la energía en el sistema y se logra recuperar su capacidad.	Los datos salientes corresponden a la métrica de recuperación.

2.4.3.1.7 CASO DE USO: ANTIGUO VECTOR

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**18**
ID. Requerimiento(s)	RC01 y RC04

Nombre Caso de Uso:	Líneas más atacadas
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un panel desplegable que muestra el antiguo vector (vector de interdicción) que fue atacado por un agente disruptor.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Haber realizado un análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar panel desplegable de antiguo vector	Despliega el panel y muestra los resultados	El dato de salida corresponde a un vector con las líneas y generadores más atacados.

2.4.3.1.8 CASO DE USO: NUEVO VECTOR

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**19**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Líneas más atacadas

Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde a un panel desplegable que muestra el resultado del vector de interdicción recuperado (nuevo vector) de la red del sistema.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Haber realizado un análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar panel desplegable nuevo vector	Despliega el panel y muestra los resultados	El dato de salida corresponde a un vector nuevo

2.4.3.1.9 CASO DE USO: GUARDAR RESULTADOS

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**20**
ID. Requerimiento(s)	RC04
Nombre Caso de Uso:	Guardar resultados
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	En este caso de uso se pueden guardar los resultados generados por el análisis de vulnerabilidad
Actores:	

1	Operador
2	Base de datos

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad
3	Haber realizado un análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador realizó un análisis de vulnerabilidad.	Despliega una ventana modal y pregunta si se desea guardar los resultados	
2	El operador confirma pulsando el botón si	El sistema realiza una petición POST al servidor de la base de datos	El dato de entrada son resultados con
3	La base de datos guarda los resultados		

Flujo alternativo			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador realizó un análisis de vulnerabilidad	Consulta a la base de datos si ya se hay resultados en la base de datos	Dato de entrada caso n
2		Si ya se guardaron resultados el sistema no pregunta si quiere guardar resultados solamente muestra resultados.	

2.4.4 MODULO FLUJO ÓPTIMO DE POTENCIA

En el módulo flujo óptimo de potencia se encuentra los casos de uso asociados como se observa en la ilustración 20:

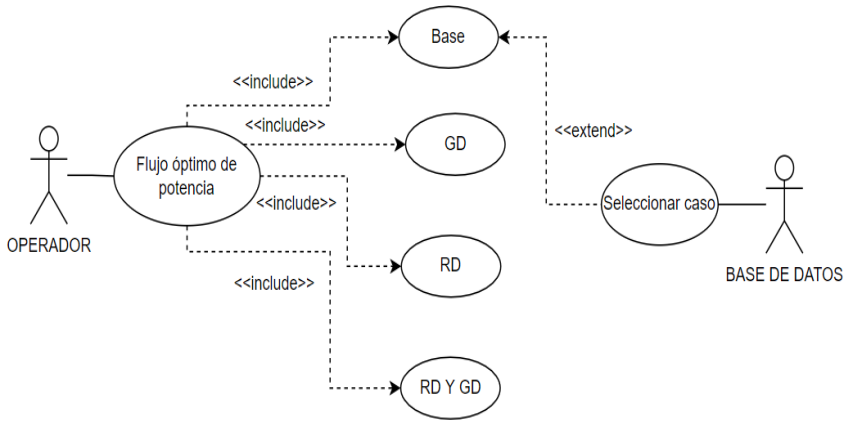


Ilustración 20. Casos de uso modulo flujo óptimo de potencia

Fuente: propia

2.4.4.1 DOCUMENTACIÓN CASOS DE USO MODULO FLUJO OPTIMO DE POTENCIA

2.4.4.1.1 CASO DE USO: BASE

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**22**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Base
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021

Descripción:	Corresponde a una pestaña (Tab) donde se puede seleccionar el caso y visualizar datos para el caso base. Corresponde a una ventana meramente informativa.
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de flujo óptimo de potencia

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	Pulsar la pestaña Base	Se puede visualizar una página que contiene caso de uso seleccionar caso y una tabla para ver resultados	

2.4.4.1.2 CASO DE USO: SELECCIONAR CASO

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**5**
ID. Requerimiento(s)	RC01 y RC04
Nombre Caso de Uso:	Seleccionar caso
Autor:	Desarrollador
Fecha:	Noviembre 2021 / actualizado diciembre 2022
Descripción:	En el módulo flujo óptimo de potencia se tiene un botón desplegable llamado “Seleccionar caso” en la pestaña de base. Sirve para consultar los resultados de los flujos óptimos para todos los casos.
Actores:	
1	Operador

2	Base de datos

Precondiciones:	
1	Estar autenticado en el sistema
2	Estar en el módulo de algoritmo genético

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador presiona el botón desplegable.	El sistema despliega todos los casos disponibles (topologías de red)	La lista de casos corresponde a un vector de enteros correspondientes al número de nodos de la topología de red.
2		El sistema realiza una petición POST al servidor de la base de datos.	El dato de entrada es la topología de red de n nodos.
3	La base de datos consulta si existen resultados para flujo óptimo de potencia	El sistema retorna los datos y los guarda en variables.	El dato de salida es un objecto que tiene todos los resultados que generó un análisis de vulnerabilidad.

Flujo alternativo			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador selecciona caso	Consulta a la base de datos si ya se hay resultados en la base de datos	Dato de entrada caso n
2	La base de datos consulta si existen resultados para flujo óptimo de potencia	Si no hay datos el sistema manda una alerta diciendo que no hay datos de flujo óptimo de potencia. El sistema pregunta	

		si desea generar análisis de vulnerabilidad.	
3	El operador confirma si desea generar un análisis de vulnerabilidad	El sistema redirecciona al módulo de análisis de vulnerabilidad	

2.4.4.1.3 CASO DE USO: RD

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**22**
ID. Requerimiento(s)	RC01 Y RC04
Nombre Caso de Uso:	RD
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde una pestaña del módulo donde se consultan resultados de la topología de red aplicando respuesta a la demanda (generados a partir del análisis de vulnerabilidad)
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo flujo óptimo de potencia

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1		Visualiza los datos de respuesta a la demanda en la	

	Entrar en la pestaña RD	tabla correspondiente se visualiza la gráfica correspondiente.	

2.4.4.1.4 CASO DE USO: GD

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**22**
ID. Requerimiento(s)	RC01 Y RC04
Nombre Caso de Uso:	GD
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde una pestaña del módulo donde se consultan resultados de la topología de red aplicando generación distribuida (generados a partir del análisis de vulnerabilidad)
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo flujo óptimo de potencia

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1		Visualiza los datos de generación distribuida en la	

	Entrar en la pestaña GD	tabla correspondiente se visualiza la gráfica correspondiente.	

2.4.4.1.5 CASO DE USO: RD Y GD

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**23**
ID. Requerimiento(s)	RC01 Y RC04
Nombre Caso de Uso:	RD y GD
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Corresponde una pestaña del módulo donde se consultan resultados de la topología de red aplicando respuesta a la demanda y generación distribuida (generados a partir del análisis de vulnerabilidad)
Actores:	
1	Operador

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo flujo óptimo de potencia

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1		Visualiza los datos de respuesta a la demanda y	

	Entrar en la pestaña RD y GD	generación distribuida en la tabla correspondiente se visualiza la gráfica correspondiente.	

2.4.5 MODULO REPORTES

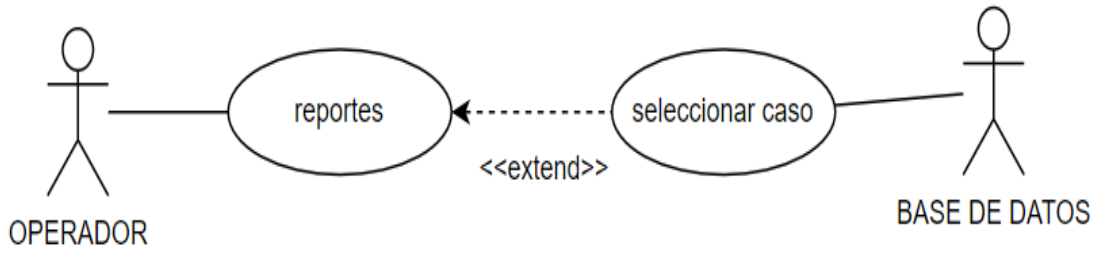


Ilustración 21. Casos de uso modulo reportes

Fuente: propia

2.4.5.1 DOCUMENTACIÓN CASOS DE USO

2.4.5.1.1 CASO DE USO: SELECCIONAR CASO

CASO DE USO DEL SISTEMA	
ID, Caso de Uso:	**5**
ID. Requerimiento(s)	RC01 Y RC04
Nombre Caso de Uso:	Seleccionar caso
Autor:	Desarrollador
Fecha:	Noviembre 2021 / última actualización diciembre 2021
Descripción:	Cuando se selecciona el caso nos consulta la base de datos por datos generados de resiliencia y resúmenes de carga por nodos.
Actores:	
1	Operador
2	Base de datos

Precondiciones:	
1	Estar autenticado en el sistema
2	Entrar al módulo análisis de vulnerabilidad

Flujo Básico o Normal			
Acción del actor		**Acción del sistema**	**Datos de entrada/ Datos de salida**
1	El operador presiona el botón desplegable “seleccionar caso”	El sistema despliega todos los casos disponibles (topologías de red)	La lista de casos corresponde a un vector de enteros correspondientes al número de nodos de la topología de red.
2	El operador selecciona la topología de red de n nodos.	El sistema realiza una petición POST al servidor de la base de datos	El dato de entrada es la topología de red de n nodos.
3	La base de datos consulta por datos de resiliencia y resumen de carga por nodos	El sistema recibe el objeto y lo que contiene se guarda en variables.	El dato de salida es un objeto.
3		El sistema genera dos gráficas que se visualizan en la ventana.	

2.5 MODELADO DEL NEGOCIO

Con modelos UML se generan modelos visuales[28] del modelo de negocio de la aplicación.

Se utilizan diagramas de actividades para describir visualmente los resultados que se obtienen en cada módulo de la aplicación.

ara obtener los resultados.

2.5.1 MODULO ALGORITMO GENETICO

2.5.1.1 DIAGRAMAS DE ACTIVIDAD

[28] Las acciones en la vista o interfaz

El módulo algoritmo genético define dos actividades principales; siendo la primera actividad principalmente seleccionar el caso donde directamente ejecutamos en MPS una función que nos devuelve las características o parámetros para generar un algoritmo genético.

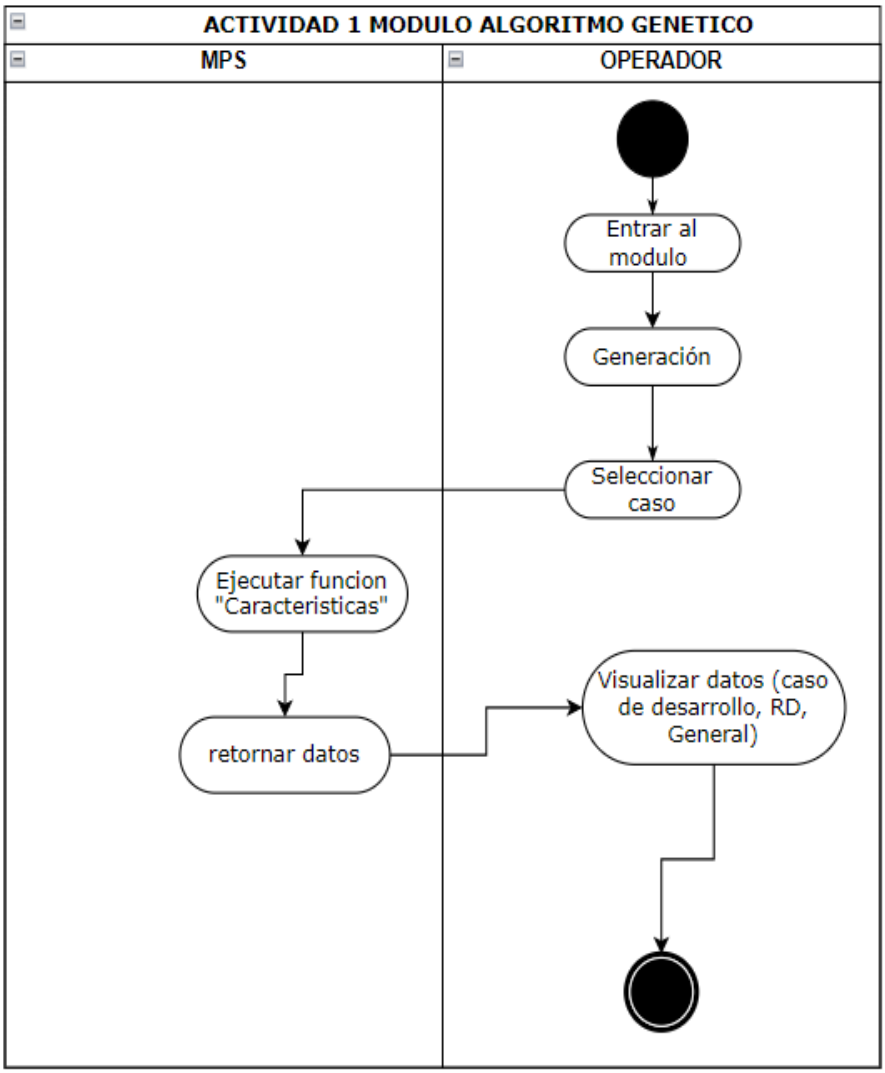


Ilustración 22. Actividad 1 Modulo algoritmo genético

Fuente: propia

La segunda actividad que se realiza en el módulo algoritmo genético es generar el vector de interdicción, donde básicamente MPS ejecuta el algoritmo genético retornando los datos y el operador decide si se guardan los resultados en la base de datos.

