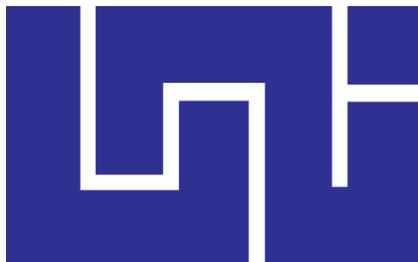


UNIVERSIDAD NACIONAL DE INGENIERÍA



Facultad de Electrotecnia y Computación (UNI-RUSB)

TRABAJO MONOGRÁFICO DE FIN DE CARRERA

Desarrollo de sistema web para determinar y visualizar la geolocalización de subestaciones eléctricas en Nicaragua.

AUTORES:

- Br. Castillo Hurtado Yader David **Carné:** 2015-0012U
- Br. Rivera Moreira Oscar Danilo **Carné:** 2015-0578U

TUTOR:

M. Sc. Narciso Aguilera Centeno.

ASESOR:

Ing. Víctor M. Hernández M.

Managua, mayo de 2021



Dedicatoria

Oscar Danilo Rivera Moreira:

A mis padres por ser las personas que me han apoyado no solo económicamente sino también por sus consejos, brindándome toda su confianza y su amor.

Y a todas aquellas personas que de alguna u otra manera me brindaron su apoyo.

Yader David Castillo Hurtado:

Quiero dedicar este trabajo a Dios por darme la vida, salud y fortaleza para continuar adelante en esta vida, con su amor y cariño nos ha dado el ánimo, la fuerza, sabiduría; para culminar esta investigación. *“Lo que es imposible para los hombres es posible para Dios”*. Lucas 18:27.

A mi familia, a mi madre Karla Hurtado por su amor, su apoyo incondicional y su ejemplo de dedicación, trabajo, paciencia y responsabilidad que la caracteriza; a mi padre Moisés Castillo por darme apoyo moral, ejemplo, cariño y sustento en todo momento; y a mis hermanas Masiel Castillo y Cinthia Castillo quienes son la luz de mis ojos por estar siempre dándome su ánimo en este camino.



Agradecimientos

Agradecemos a Dios por la vida, la salud, sabiduría y por habernos acompañado y guiado a lo largo de nuestra carrera, por ser nuestra fortaleza en momentos de debilidad y por brindarnos una vida llena de aprendizajes y experiencias.

A nuestras familias que estuvieron ahí en nuestras noches de desvelo y dedicación para la culminación de este enriquecedor camino, que no termina aquí, sino que emerge una bifurcación nueva.

A nuestro maestro Narciso Aguilera por habernos apoyado a lo largo del desarrollo de este trabajo con dedicación y por ayudarnos a crecer profesionalmente y aprender nuevas cosas. Y en especial al máster Adilson González que nos brindó su apoyo incondicional al inicio de esta jornada, proveyendo excepcional retroalimentación para el desarrollo de este proyecto.

Al ingeniero Víctor Hernández por su colaboración con la realización de este documento interdisciplinario y sus explicaciones en áreas que desconocíamos en el campo eléctrico.

A todos los maestros de la Universidad Nacional de Ingeniería que estuvieron involucrados en brindarnos la enseñanza requerida y adecuada a lo largo de estos 5 años de formación profesional y también por habernos brindado su amistad.



Resumen

El presente trabajo monográfico titulado “Desarrollo de sistema web para determinar y visualizar la geolocalización de subestaciones eléctricas en Nicaragua” tiene como finalidad describir el desarrollo de un prototipo de sistema web interdisciplinario para localizar el punto óptimo para la ubicación de una subestación eléctrica en Nicaragua. Este sistema web está enfocado en ser amigable y adaptable a los diversos dispositivos utilizados por el usuario, dicho sistema se elaboró aplicando la metodología de desarrollo UWE.

El objetivo general es desarrollar un prototipo de sistema web que permita determinar la ubicación geográfica óptima de subestaciones eléctricas aplicando un modelo matemático soportado por funciones de Liapunov.

El desarrollo del sistema inició con la recopilación de información brindada por parte de expertos en el tema de diseños de subestaciones eléctricas, obteniendo información de los procesos y algoritmos empleados para obtención de la elipse y su centro que indica dónde se debería construir dicha subestación. Se diseñó el prototipo de sistema web, se realizaron pruebas y se validó el desempeño y facilidad de uso en comparación de los procesos que actualmente se realizan; y el cumplimiento de los objetivos establecidos.