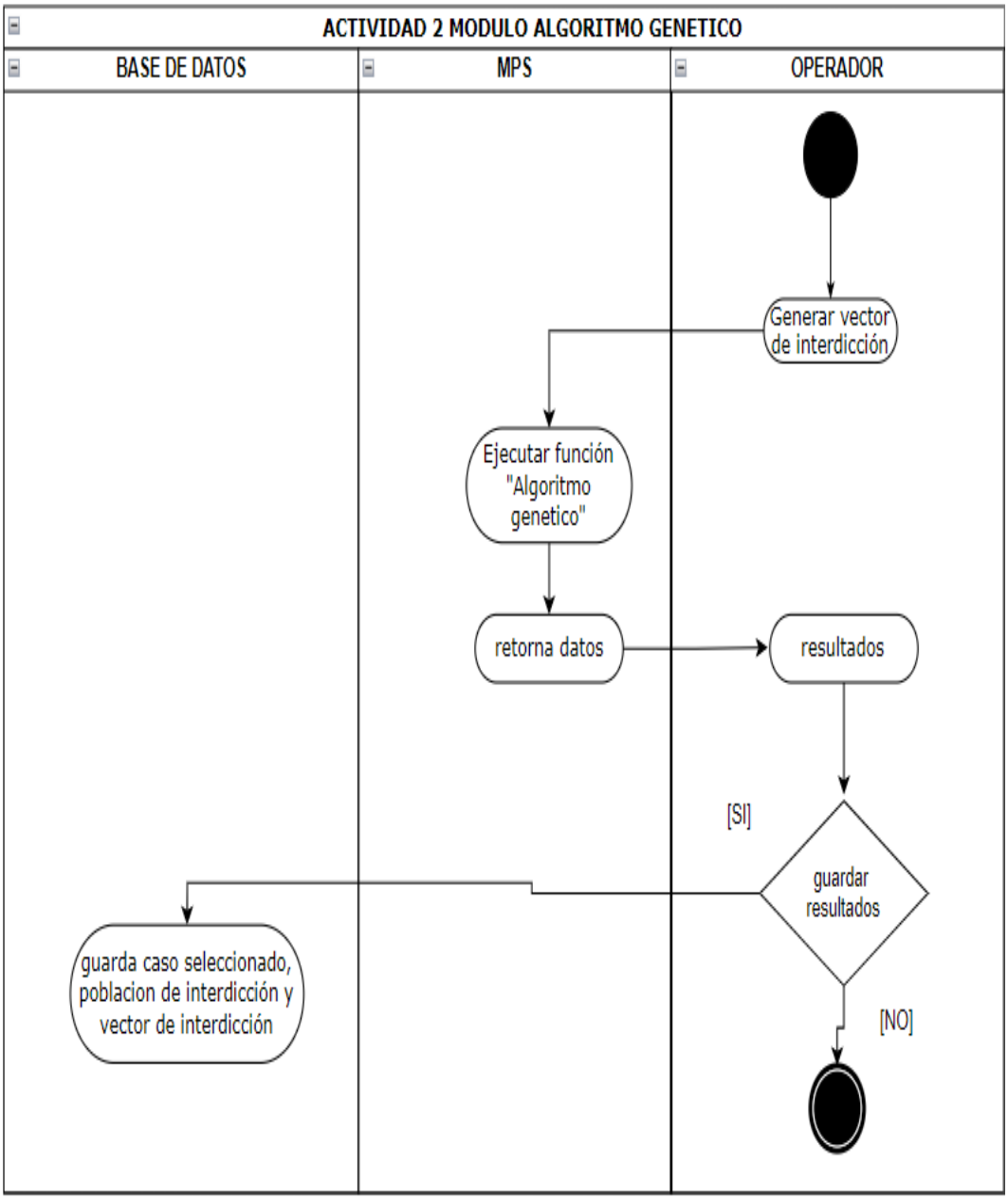


Ilustración 23. Actividad 2 modulo Análisis de vulnerabilidad

Fuente: propia

2.5.1.2 DIAGRAMA DE SECUENCIA

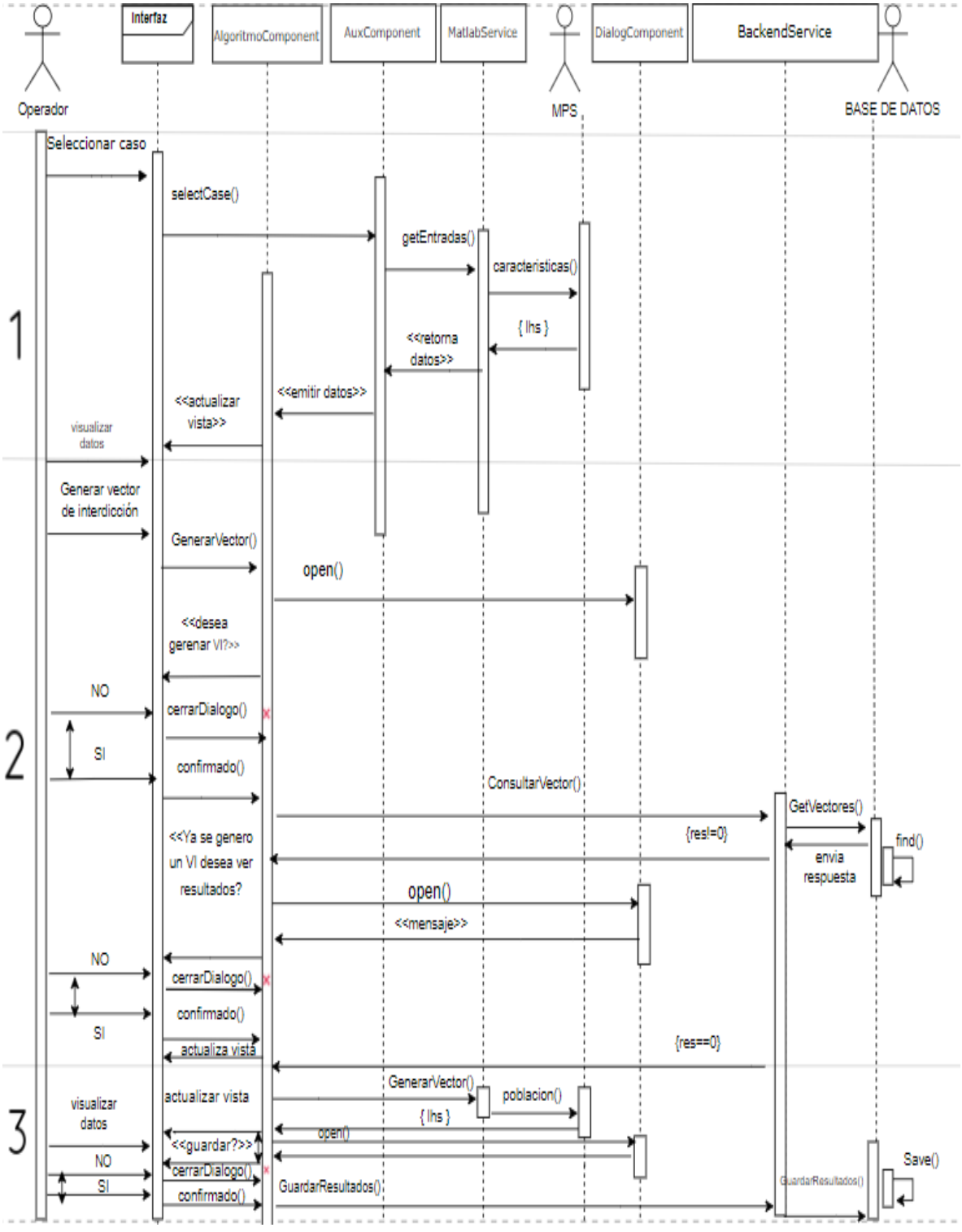


Ilustración 24. Diagrama de secuencia Modulo algoritmo genético

Fuente: propia

2.5.1.2.1 DOCUMENTACIÓN DIAGRAMA DE SECUENCIA

En la ilustración 24 se muestra las secuencias completas del módulo algoritmo genético. La secuencia permite ver la interacción interna de las clases con sus funciones y una vista más amplia de los diagramas de actividad. En la siguiente tabla se hace la descripción detallada donde se asocian las interacciones secuenciales con las correspondientes funciones de las clases lo que permite un mayor entendimiento del módulo:

1	**Seleccionar caso:** Es la acción que realiza el usuario en la interfaz y que se define en 2.4.2.1.2
	selectCase(): Es la función que se llama cuando se selecciona el caso (recibe el parámetro caso de tipo string que es el numero de la topología de red) . Esta función se encuentra definida en AuxComponent[29];esta clase cuando se inicializa reconoce la ruta del módulo en el que actual se encuentra, en este caso encuentra que se llama desde el módulo algoritmo genético y por lo tanto ejecutará la función getEntradas() definida en la misma clase y le pasará el parámetro de el numero de la topología de red.
	getEntradas(): Es una función definida en MatlabService se accede a ella cuando la función con el mismo nombre, pero definida en la clase auxiliar la llama[30]. Su función es hacer una petición POST al servidor MPS con el dato de seleccionar el caso, para que MPS internamente nos devuelva una serie de datos y caracteristicas de la topología de red.
	caracteristicas(): Es la función que se ejecuta en la instancia del servidor de Matlab llamada “Entradas”.
	{lhs}: Cuando se ejecuta la anterior función MPS devuelve como respuesta un objeto llamado lhs
	<<emitir datos>>: Dentro del scope o alcance de la función getEntradas() definida en AuxComponent ocurrieron las anteriores acciones. Cuando finalmente se tiene el objeto lhs

[29] Esta clase auxiliar se reutiliza en todos los módulos porque su diseño comprende el seleccionar siempre el caso de la topología de red 2.4.2.1.2, 2.4.3.1.2 , 2.4.4.1.2 , 2.4.5.1.1

[30] Al seleccionar el caso

	este se emite[31] a la clase AlgoritmoComponent en una función llamada selectCase() y que recibe ese evento.
	<<actualizar vista>>: Finalmente la acción que realiza AlgoritmoComponent es actualizar su interfaz, con el diseño definido se pintan los datos emitidos.
	visualizar datos: Una vez ocurrido el anterior proceso el operador podrá visualizar los datos con los casos de uso definidos en 2.4.2.1.3, 2.4.2.1.4 , 2.4.2.1.5.
2	**Generar vector de interdicción:** Es la acción que el usuario debe realizar en 2.4.2.1.6
	generarVector(): Se define esta función en la clase AlgoritmoComponent. Cuando se pulsa el anterior botón realiza una serie de instrucciones dentro de su scope, una de ellas es llamar la función open()
	open(): ¿Es una función que se encuentra en definida DialogComponent se utiliza para abrir una ventana modal , cuando el usuario pulsa el botón de generar vector interdicción pregunta si <<desea generar un vector de interdicción?>> Al responder no internamente la ventana modal ejecuta la función cerrarDialogo() donde no ocurre ninguna acción. Al responder si se ejecuta la función confirmado()[32] que retorna un boolean true a la función open() y que permite realizar llamar a la siguiente función.
	consultarVector(): Consultar vector es una función que está definida en la clase BackenService y que se llama solo si existe la confirmación del usuario. Lo que va a hacer esta función es una petición POST al servidor de la base de datos, el parámetro es el caso o la topología de red.

[31] Cuando se quieren pasar datos de una clase a otra en angular se utiliza un evento llamado @Output el cual emite los datos necesarios.
[32] estas funciones están definidas en la clase DialogComponent, pero se llaman en el scope de la función open().

	GetVectores(): Es una función que esta definida en la ruta especifica para consultar vectores de interdicción, lo que hace es con el parámetro del caso es consultarlo en la base de datos, la función retorna la respuesta a la clase BackenService.
	{res!=0}: La anterior respuesta es retornada a la clase AlgoritmoComponent dentro del scope de la función generarVector() en donde si es res!=0 significa que ya hay un vector de interdicción. Por lo tanto, dentro la función generarVector() se abrirá nuevamente un dialogo y el mensaje será <<Ya se generó un vector de interdicción desea ver resultados?>> En donde si el usuario responde que no simplemente se cierra el módulo, en el caso que el usuario responda que si utiliza los mismos datos de las respuesta se alojan en variables y se actualiza la vista permitiendo saltar a 2.4.2.1.8.
3	**{res==0}:** Si la respuesta es igual a cero significa que la base de datos no tiene resultados generados para un vector de interdicción por lo tanto es posible generar uno.
	generarVector(): Es una función que se define en MatlabService y que se llama luego de no encontrar resultados generados. Está función permite hacer una petición POST a MPS[33].
	población(): Es una función que ejecuta en la instancia del servidor de MPS llamada “Algoritmo” de donde básicamente resultan el vector de interdicción y la población de interdicción.

[33] Los parámetros que recibe la función generarVector() son los resultantes del objeto lhs resultante de 2.4.2.1.2.

	{lhs}: Nuevamente MPS retorna un objeto lhs , el cual es básicamente la población de interdicción y el vector de interdicción. Después del retorno se actualiza la vista porque lhs llegó a AlgoritmoComponent. Cuando los datos cargan y se visualizan automáticamente en 2.4.2.1.8 ocurre un proceso paralelo donde primero se abre una ventana modal open(), y básicamente se pregunta si se quieren guardar los resultados.
	GuardarResultados(): Cuando el usuario confirma si desea guardar resultados se ejecuta la función guardar resultados definida en BackendService, es llamada dentro del scope de la función generarVector de el modulo algoritmo genético. Esta función realiza una petición POST al servidor de la base de datos[34] donde el parámetro que se le envía será un objeto donde se tiene el caso de la topología de red , la población y el vector de interdicción.
	GuardarResultados(): Es una función definida en el servidor de la base de datos contiene la lógica[35] que guarda en la base de datos el parámetro que se recibe, se accede a ella cuando se realiza la petición (en la anterior función) y se le entrega el parámetro el cual se guardará en la base de datos mediante su función estándar save(). De esa manera se cumple 2.4.2.1.7 y permite realizar todo el Flow de acciones que realiza el módulo algoritmo genético.

[34] Básicamente se le entregan como parámetros la URI y el dato que se quiere guardar para hacer la petición
[35] De la URI

2.5.2 MODULO ANÁLISIS DE VULNERABILIDAD

2.5.2.1 DIAGRAMAS DE ACTIVIDAD

En la primera actividad del módulo análisis de vulnerabilidad describe la acción de MPS y la base de datos al seleccionar el caso; para retornar datos necesarios para el análisis de vulnerabilidad.

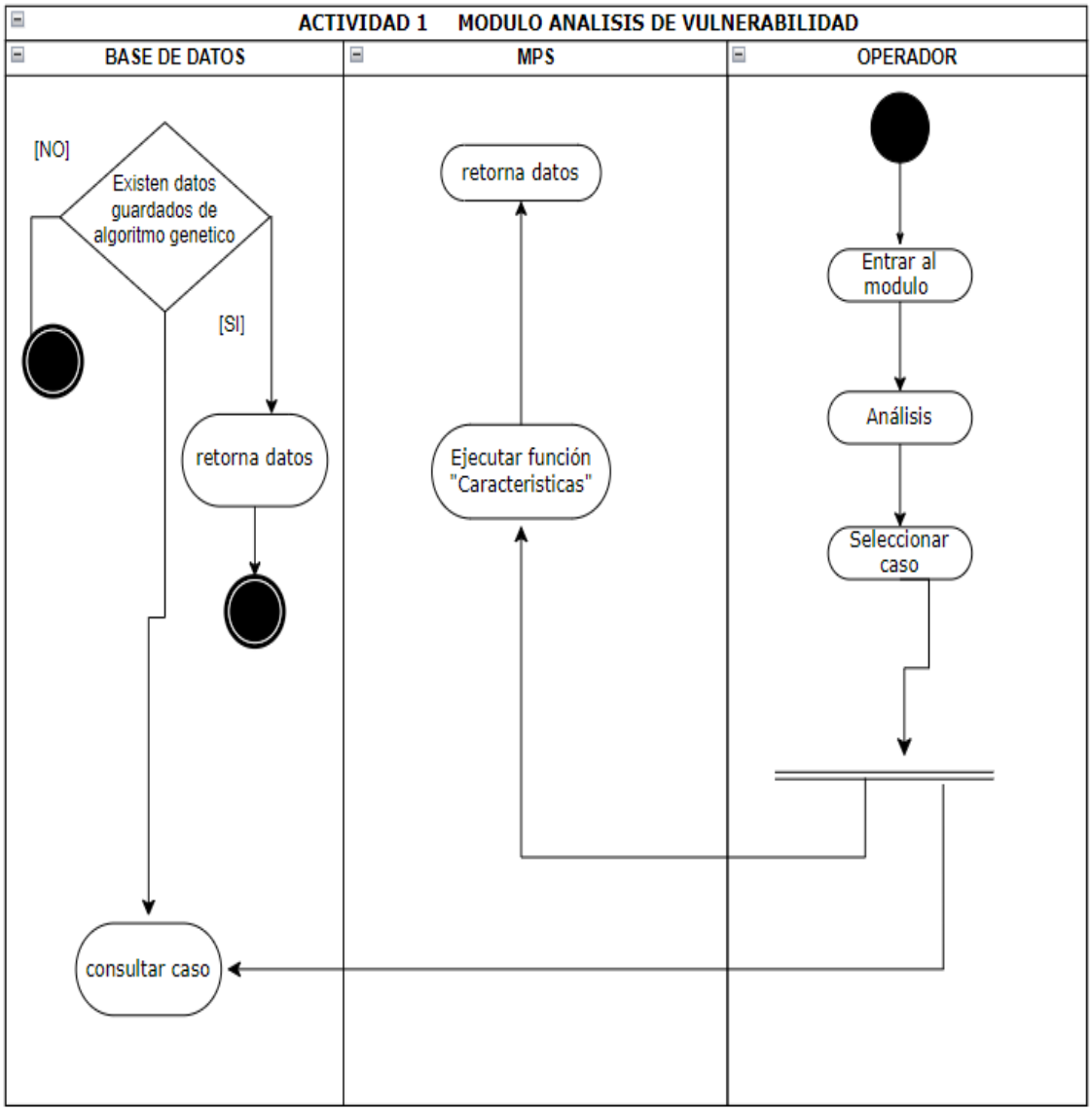


Ilustración 25. Actividad 1 Modulo análisis de vulnerabilidad

Fuente: propia

La segunda actividad que debe realizar el operador en el módulo análisis de vulnerabilidad es realizar el análisis; donde interactúan MPS y la base de datos para ejecutar la función principal donde está el algoritmo del análisis de vulnerabilidad. El cual retorna datos que se guardan en la base de datos.