Índice de Contenido

I. INTRODUCCIÓN	11
II. ANTECEDENTES	12
III. JUSTIFICACIÓN	14
IV. OBJETIVOS	15
IV.1 Objetivo general	15
IV.2 Objetivos específicos	15
V. SUBESTACIONES ELÉCTRICAS	16
V.1 Definición	16
V.2 Localización de subestaciones eléctricas	16
V.3 Sistema Informático	26
V.3.1 Definición	26
V.3.2 Base de datos	26
V.3.3 Modelo Relacional	27
V.3.4 Aplicación Web	28
V.3.5 Modelo Cliente Servidor	28
V.4 Metodología del desarrollo de Software	30
V.4.1 Ciclo de vida	30
V.4.2 UML	32
V.4.3 Metodología UWE	33
V.4.4 Arquitectura MVC	35
V.5 Herramientas de desarrollo	36
V.5.1 HTML5	36
V.5.2 CSS	37
V.5.3 JavaScript	37
V.5.4 JQuery	38
V.5.5 Bootstrap	38
V.5.6 Fabric JS	38
VI. DISEÑO METODOLÓGICO	40
VI.1 Estudio de Prefactibilidad	41
VI.1.1 Factibilidad Técnica	41
VI.1.2 Factibilidad Económica	45



VI.1.3 Factibilidad Operativa	46
VI.1.4 Factibilidad Legal	47
VI.1.4.1 Requerimientos legales	47
VI.1.4.2 Las licencias para el software	47
VI.1.4.3 Propiedad Intelectual	47
VI.2 Captura, análisis y especificación de requerimientos	48
VI.2.1 Actores	48
VI.2.2 Requerimientos Funcionales	48
VI.2.3 Requerimientos No Funcionales	49
VI.2.4 Requerimientos de Interfaz	51
VI.2.5 Restricciones de Diseño	52
VI.2.6 Requerimientos de Licencia	53
VI.3 Diseño del Sistema	53
VI.3.1 Modelo de casos de uso	53
VI.3.2 Modelo conceptual	68
VI.3.3 Modelo de espacio de navegación y modelo de estructura de navegación ..	69
VI.3.4 Modelo de presentación	80
VI.4 Codificación del software	85
VI.4.1 Lenguaje de programación y frameworks	86
VI.4.2 Base de datos	92
VI.4.3 Seguridad	92
VI.5 Pruebas	94
VI.5.1 Evolución del sistema	94
VI.5.2 Control de versiones	97
VI.5.3 Pruebas unitarias	97
VI.6 Caso de Prueba	99
VI.7 Evaluación de resultados	102
VI.7.1 Desempeño	102
VI.7.2 Facilidad de uso	103
VI.7.3 Proceso de evaluación de pruebas	103
VI.7.4 Evaluación de desempeño	104
VI.7.5 Evaluación de facilidad de uso	104



VI.8 Implementación y despliegue	105
VII. CRONOGRAMA	110
VIII. CONCLUSIONES	111
IX. RECOMENDACIONES	112
X. REFERENCIA BIBLIOGRÁFICAS	113
XI. GLOSARIO DE TÉRMINOS	115
XII. ANEXOS	116



Índice de ilustraciones

Ilustración 1 - Superficie de distribución normal	19
Ilustración 2 - Proyección del seccionamiento transversal de la superficie curva de distribución normal.	19
Ilustración 3 - Ejemplo de esquema del modelo relacional.....	27
Ilustración 4 - Puesta en escena del modelo cliente/servidor.	29
Ilustración 5 - Ciclo de vida Lineal.....	31
Ilustración 6 - Ciclo de vida en V.....	31
Ilustración 7 - Ciclo de vida en espiral.....	32
Ilustración 8 - Etapas de la metodología UWE.....	33
Ilustración 9 - Arquitectura MVC	36
Ilustración 10 - Estructura del sistema	40
Ilustración 11 - DCU - Graficar Elipse	54
Ilustración 12 - DCU - Guardar espacio de trabajo.....	55
Ilustración 13 - DCU - Cargar datos masivamente	56
Ilustración 14 - DCU - Generar superficie normal.....	57
Ilustración 15 - DCU - Generar gráfica de consumidores	58
Ilustración 16 - DCU - Descargar plantilla de carga de datos	59
Ilustración 17 - DCU - Limitar Mapa	59
Ilustración 18 - DCU - Quitar limitación de Mapa	60
Ilustración 19 - DCU - Reiniciar.....	61
Ilustración 20 - DCU - Poner pin de prueba	61
Ilustración 21 - DCU - Configurar vista del Mapa	62
Ilustración 22 - DCU - Configurar vista proyección.....	63
Ilustración 23 - DCU - Configurar vista global	64
Ilustración 24 - DCU - Generar reportes.....	65
Ilustración 25 - DCU - Visualizar reportes	66
Ilustración 26 - DCU - Visualizar proyecciones	67
Ilustración 27 - DCU - Configurar el sistema	68
Ilustración 28 - DC – GLOCSE	69
Ilustración 29 - DN - Módulo Mapa.....	70
Ilustración 30 - DN - Módulo Gráficas	71
Ilustración 31 - DN - Módulo Reportes	72