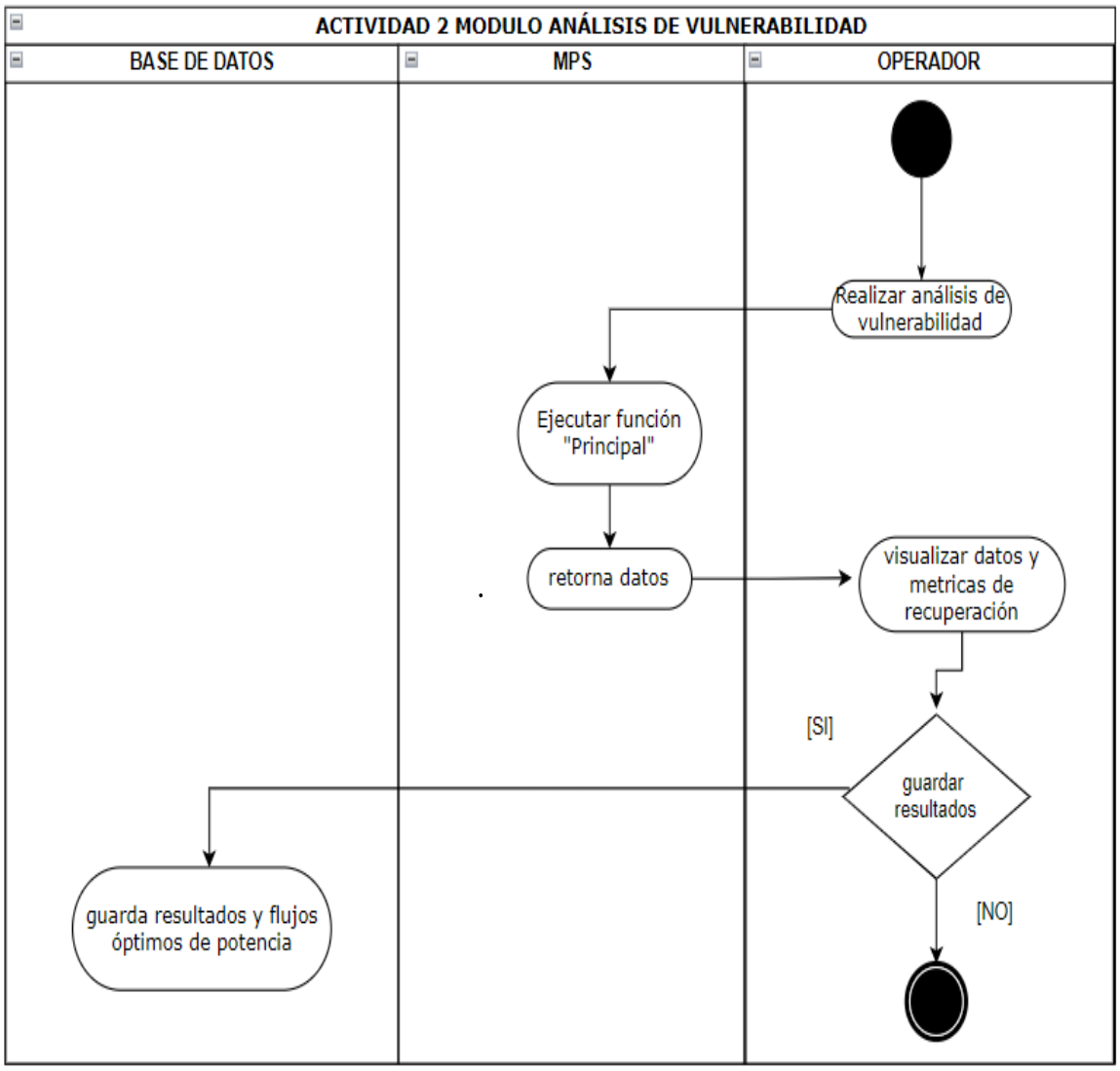


Ilustración 26. Actividad 2 modulo Análisis de vulnerabilidad

Fuente: propia

2.5.2.2 DIAGRAMA DE SECUENCIA

En la ilustración 27 se puede observar el diagrama de la secuencia completa del módulo Análisis de vulnerabilidad

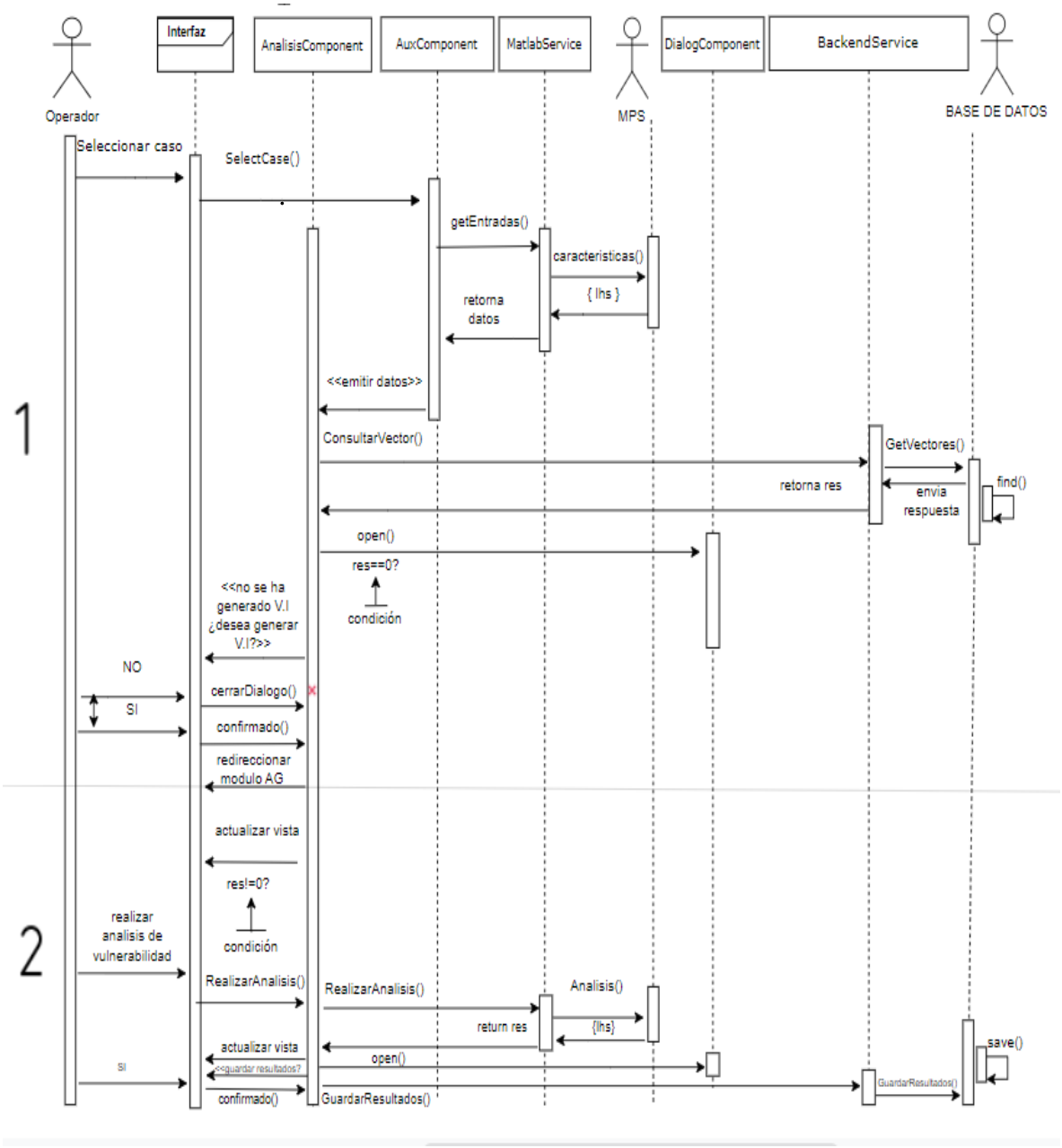


Ilustración 27. Diagrama de secuencia modulo Análisis de vulnerabilidad

Fuente: propia

2.5.2.2.1 DOCUMENTACIÓN DIAGRAMA DE SECUENCIA

En la ilustración 27 se muestra la secuencia que realiza el sistema y su interacción con las clases del sistema en el módulo análisis de vulnerabilidad. En el diagrama se expone a profundidad como la unión de las actividades que debe realizar el operador dan un resultado final que podrá ser guardado en la base de datos. En la siguiente tabla se hace la descripción detallada donde se asocian las interacciones secuenciales con las correspondientes funciones de las clases lo que permite un mayor entendimiento del módulo:

1	**Seleccionar caso:** Es la acción que realiza el usuario en la interfaz y que se define en 2.4.3.1.2 . Cuando se selecciona el caso en el módulo análisis de vulnerabilidad se realiza la misma la petición a la instancia del servidor llamada entradas , porque se necesitan algunos datos de esas caracteristicas de la red para realizar un análisis de vulnerabilidad. Por lo tanto de la clase AuxComponent se emiten tales datos a AnalisisComponent.
	ConsultarVector(): Es una función se llama justo antes de realizar el anterior evento. Esta función está definida en la clase BackendService , básicamente hace la petición con la URI y el parámetro de cuando se selecciona el caso.
	GetVectores(): Es la función que hace parte de la lógica de la URI especifica para consultar vectores de interdicción, usando el parámetro de seleccionar caso. Esta función del servidor de la base de datos consulta a la base de datos si existen datos según el caso seleccionado. Se retorna la respuesta a BackendService y AnalisisComponent es notificado con un subscribe[36].
	res==0[37]: Si la respuesta anterior no contiene resultados , se abrirá una ventana modal en la interfaz diciendo <<No se ha generado vector de interdicción, desea generar un vector de interdicción>>? Si se responde que no se cierra la acción. Si se responde que si se redirecciona al módulo Análisis de vulnerabilidad.

[36] Con un observable se retorna la respuesta a AnalisisComponent
[37] Se deja explicita como una condición en el diagrama de secuencia

	Res!=0: Cuando la respuesta es diferente de cero se puede pulsar el botón de realizar análisis de vulnerabilidad[38].
	Realizar análisis de vulnerabilidad: Es la acción que se realiza en 2.4.3.1.3 .
	RealizarAnalisis(): Es la función definida en el AnalisisComponent, es llamada después de la anterior acción en la interfaz. Dentro de su scope contiene los datos de las siguientes funciones.
	RealizarAnalisis(): Es una función que está definida en MatlabService , realiza una petición POST a MPS a la instancia de servidor llamada “Análisis”, usando la URI definida con los datos del caso y el vector de interdicción.
	Analisis(): Es la función que se ejecuta en la instancia del servidor de Matlab para hacer el análisis de vulnerabilidad.
	{lhs}: MPS retorna el objeto lhs de la anterior función a MatlabService y se notifica a AnalisisComponent de los datos donde se actualiza su vista. En ese momento se abre una ventana modal preguntando si se desean guardar los resultados[39]. Si el usuario confirma guardar resultados se cumple 2.4.3.1.9

[38] En el diagrama se define como <<Actualizar vista>>

[39] **Nota:** Aunque en el diagrama de secuencia no se dejo explicito, el programa busca si ya hay análisis de vulnerabilidad generados, en caso de que ya haya uno no se pregunta si se desea guardar resultados caso contrario abre una ventana modal preguntando si se desea generar uno.

2.5.3 MODULO FLUJO OPTIMO DE POTENCIA

2.5.3.1 DIAGRAMA DE ACTIVIDAD

En el módulo flujo óptimo de potencia se consultan los resultados del anterior modulo. En el simplemente seleccionamos el caso; donde se interactúa con la base de datos para retornar esos resultados. Al final podremos visualizarlos mediante TABS.

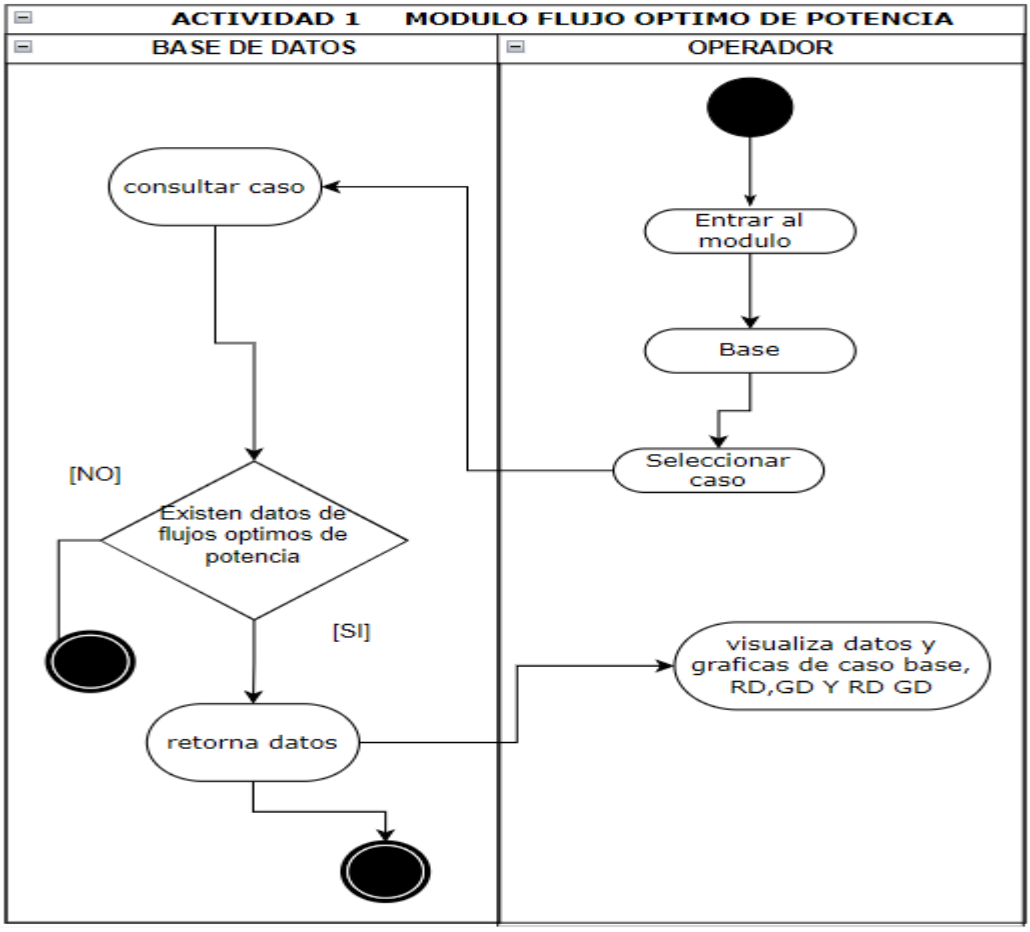


Ilustración 28. Diagrama de actividad modulo flujo óptimo de potencia

Fuente: propia

2.5.3.2 DIAGRAMA DE SECUENCIA

En el diagrama de secuencia del módulo flujo óptimo de potencia se ilustra principalmente la secuencia que debe hacer el sistema internamente para consultar los datos en la base de datos y retornarnos el resultado para visualizarlos.

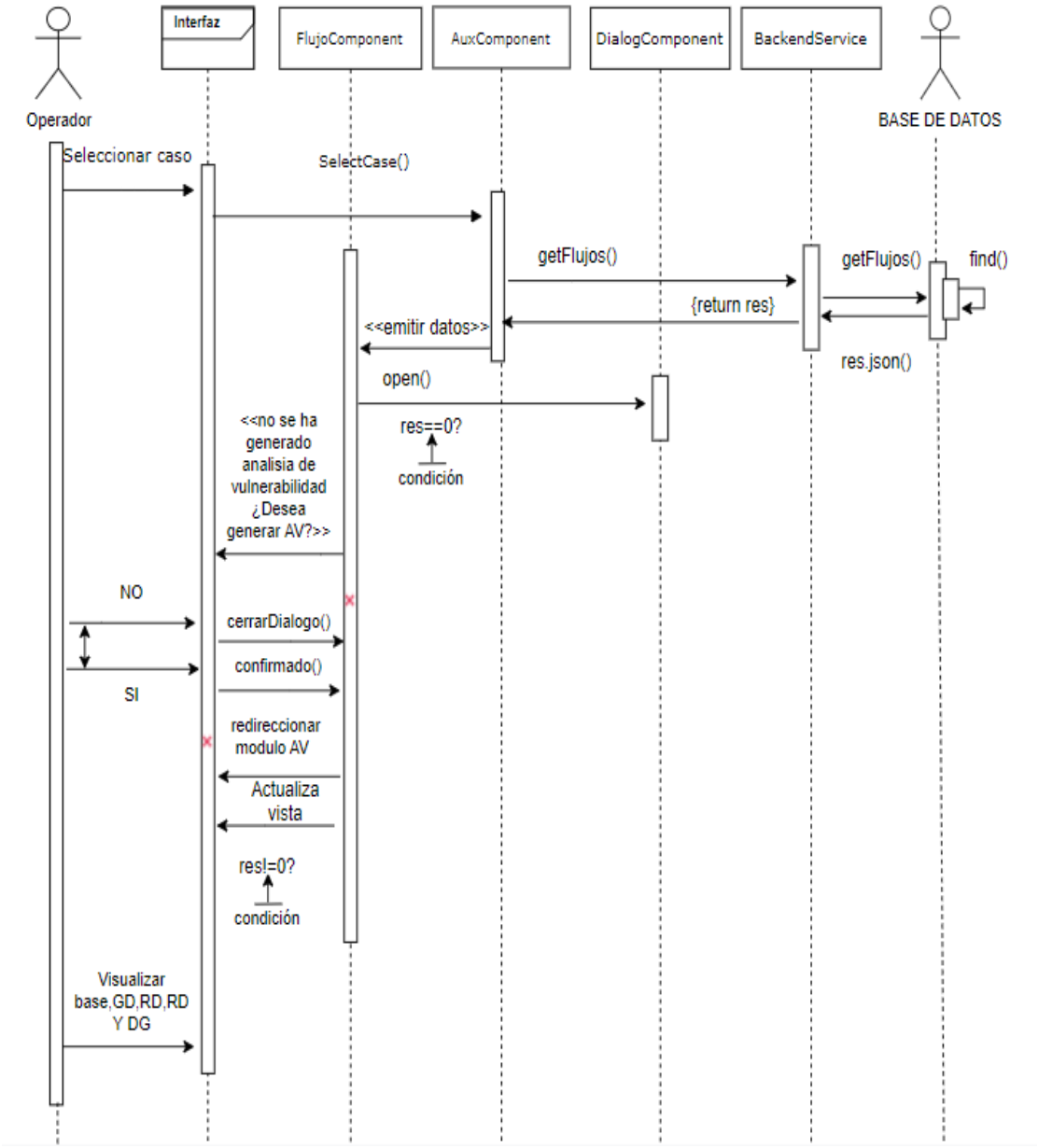


Ilustración 29. Diagrama de secuencia modulo Flujo óptimo de potencia

Fuente: propia

2.5.4 MODULO REPORTES

2.5.4.1 DIAGRAMA DE ACTIVIDAD

Al igual que el anterior modulo, en reportes consultamos los resultados sin embargo los datos se manejan de una manera diferente para ser adaptados al módulo mediante gráficos; la actividad que se realiza en este módulo es principalmente seleccionar el caso y consultar la base de datos.

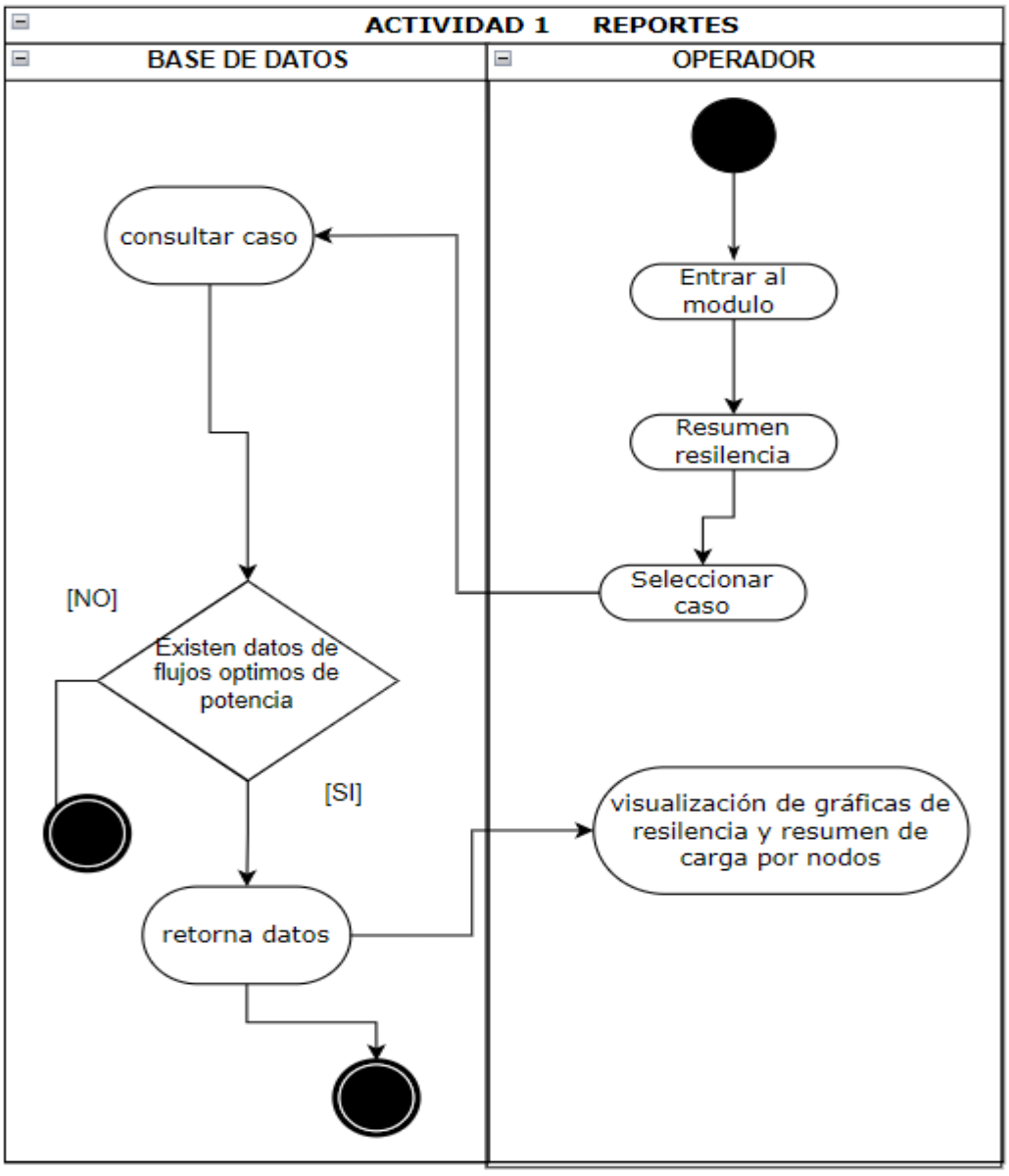


Ilustración 30. Diagrama de actividad modulo Reportes

Fuente: propia

2.5.4.2 DIAGRAMA DE SECUENCIA

En el diagrama de secuencia se ilustra la interacción de las clases que se usan para la secuencia del módulo de reportes donde al haber datos en la base de datos para los resultados se puede visualizar graficas: resumen de residencia y resumen de carga por nodos.

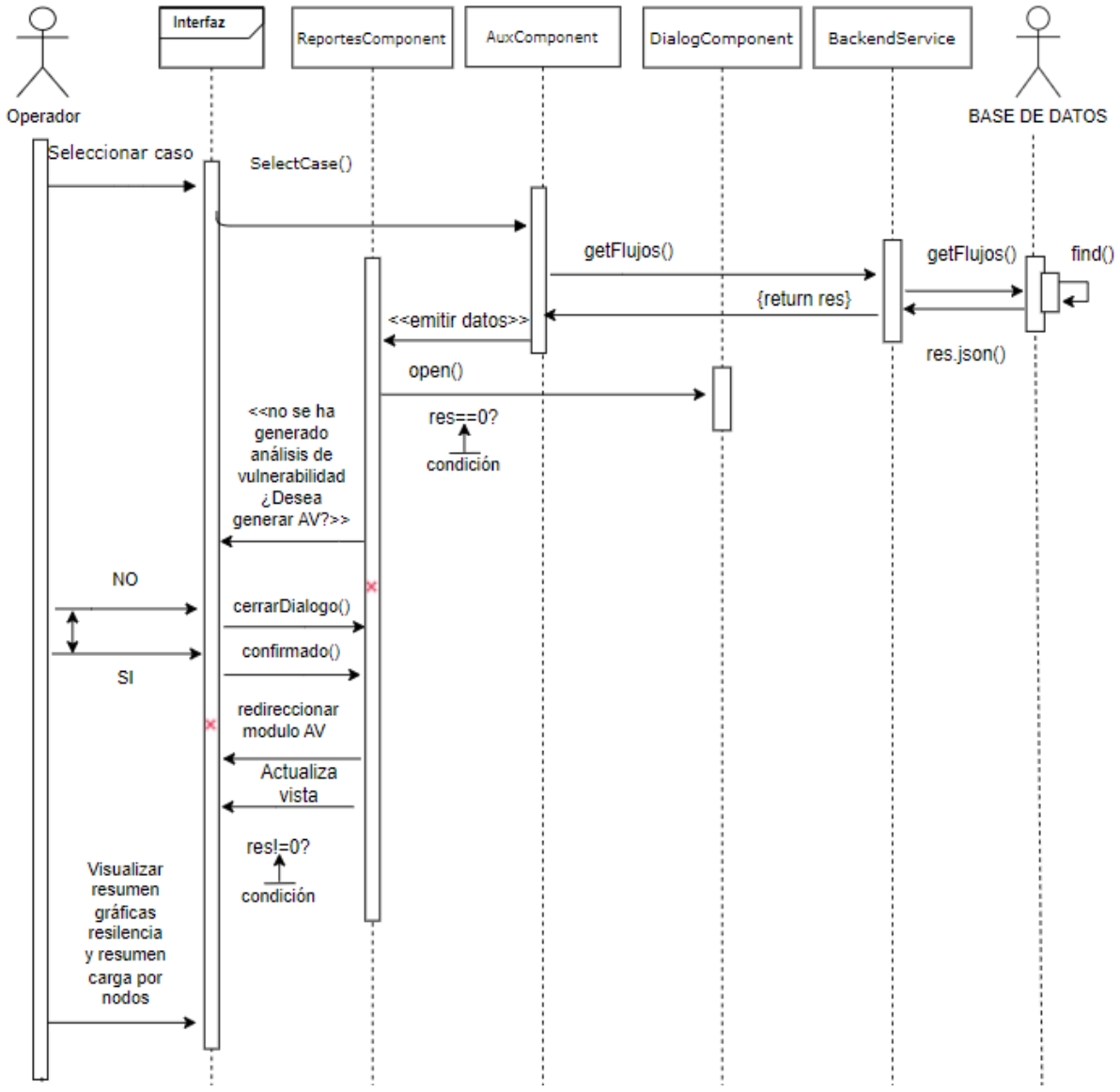


Ilustración 31. Diagrama de secuencia modulo Reportes

Fuente: propia

2.6 DIAGRAMA DE CLASES ANGULAR

En el diagrama de clases se exponen las clases utilizadas en los diagramas de secuencia y sus relaciones y dependencias. Las clases representadas con azul corresponden a los modelos y las clases en verde representan los controladores como se muestra en la ilustración

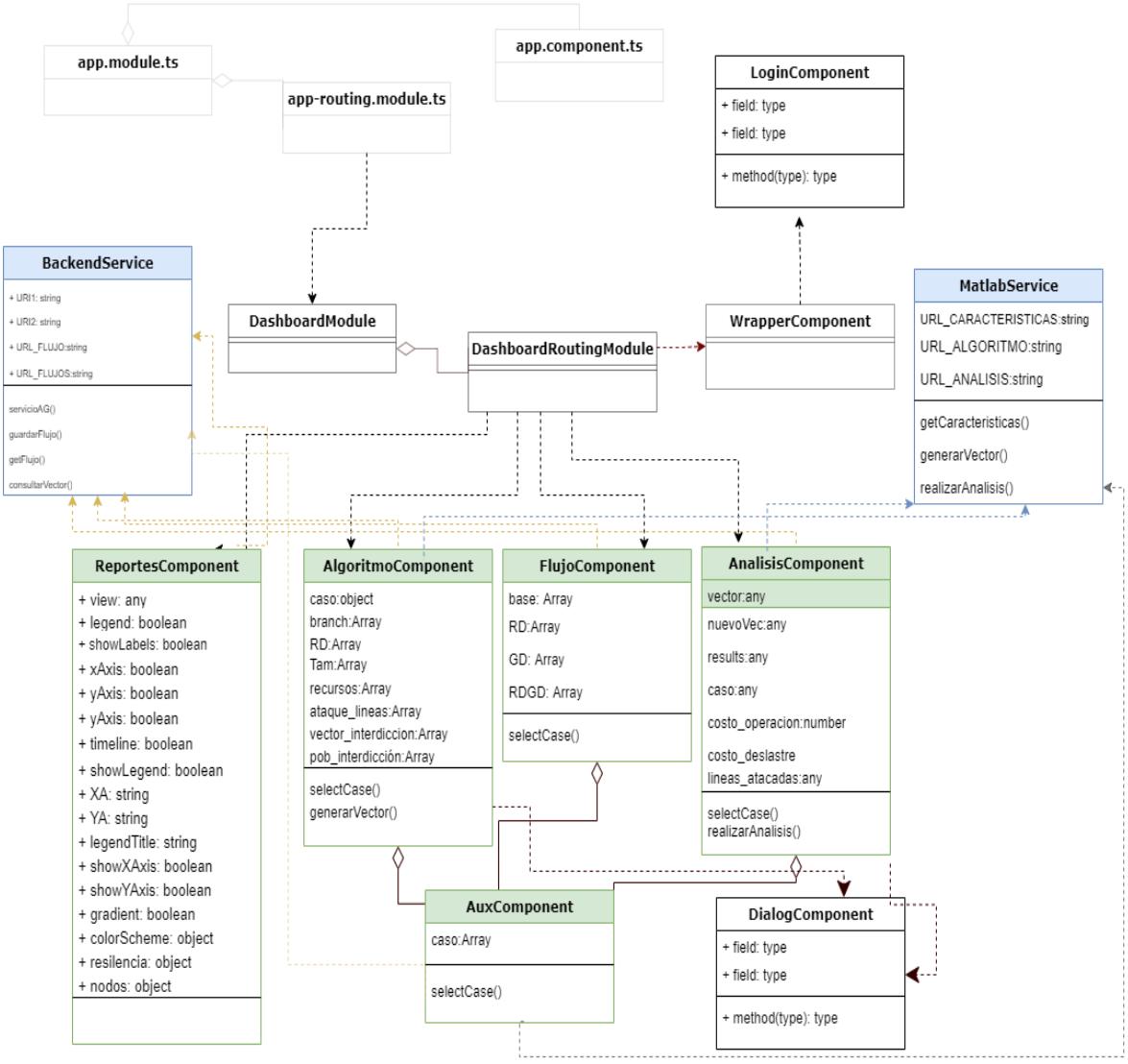


Ilustración 32. Diagrama de clases Angular

Fuente: propia

2.7 DIAGRAMA DE PAQUETES

2.7.1 CLIENTE

Dentro de la app de angular existen 5 archivos que tienen como funcionalidad la implementación del SPA[40]. Los archivos con extensión ts son las clases principales del programa; en el caso de app.module.ts[41] importa todos los recursos más generales como Bootstrap, fuentes o iconos. La clase especial app-routing.module.ts importa la carpeta dashboard[42],que contiene todos los componentes, para hacer rutas especiales con los estos[43] marcados en negro (Los módulos de la plataforma). Mientras que dashboard-routing.module.ts define el path es decir las rutas internas de la aplicación.