Ilustración 32 - DN - Módulo Proyecciones	73
Ilustración 33 - DN - Módulo Configuración.....	74
Ilustración 34 - DS - Autenticación de Usuario	75
Ilustración 35 - DS - Creación de proyectos.....	76
Ilustración 36 - DS - Creación de proyección	77
Ilustración 37 - DS - Creación de múltiples proyecciones	78
Ilustración 38 - DS - Generación de reporte.....	79
Ilustración 39 - Vista Inicio de sesión	80
Ilustración 40 - Vista Menú Principal	81
Ilustración 41 - Vista Configuración Menú Principal	81
Ilustración 42 - Vista Opciones de Usuario	82
Ilustración 43 - Vista Módulo Gráficas.....	82
Ilustración 44 - Vista Módulo Reportes	83
Ilustración 45 - Vista Módulo Proyecciones.....	83
Ilustración 46 - Vista Módulo Configuración	84
Ilustración 47 - Modelo arquitectónico.....	85
Ilustración 48 - Descargar Visual Studio Community.....	86
Ilustración 49 - Creación de directorio con nombre de Proyecto	86
Ilustración 50 - Creación de un repositorio.....	87
Ilustración 51 - Dirección del repositorio creado.....	87
Ilustración 52 - Clonación de proyecto	88
Ilustración 53 - Creación de proyecto en Visual Studio	88
Ilustración 54 - Selección tipo de proyecto en Visual Studio	89
Ilustración 55 - Configuración de proyecto en Visual Studio.....	89
Ilustración 56 - Selección de Aplicación Web.....	90
Ilustración 57 - Proyecto creado satisfactoriamente	90
Ilustración 58 - Vista de los principales componentes de un proyecto ASP.net MVC.....	91
Ilustración 59 - Protección CSRF implementada.....	93
Ilustración 60 - Cadena de conexión a Base de Datos.....	93
Ilustración 61 - Manejo de excepciones en el sistema	93
Ilustración 62 - Se muestran algunos cambios realizados en la elaboración del prototipo final	97
Ilustración 63 - Pruebas Unitarias	98



Ilustración 64- Tabla de Cargas eléctricas de Caso de Prueba.....	99
Ilustración 65 - Ubicación estimada de la elipse.....	101
Ilustración 66 - Gráfica de Elipse con datos proporcionados.....	102
Ilustración 67 - Proceso para publicar una aplicación	105
Ilustración 68 - Sección de implementación de Visual Studio.....	106
Ilustración 69 - Destinos de implementación que nos brinda Visual Studio para nuestras aplicaciones.....	106
Ilustración 70 - Sub destinos de una aplicación en Azure	107
Ilustración 71 - Configuración de instancia.....	107
Ilustración 72 - Resumen de implementación	108
Ilustración 73 - Mensaje de implementación exitosa	108
Ilustración 74 - Sistema desplegado en Azure	109
Ilustración 75 - Cronograma del proyecto	110
Ilustración 76 - Constancia de participación en evento COMPDES.....	116
Ilustración 77 - Certificado de participación - Oscar Rivera.....	117
Ilustración 78 - Certificado de participación - Yader Castillo.....	118



Índice de Tablas

Tabla 1 - Aspectos técnicos de hardware.	44
Tabla 2 - Aspectos técnicos de software.....	44
Tabla 3 - Costos de hardware.....	45
Tabla 4 - Costos de software.	45
Tabla 5 - Costos de desarrollo.	46
Tabla 6 - Costos totales.	46
Tabla 7 - Descripción de CU - Graficar Elipse.....	54
Tabla 8 - Descripción de CU - Guardar espacio de trabajo	55
Tabla 9 - Descripción de CU - Cargar datos masivamente	56
Tabla 10 - Descripción de CU - Generar superficie normal	57
Tabla 11 - Descripción de CU - Generar gráfica de consumidores.....	58
Tabla 12 - Descripción de CU - Descargar plantilla de carga de datos	59
Tabla 13 - Descripción de CU - Limitar Mapa	60
Tabla 14 - Descripción de CU - Remover limitación de Mapa	60
Tabla 15 - Descripción de CU - Reiniciar	61
Tabla 16 - Descripción de CU - Poner pin de prueba.....	62
Tabla 17 - Descripción de CU - Configurar vista del mapa.....	62
Tabla 18 - Descripción de CU - Configurar vista proyección	63
Tabla 19 - Descripción de CU - Configurar vista global.....	64
Tabla 20 - Descripción de CU - Generar reportes	65
Tabla 21 - Descripción de CU - Visualizar reportes.....	66
Tabla 22 - Descripción de CU - Visualizar Proyecciones	67
Tabla 23 - Descripción de CU - Configurar el sistema.....	68
Tabla 24 - Coordenadas en Mapa para consumidores.....	100
Tabla 25 - Evaluación de desempeño.....	104
Tabla 26 - Evaluación de facilidad de uso.....	105