Por último, el paquete services o servicios ayuda a separar las carpetas que contienen las clases para usar los servicios REST.

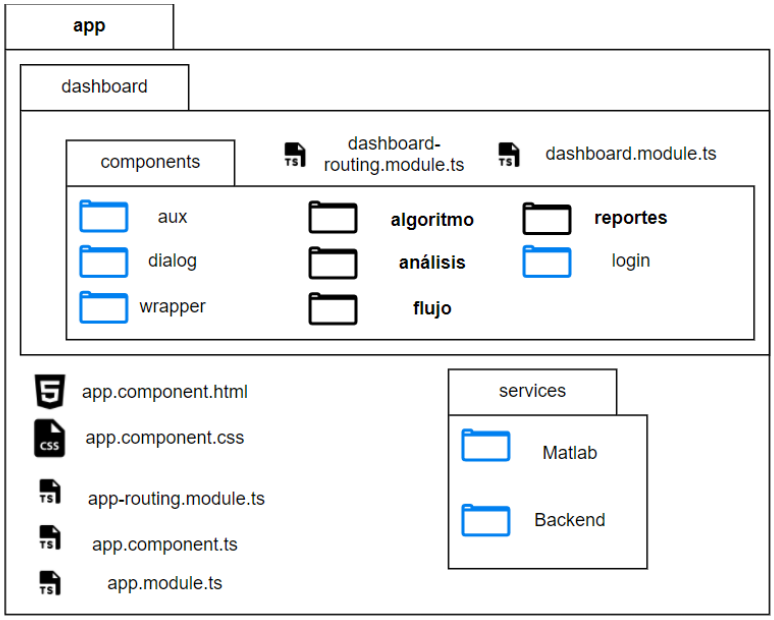


Ilustración 33. Diagrama de paquetes de Angular

Fuente: propia

[40] Single Page Aplication (Aplicación de página única) que permite ejecutar todo el contenido en una sola página, o cargar los contenidos HTML y CSS dependiendo la acción del usuario, pero se realiza en una página, en este caso app.component.html

[41] Los archivos terminados en component.ts importan la mayoría de los recursos visuales, por ejemplo, el archivo dashboard.module.ts importa menús, tablas, graficas.

[42] La carpeta dashboard (panel de control) es un contenedor que ayuda a estructurar mejor la aplicación.

[43] Los componentes son paquetes que constan básicamente de 4 archivos: un CSS, un HTML y una clase Typescript

2.7.2 MPS

Los algoritmos que se desarrollan en Matlab[44], utilizan el Matlab runtime para poder empaquetarlos 1.5.4.10.3. En la carpeta MATLAB_Runtine se alojan estos algoritmos en formato ctf que luego serán usados por el servidor MPS para crear las respectivas instancias del servidor con esas funciones.

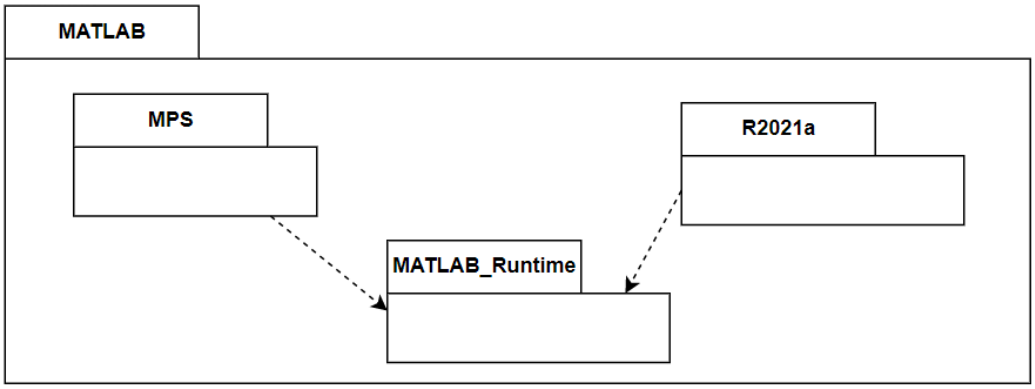


Ilustración 34. Diagrama de paquetes MPS

Fuente: propia

2.7.3 BACKEND

El servidor de base de datos tiene unas URI que son definidas en routes o rutas, estas rutas deben tener utilizan la lógica del controlador. y que hacen parte del paquete control. El control utiliza esquemas o modelos donde se define el prototipo para guardar datos en la base de datos; estos modelos se ubican en models.

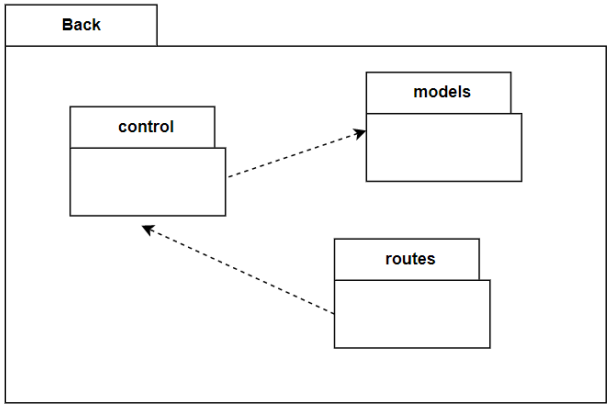


Ilustración 35. Diagrama de paquetes servidor base de datos

Fuente propia

[44] La carpeta R2021a describe el año de la versión y la licencia utilizada

2.8 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue describe los componentes internos del sistema a nivel de software y a nivel de hardware. Este diagrama ayuda a entender la arquitectura del sistema en su ejecución.

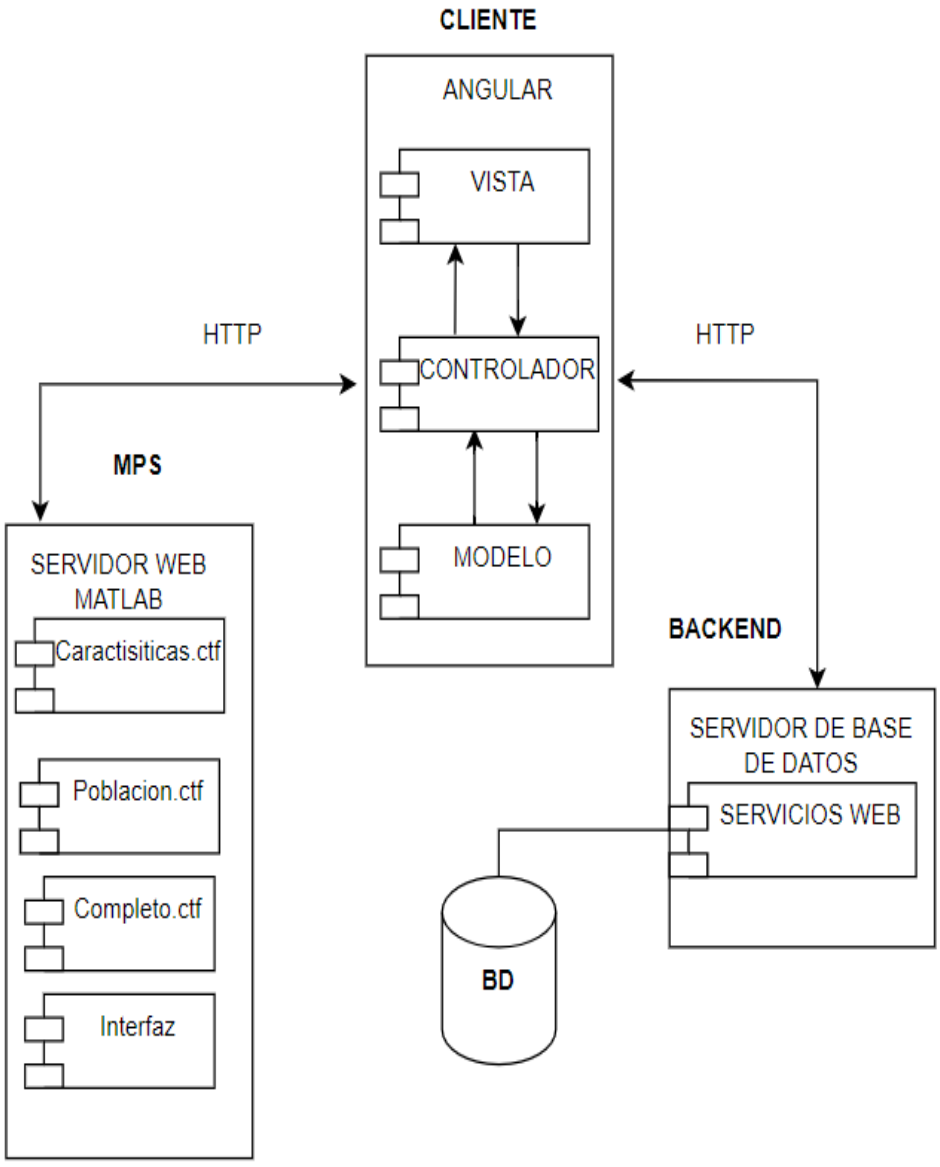


Ilustración 36. Diagrama de despliegue del sistema

Fuente propia

3 FASE DE CONSTRUCCIÓN

3.1 ARQUITECTURA DEL SISTEMA

El sistema general tiene la arquitectura cliente servidor[45]. Con esta arquitectura se pretende mediante el cliente usar el protocolo HTTP para para hacer peticiones a los servidores.

Los servidores reciben peticiones y procesan los datos para devolver una respuesta

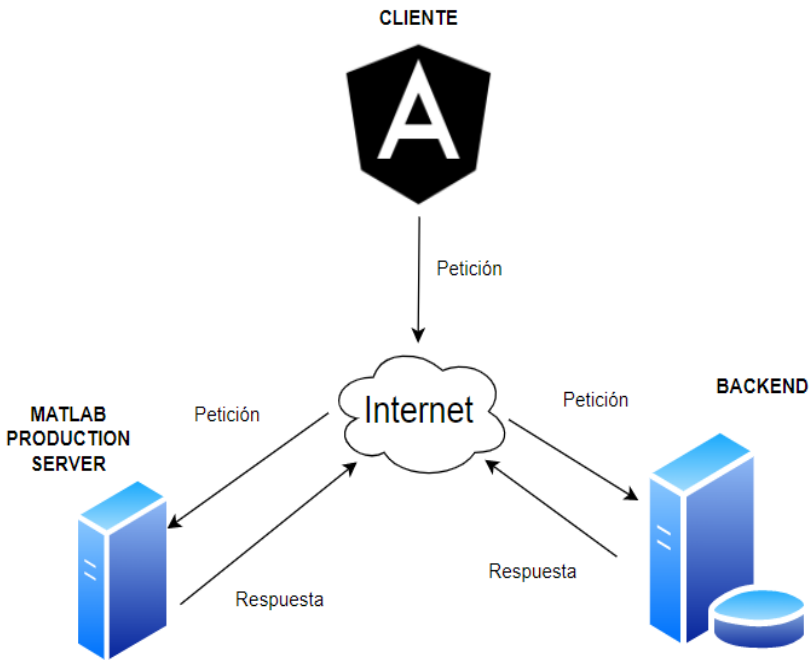


Ilustración 37. Diagrama arquitectura del sistema

Fuente propia

3.1.1 CLIENTE

Nuestro cliente angular utiliza el patrón modelo vista controlador[46]. Este patrón de arquitectura se basa en componentes y servicios.

[45] Define que los clientes solicitan servicios mediante protocolos a sistemas centralizados es decir servidores
[46] Estilo de arquitectura que separa los datos de la aplicación, su interfaz y lógica.

En general, la aplicación de angular utiliza dos capas, una que contiene los componentes y la otra los servicios, o components y services como se ilustra en 2.7.1.

Las clases que se definen en 2.6 que aparecen en verde se van a comunicar con los servicios que están en azul (modelos) .Las clases controlador también se comunican y con la vista. Con la ayuda de los controladores y los servicios angular realiza peticiones HTTP a los dos servidores.

3.1.2 SERVIDORES

La arquitectura del sistema utiliza dos servidores, MPS para ejecutar las funciones de Matlab y backend para guardar resultados en la base de datos. Los servidores procesan las peticiones y devuelven las respuestas al cliente.

3.1.2.1 MATLAB PRODUCTION SERVER

El servidor MPS funciona creando instancias de servidores. Se tienen tres instancias que son los algoritmos realizados en Matlab.

Para realizar las peticiones se deben hacer peticiones de tipo POST como se define en 1.5.4.10.4.1. Estas peticiones sincrónicas definen solo una llamada al servidor. Se acceden mediante una URI[47].La siguiente ilustración muestra las instancias de MPS.

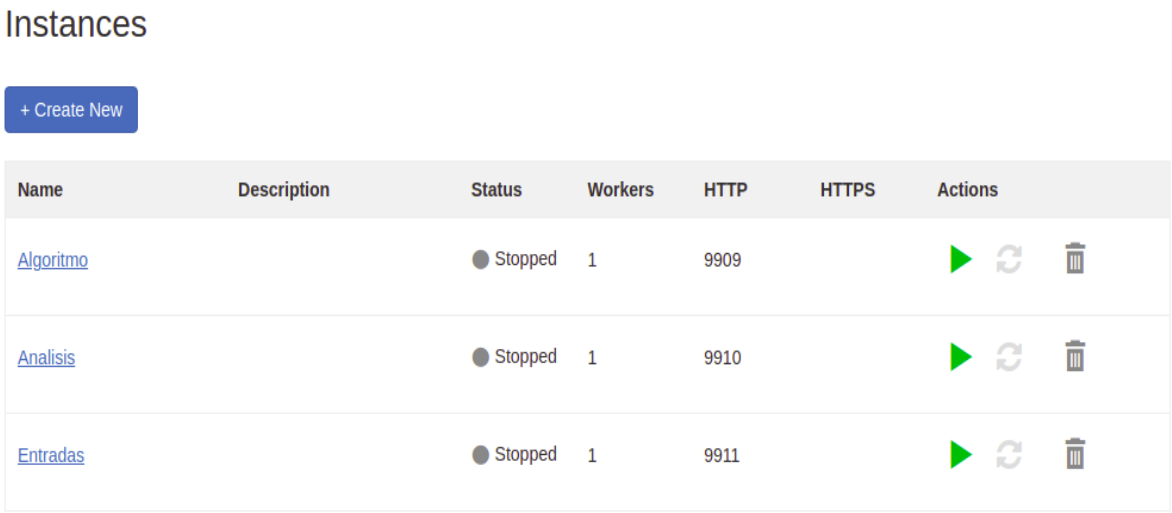


Ilustración 38. Instancias del servidor MPS

Fuente: propia

[47] Las URI definidas para las instancias del servidor se definen con el HTTP configurado en la instancia y el nombre es de la función original.

3.1.2.2 BACKEND

3.1.2.2.1 RUTAS

Las rutas del servidor son de tipo POST. Sirven básicamente para guardar y consultar los resultados generados por los módulos de la aplicación

Mediante la ilustración n se muestra un fragmento de código donde se describen las URI[48] necesarias para que el cliente realice peticiones y se consulten los datos en la base de datos.

```
const { Router } = require('express');
const router = Router();
const control = require('../control/logica.js');
router.post('/', control.GuardarVector);
router.post('/Vector', control.GetVectores);
router.post('/Results', control.GuardarFlujosOptimos);
router.post('/Flujo', control.GetFLujo);
module.exports = router;
```

Ilustración 39. Rutas servidor base de datos (Backend)

Fuente: propia

3.1.2.2.2 ESQUEMAS

Los esquemas moldean como se van a guardar y consultar los datos de MongoDB. Las consultas se harán con el caso.

[48] La lógica de los paquetes se describe en 2.7.3

3.1.2.2.2.1 ESQUEMA PARA MODULO ALGORITMO GENETICO

Se guardan los atributos x que hace referencia al dato del vector de interdicción, p, la población de interdicción, caso hace referencia al número de nodos de la topología de red.

```
const { Schema, model } = require('mongoose');

const AlGen = new Schema({
    x: [Number],
    p: [[Number]],
    caso:String,
});
```

Ilustración 40. Esquema base de datos para algoritmo genético

3.1.2.2.2.2 ESQUEMA PARA MODULOS ANALISIS DE VULNERABILIDAD, FLUJO OPTIMO DE POTENCIA Y REPORTES

Se reutiliza el esquema de la siguiente ilustración donde x, hace referencia a los flujos óptimos de potencia por nodos (casos) y u hace referencia a las métricas usadas para gráficas resiliencia.

```
const { Schema, model } = require('mongoose');

const FlujoOpt = new Schema({
   x: Object,
   caso:String,
    u:Array,
});

module.exports = model("FlujoOptimo", FlujoOpt);
```

Ilustración 41. Esquema base de datos para módulos análisis de vulnerabilidad, flujo óptimo de potencia y reportes

Fuente: propia

4 FASE DE TRANSICIÓN

Se tienen los módulos de la aplicación[49] en su fase de transición[50] , para accederlos se propone el inicio de sesión con los servicios de Google[51] como se ilustra en la figura 42:

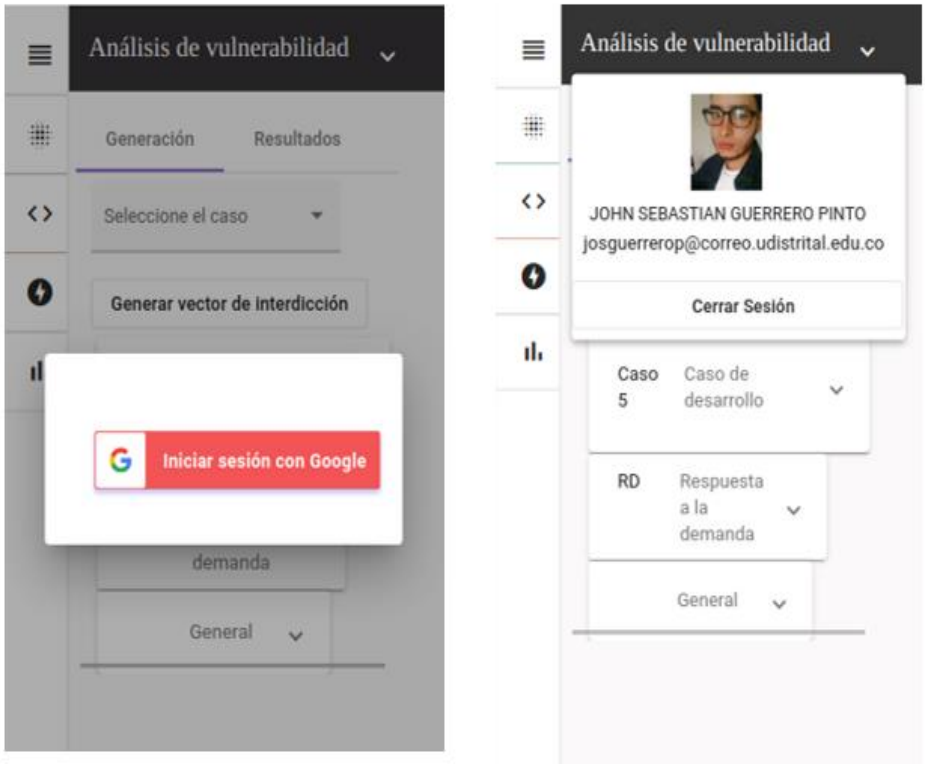


Ilustración 42. Inicio de sesión

Fuente: propia

Una vez se accede a la aplicación se utiliza un menú permanente para acceder a los módulos de la aplicación como se ilustra en la figura n y que forman los casos de uso generales de la aplicación 2.4.1. Como se muestra en la figura:

[49] El nombre de la aplicación se llama "Análisis de vulnerabilidad" como el módulo "Análisis de vulnerabilidad"
[50] Aplicación final utilizando la arquitectura planteada y los componentes. Se utiliza el caso 5 para ilustrar resultados.
[51] Se utiliza Social Login que usa los servicios de OAuth 2.0 para autorizar a los usuarios fácilmente

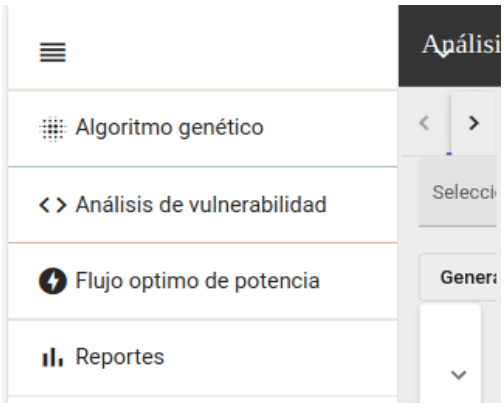


Ilustración 43. Menú de la aplicación

Fuente: propia

4.1 MODULO ALGORITMO GENÉTICO

El módulo algoritmo genético está compuesto por los casos de uso definidos en 2.4.2. cómo se observa en la ilustración :

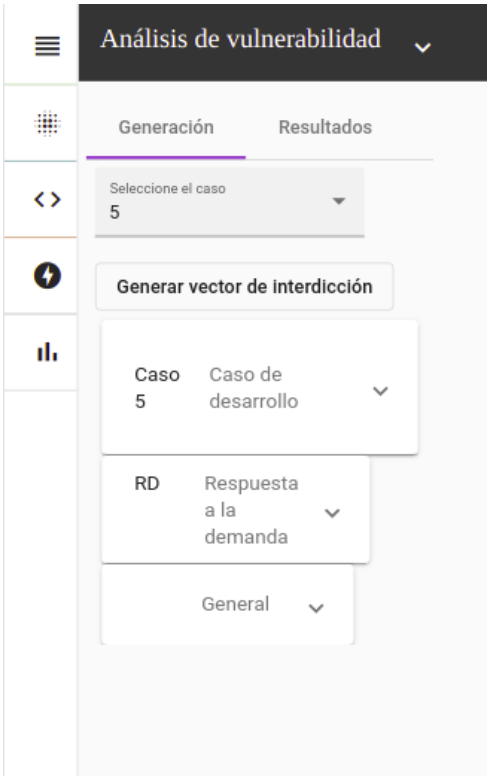
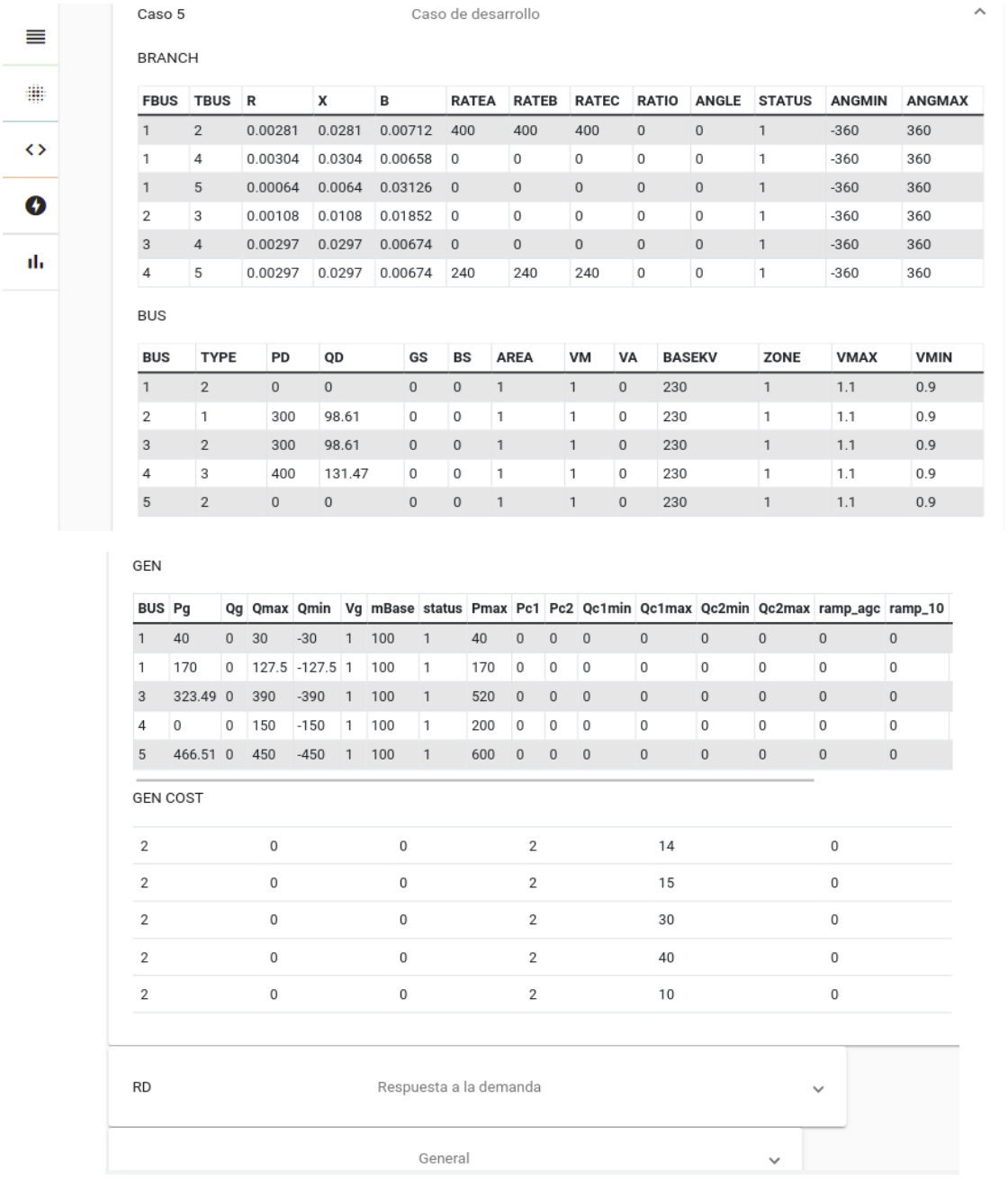


Ilustración 44. Modulo algoritmo genético

Fuente: propia

4.1.1 CASO DE DESARROLLO

Se observan resultados para los casos de uso , de acuerdo con el caso o topología de red seleccionada como se observa en la ilustración 45 y posteriores ilustraciones.



Caso 5 — Caso de desarrollo

BRANCH

FBUS	TBUS	R	X	B	RATEA	RATEB	RATEC	RATIO	ANGLE	STATUS	ANGMIN	ANGMAX
1	2	0.00281	0.0281	0.00712	400	400	400	0	0	1	-360	360
1	4	0.00304	0.0304	0.00658	0	0	0	0	0	1	-360	360
1	5	0.00064	0.0064	0.03126	0	0	0	0	0	1	-360	360
2	3	0.00108	0.0108	0.01852	0	0	0	0	0	1	-360	360
3	4	0.00297	0.0297	0.00674	0	0	0	0	0	1	-360	360
4	5	0.00297	0.0297	0.00674	240	240	240	0	0	1	-360	360

BUS

BUS	TYPE	PD	QD	GS	BS	AREA	VM	VA	BASEKV	ZONE	VMAX	VMIN
1	2	0	0	0	0	1	1	0	230	1	1.1	0.9
2	1	300	98.61	0	0	1	1	0	230	1	1.1	0.9
3	2	300	98.61	0	0	1	1	0	230	1	1.1	0.9
4	3	400	131.47	0	0	1	1	0	230	1	1.1	0.9
5	2	0	0	0	0	1	1	0	230	1	1.1	0.9

GEN

BUS	Pg	Qg	Qmax	Qmin	Vg	mBase	status	Pmax	Pc1	Pc2	Qc1min	Qc1max	Qc2min	Qc2max	ramp_agc	ramp_10
1	40	0	30	-30	1	100	1	40	0	0	0	0	0	0	0	0
1	170	0	127.5	-127.5	1	100	1	170	0	0	0	0	0	0	0	0
3	323.49	0	390	-390	1	100	1	520	0	0	0	0	0	0	0	0
4	0	0	150	-150	1	100	1	200	0	0	0	0	0	0	0	0
5	466.51	0	450	-450	1	100	1	600	0	0	0	0	0	0	0	0

GEN COST

2	0	0	2	14	0
2	0	0	2	15	0
2	0	0	2	30	0
2	0	0	2	40	0
2	0	0	2	10	0

RD — Respuesta a la demanda

General

Ilustración 45. Características caso de desarrollo

Fuente: propia

4.1.2 RESPUESTA A LA DEMANDA Y GENERAL

En la siguiente ilustración se observan parámetros de respuesta a la demanda.

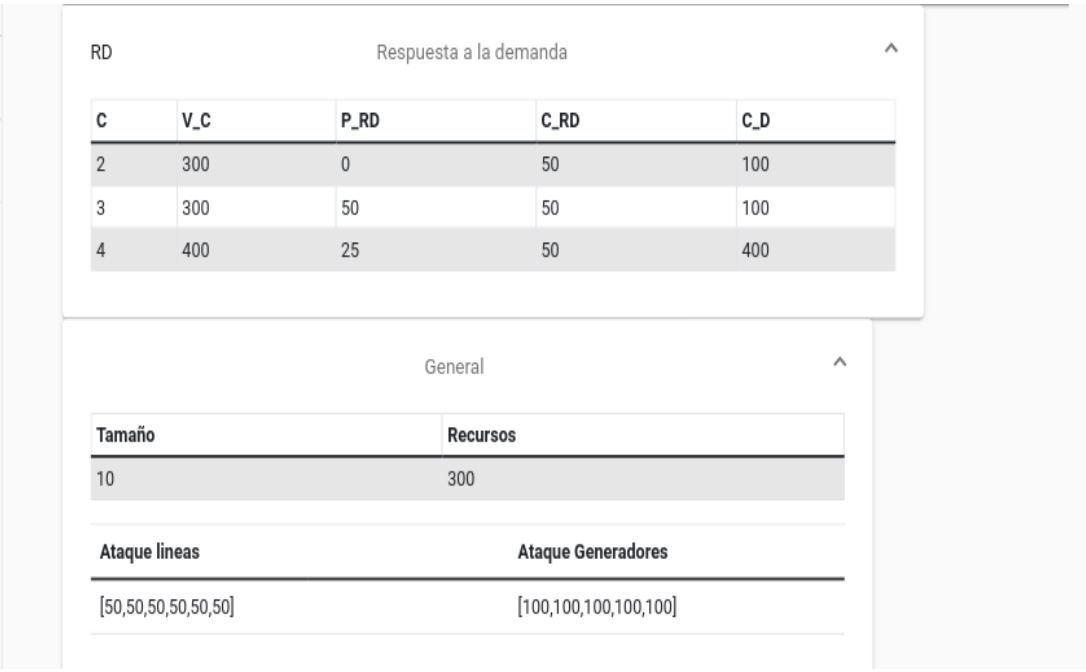


Ilustración 46. Parámetros respuesta a la demanda

Fuente: propia

4.1.3 GENERAR VECTOR DE INTERDICCIÓN

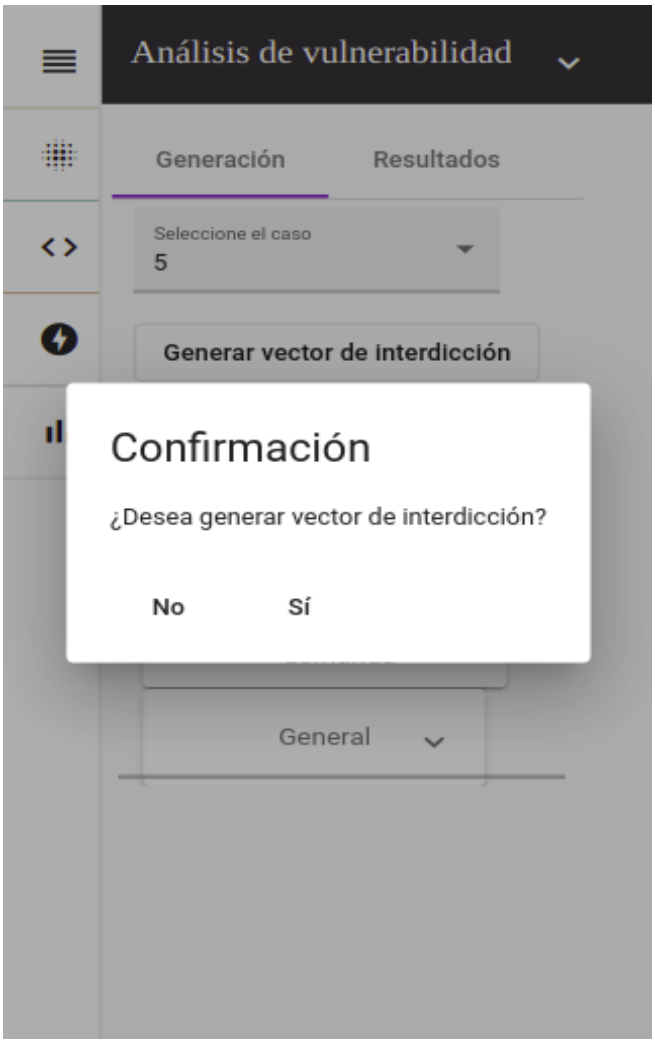


Ilustración 47. Generar vector de interdicción

Fuente: propia

4.1.4 RESULTADOS

Cuando se confirma la generación de un vector de interdicción , la petición se procesa en la instancia del servidor MPS “Algoritmo” 3.1.2.1 . Los resultados de generar un vector de interdicción se observan en la ilustración 48 donde se visualiza una población de interdicción y su costo final (última columna de la tabla) .



Ilustración 48. Resultados generación vector de interdicción

Fuente: propia

4.1.5 GUARDAR RESULTADOS

Cuando se realiza un análisis de vulnerabilidad saldrá una confirmación para guardar resultados en la base de datos como se observa en la ilustración 49.

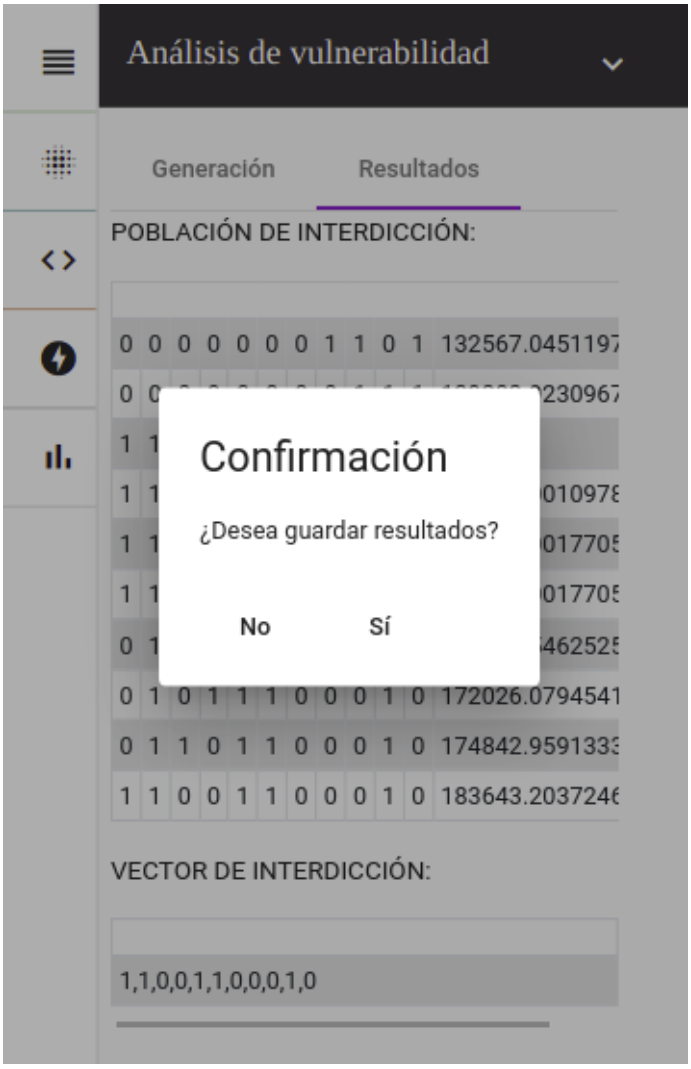


Ilustración 49. Guardar resultados

Fuente: propia

4.2 MODULO ANALISIS DE VULNERABILIDAD

4.2.1 ANÁLISIS

En la ilustración se observa un análisis de vulnerabilidad realizado donde podemos visualizar información de acuerdo con los casos de uso definidos en 2.4.3 como se ilustra en la figura 49.

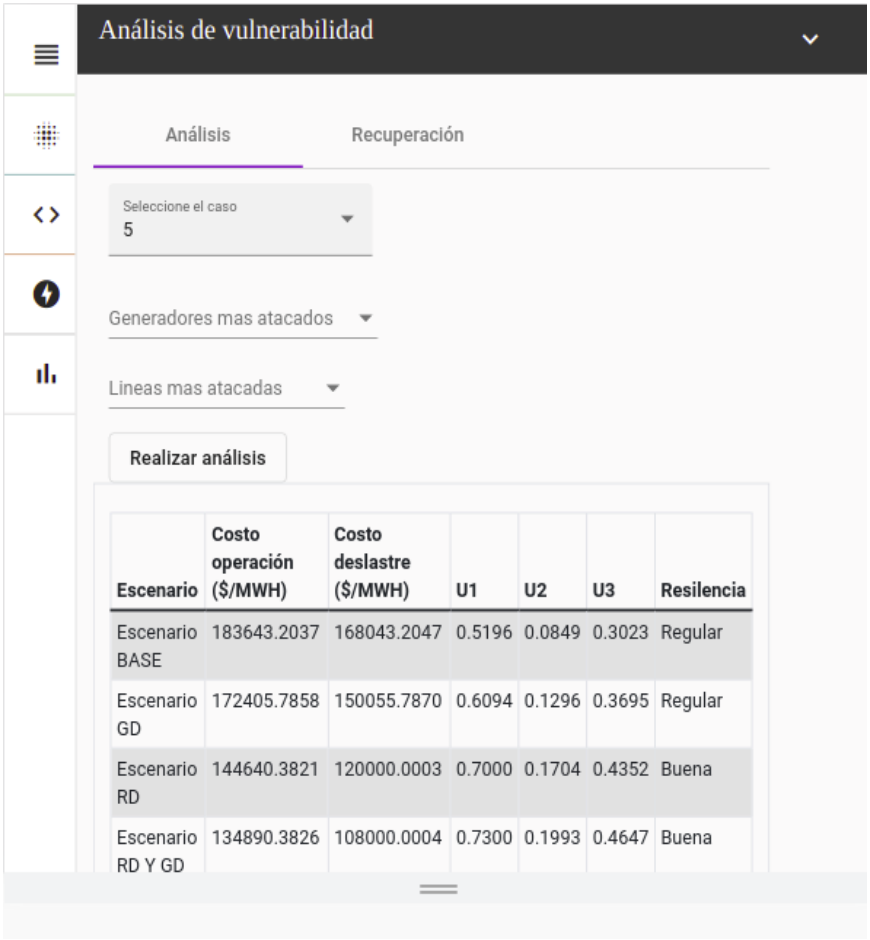


Ilustración 50. Modulo Análisis de vulnerabilidad

Fuente: propia

4.2.2 GUARDAR RESULTADOS

Cuando se termina de realizar un análisis de vulnerabilidad se pregunta al operador si desea guardar los datos como se visualiza en la ilustración , una vez confirmado se guardarán en la base de datos

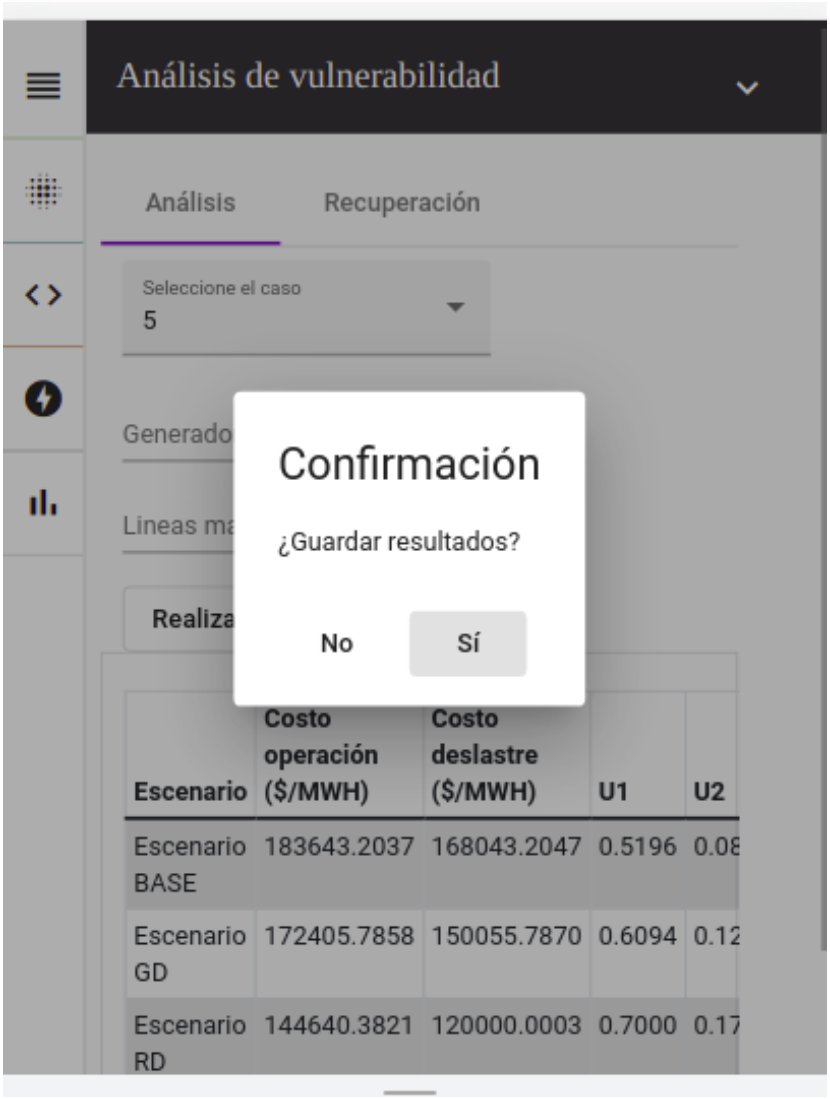


Ilustración 51. Guardar resultados

Fuente: propia

4.2.3 RECUPERACIÓN

En la pestaña recuperación se encuentran las métricas de cuando se recobra el sistema con un nuevo vector (Nueva red y distribución de energía).

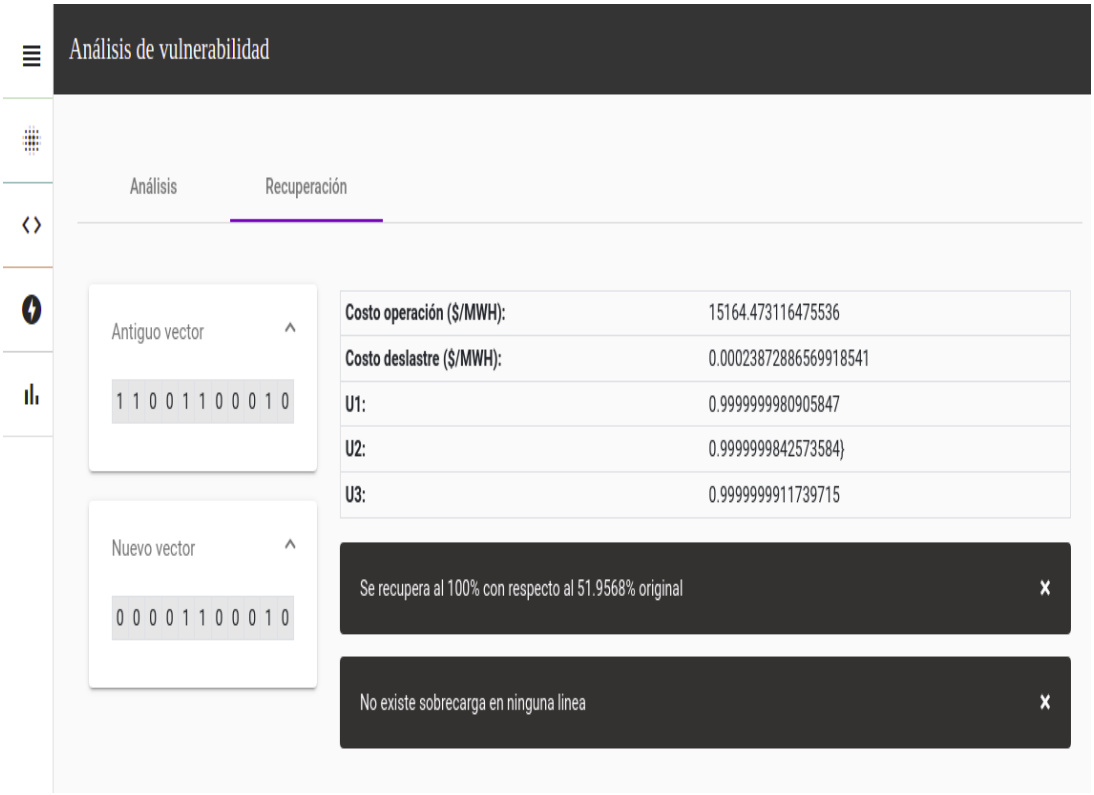


Ilustración 52. Métricas de recuperación

Fuente propia

4.3 MODULO FLUJO OPTIMO DE POTENCIA

El módulo flujo óptimo de potencia es de consulta de la información de un análisis de vulnerabilidad realizado para los flujos óptimos de potencia aplicados a los escenarios (Base, RD, GD, RD Y GD). En el módulo flujo óptimo de potencia se visualizan los casos de uso 2.4.4.1

4.3.1 BASE

En la pestaña base se encuentra el seleccionador del caso. Cuando se selecciona actualiza la información en cada escenario, donde se puede observar una tabla como se observa en la ilustración 51.

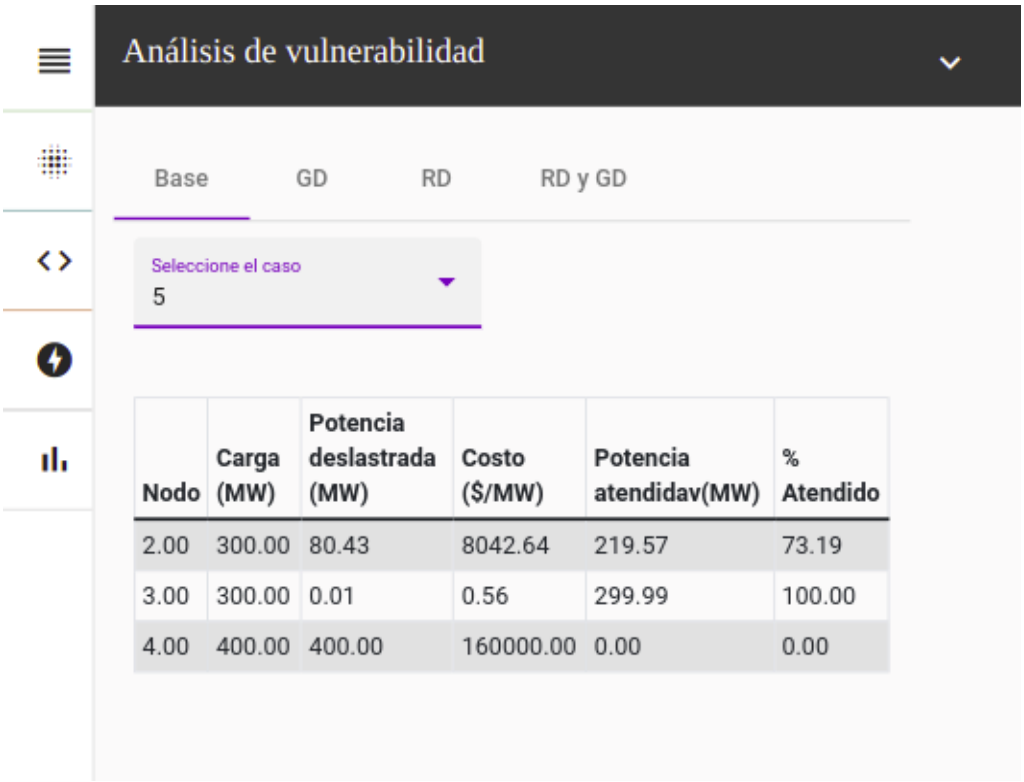


Ilustración 53. Escenario Base

Fuente: propia

4.3.2 GD

En la ilustración 52 se visualiza los nodos a los que se les distribuye la nueva energía al escenario GD.



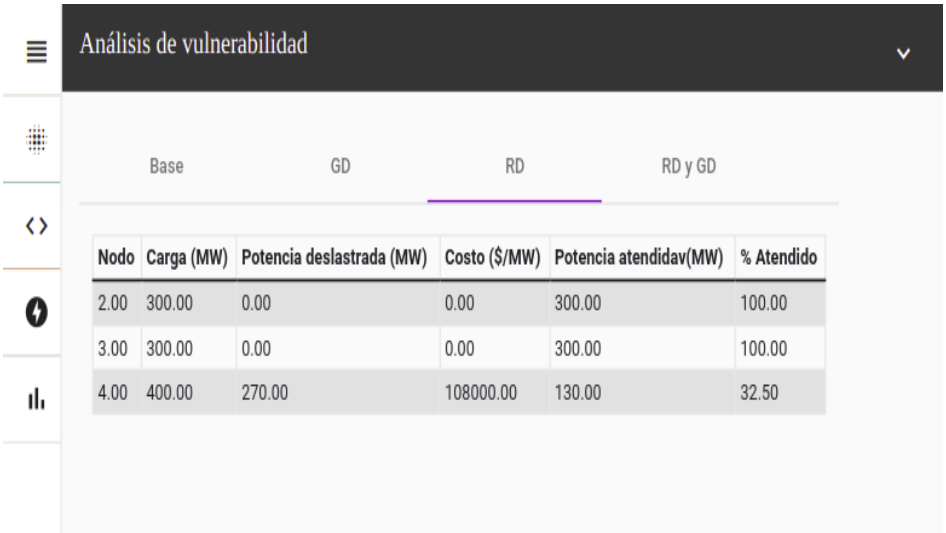
Nodo	Carga (MW)	Potencia deslastrada (MW)	Costo ($/MW)	Potencia atendidav(MW)	% Atendido
2	300	20.55765325225086	2055.7653252250857	279.4423467477491	93.1474489159163
3	300	0.00021672717580246747	0.021672717580246748	299.9997832728242	99.9999277576080
4	400	370	148000	30	7.5

Ilustración 54. Escenario GD

Fuente: propia

4.3.3 RD

En la ilustración 53 se visualiza los nodos a los que se les distribuye la nueva energía al escenario RD.



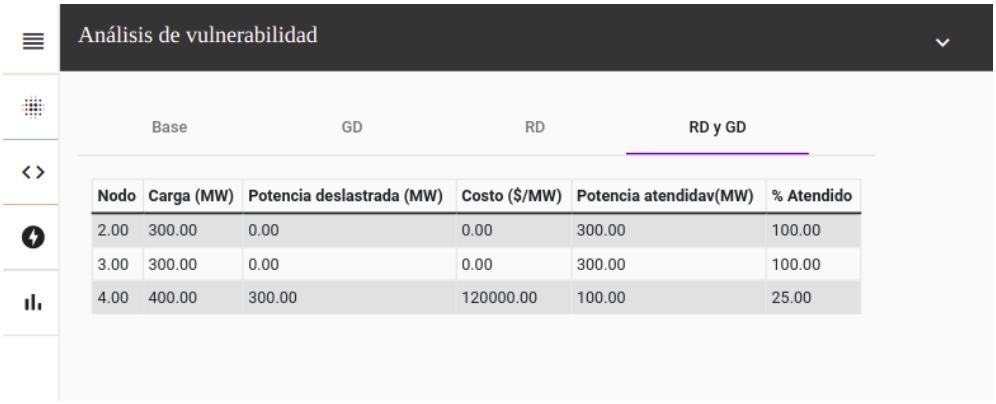
Nodo	Carga (MW)	Potencia deslastrada (MW)	Costo ($/MW)	Potencia atendidav(MW)	% Atendido
2.00	300.00	0.00	0.00	300.00	100.00
3.00	300.00	0.00	0.00	300.00	100.00
4.00	400.00	270.00	108000.00	130.00	32.50

Ilustración 55. Escenario RD

Fuente: propia

4.3.4 RD Y GD

En la ilustración 54 se visualiza los nodos a los que se les distribuye la nueva energía al escenario RD y GD.



Nodo	Carga (MW)	Potencia deslastrada (MW)	Costo ($/MW)	Potencia atendidav(MW)	% Atendido
2.00	300.00	0.00	0.00	300.00	100.00
3.00	300.00	0.00	0.00	300.00	100.00
4.00	400.00	300.00	120000.00	100.00	25.00

Ilustración 56. Escenario RD y GD

Fuente: propia

4.4 MODULO REPORTES

En el módulo al seleccionar caso se actualiza dinámicamente las gráficas de cargas atenidas y resumen de cara por nodos como se ilustra en la figura 55.

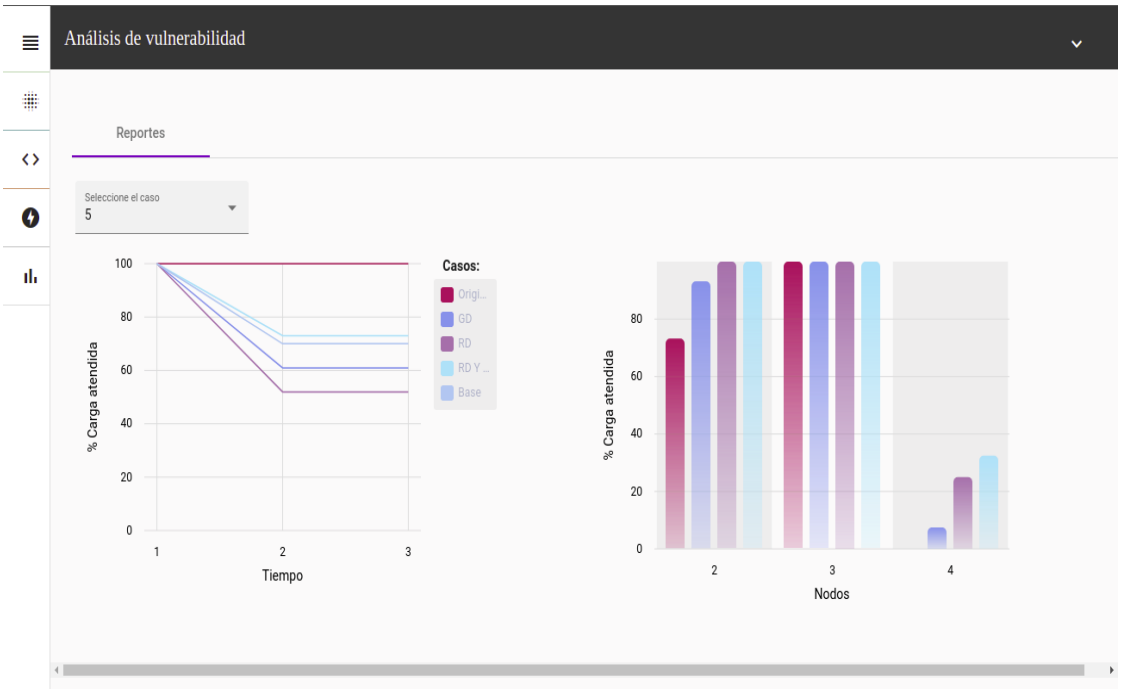


Ilustración 57. Modulo reportes

Fuente: propia

4.5 CONCLUSIONES

- Con Angular se permitió el modularidad de la aplicación , y la reutilización de componentes como seleccionar caso[52]. Se logró estilizar y hacer responsiva la aplicación.

  Angular facilitó hacer peticiones al servidor de base de datos y a Matlab Producción server para obtener , guardar y consultar los datos. Con el manejo de esta información fue posible realizar gráficas dinámicas. Lo anterior permite el cumplimiento de los objetivos de la aplicación.

- Las instancias de los servidores de Matlab permitieron la ejecución de algoritmos de Matlab con Matpower, sin embargo, es necesario optimizarlos para mejorar el rendimiento de la aplicación ya que cuando pasaron del proceso de programación a convertirlos en archivos .ctf aumentaron los tiempos de respuesta para servicios web[53].
- La base de datos tuvo un buen rendimiento tanto para guardar como para consultar los datos que se generan en los casos de uso. Permitió así mismo reutilizar esquemas en los módulos.

[52] AuxComponent definido en 2.6
[53] La creación de vectores de interdicción tardo más tiempo de lo esperado.

5 BIBLIOGRAFÍA

[1] A. Abedi, L. Gaudard, and F. Romerio, “Review of major approaches to analyze vulnerability in power system,” Reliab. Eng. Syst. Saf., vol. 183, no. November 2018, pp. 153–172, 2019, doi: 10.1016/j.ress.2018.11.019.

[2] M. Panteli and P. Mancarella, “Modeling and Evaluating the Resilience of Critical Electrical Power Infrastructure to Extreme Weather Events Resilience of electric power systems to earthquakes and tsunamis View project DIMMER View project,” Artic. IEEE Syst. J., vol. 11, no. 3, pp. 1733–1742, 2017, doi: 10.1109/JSYST.2015.2389272.

[3] R. Yarveisy, C. Gao, and F. Khan, “A simple yet robust resilience assessment metrics,” Reliab. Eng. Syst. Saf., vol. 197, no. January 2019, 2020, doi: 10.1016/j.ress.2020.106810.

[4] N. Bhusal, M. Abdelmalak, M. Kamruzzaman, and M. Benidris, “Power system resilience: Current practices, challenges, and future directions,” IEEE Access, vol. 8, pp. 18064–18086, 2020, doi: 10.1109/ACCESS.2020.2968586.

[5] Z. Bie, Y. Lin, G. Li, and F. Li, “Battling the Extreme: A Study on the Power System Resilience,” Proc. IEEE, vol. 105, no. 7, pp. 1253–1266, Jul. 2017, doi: 10.1109/JPROC.2017.2679040.

[6] M. Digitales, “LA DIGITALIZACIÓN EN EL SECTOR ENERGÍA TRANSFORMACIÓN TECNOLÓGICA Y ENERGÉTICA,” 2019.

[7] C.C. Liu, W. Sun, Development and Evaluation of System Restoration Strategies from a Blackout, 2009, PSERC Publication 09-08.

[8] J. P. H.-V. B. J. R.-C. J. M. López-Lezama, “A Bilevel Attacker-Defender Model for Enhancing Power Systems Resilience with Distributed Generation,” Sci. Tech., vol. 25, no. 4, pp. 540–547, 2020, doi: 10.22517/23447214.23721.

[9] L. F. M. van Summeren, A. J. Wieczorek, G. J. T. Bombaerts, and G. P. J. Verbong, “Community energy meets smart grids: Reviewing goals, structure, and roles in Virtual Power Plants in Ireland, Belgium and the Netherlands,” Energy Res. Soc. Sci., vol. 63, no. May, 2020, doi: 10.1016/j.erss.2019.101415.

[10] L. Maharjan, M. Ditsworth, M. Niraula, C. Caicedo Narvaez, and B. Fahimi, “Machine learning based energy management system for grid disaster mitigation,” IET Smart Grid, vol. 2, no. 2, pp. 172–182, Jun. 2019, doi: 10.1049/iet-stg.2018.0043.

[11] H. Tu, Y. Xia, C. K. Tse and X. Chen, "A Hybrid Cyber Attack Model for Cyber-Physical Power Systems," in IEEE Access, vol. 8, pp. 114876-114883, 2020, doi: 10.1109/ACCESS.2020.3003323.

[12] Heitzig, J., Fujiwara, N., Aihara, K. et al. Interdisciplinary challenges in the study of power grid resilience and stability and their relation to extreme weather events. Eur. Phys. J. Spec. Top. 223, 2383–2386 (2014). https://doi.org/10.1140/epjst/e2014-02281-0

[13] Zio, E. Critical Infrastructures Vulnerability and Risk Analysis. Eur J Secur Res 1, 97–114 (2016). https://doi.org/10.1007/s41125-016-0004-2

[14] D. H. Kim, D. A. Eisenberg, Y. H. Chun, and J. Park, “Network topology and resilience analysis of South Korean power grid,” Phys. A Stat. Mech. its Appl., vol. 465, pp. 13–24, 2017, doi: 10.1016/j.physa.2016.08.002.

[15] J. M. López-Lezama, J. C. C. Galeano, and E. R. Trujillo, “Assessment of the electric grid interdiction problem considering different network models,” J. Appl. Sci. Eng., vol. 23, no. 2, pp. 175–183, 2020, doi: 10.6180/jase.202006_23(2).0001.

[16] T. D. O’Rourke, “Critical Infrastructure, Interdependencies, and Resilience,” 2007. Accessed: Jul. 25, 2019. [Online]. Available: https://pdfs.semanticscholar.org/6c17/b35ec7555a9f27d5ccb6ca1d357a20b5ce0a.pdf.

[17] Abdullah Akram Bajwa, Hazlie Mokhlis, Saad Mekhilef, and Marizan Mubin. 2019.
Enhancing power system resilience leveraging microgrids: A review. J. Renew. Sustain.
Energy 11, 3 (May 2019), 035503. DOI:https://doi.org/10.1063/1.5066264

[18] c. Development, "Blending in the energy sector.", Op.europa.eu, 2021. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/987e6ca1-11ba-11e6-ba9a-01aa75ed71a1/language-en/format-PDF/source-195914046#. [Accessed: 21- Mar- 2021].

[19] Ramírez Castaño, S Redes de distribución de energía. [Internet]. Montevideo, Uruguay: Universidad de la RepÃºblica ; 2009. [citado: 2021, marzo] Universidad Nacional de Colombia Sede Manizales Facultad de Ingeniería y Arquitectura Departamento de Ingeniería Eléctrica, Electrónica y Computación.

[20] Jorge Iñesta Burgos Pedro Antonio García Fernández "LA ELECTRICIDAD EL RECORRIDO DE LA ENERGÍA" 2002 E.i.S.E. Domènech, S.A

[21] Panteli M, Mancarella P. Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies. Electr Power Syst Res 2015;127:259–70.

[22] David L. Pells "Disruptive Events! Are you, your project or your organization prepared?" PM World Journal 2009 Accessed: Jul. 25, 2019. [Online]. Available: https://pmworldlibrary.net/wp-content/uploads/2016/12/2009-Sept-Pells-Disruptive-Events-Are-you-prepared.pdf

[23] L. Gary, “Vulnerability Risk Assessment,” no. June, 2000.

[24] Cortina, Juan J., et al. "Metaheurísticas Aplicadas al Problema de Interdicción en Sistemas de Potencia." Informacion Tecnologica, vol. 29, no. 2, 2018, p. 73+. Gale OneFile: Informe Académico, . Accessed 12 Mar. 2021.

[25] A. Keller, Multi-Objective Optimization in Theory and Practice II. Sharjah: Bentham Science Publishers, 2019.

[26] D. Whitley, "A genetic algorithm tutorial", Statistics and Computing, vol. 4, no. 2, 1994. Available: 10.1007/bf00175354.