I. INTRODUCCIÓN

Este documento describe la propuesta de desarrollo de un sistema web que implementa un prototipo para una aplicación que, basada en un algoritmo matemático, permite la localización conveniente de subestaciones eléctricas (SE), para su posterior construcción. Con el fin de facilitar este cálculo y los distintos aspectos que conlleva nace el proyecto GLOCSE¹, una herramienta que permitirá calcular la localización óptima de SE².

Dicha herramienta contemplará una visualización interactiva en un plano de coordenadas 2D, un área delimitada que servirá para realizar los cálculos y estudios donde será ubicada la subestación eléctrica. En esta misma se ubicarán a los consumidores³ y centros de cargas⁴, para posteriormente calcular los otros elementos importantes (centros eventuales de carga, el punto de localización⁵ y la zona de dispersión, entre otros).

El proyecto GLOCSE toma en cuenta el ciclo general de desarrollo de software con la metodología UWE. Los detalles se pueden consultar en el apartado Metodología de Software.

Como solución se propone desarrollar un prototipo de sistema web, ya que asegura la compatibilidad con cualquier sistema operativo y teléfono móvil.

¹ GLOCSE: Geolocalización de Subestaciones eléctricas.

² SE: Subestaciones eléctricas.

³ Consumidores: Hace referencia a proveedores de energía (transformadores) que se encuentren en hospitales, parques, residenciales, entre otros.

⁴ Centro de carga: Energía consumida por consumidor en un tiempo delimitado de una hora.

⁵ Punto de Localización: Ubicación en donde será construida la subestación eléctrica.



II. ANTECEDENTES

Según declaraciones del presidente de la Empresa Nacional de Trasmisión Eléctrica, Nicaragua ha alcanzado un índice de electrificación del 97.69% hasta el mes de mayo del año en curso y tiene como objetivo lograr alcanzar un 98.42% para finales del 2020. Atendiendo a más de 1,000 habitantes con los proyectos San Nicolás, Santo Tomás del Norte, Rancho Grande, Nandasmo, Tola y Matagalpa [1]. No obstante, hacemos la observación que no se encontró un estudio completo actualizado que soportara estas declaraciones y datos estadísticos.

Datos previos, fueron encontrados en [2] donde el presidente del Ministerio de Energía y Minas establece que la cobertura a nivel nacional en ese entonces era del 80%.

Con el fin de profundizar la comprensión del cómo se realiza la ubicación de las subestaciones eléctricas en Nicaragua se realizaron entrevistas a especialistas de la UNI y ENATREL en el ramo energético. Como resultado se logró concluir que la forma tradicional de la construcción de subestaciones eléctricas ha sido mediante el cálculo de algoritmos matemáticos y estadísticos que permiten determinar la ubicación geográfica óptima de esta, tomando en cuenta diversos elementos, así como consumo eléctrico por diferentes usuarios a cada hora del día y posiciones geográficas de los mismos.

El principal factor para tomar en cuenta es la potencia eléctrica consumida. A continuación, se crean proyecciones realizando diversos cálculos hasta que se encuentra la ubicación más adecuada. Naturalmente estos procesos conllevan una cantidad razonable de tiempo, y esto se debe al gran volumen de cálculos que se realizan. A veces podría tomar hasta un par de días, en dependencia del lugar y la cantidad de consumidores a tomar en consideración.

Encontramos investigaciones y softwares que proveen pautas para la construcción de subestaciones eléctricas tales como:

- AutoCAD Electrical software for substations, de Autodesk para el diseño de circuitos eléctricos.



- Substation Explorer design software, de Utility Products que construye múltiples diseños conceptuales para sistemas de altos voltaje.
- Substation Design Suite, de Spatial Business Systems, software que realiza el análisis del terreno donde se construirá la SE, la cual es una de sus funcionalidades.

Sin embargo, el resultado de nuestra búsqueda señala que actualmente no se cuenta con aplicaciones y/o sistema web que realicen la función que GLOCSE ofrece. Por otro lado, ENATREL no especifica en sus informes o sitios webs la utilización de software alguno que determine la ubicación de las SE.



III. JUSTIFICACIÓN

La ubicación de una subestación eléctrica depende de la complejidad de la zona en la que se hará el análisis. Puede llegar a ser un procedimiento problemático y, sobre todo, lento. Es de gran utilidad contar con una aplicación con la cual se puedan realizar un sin número de proyecciones, tomando en cuenta un mapa real y actualizado, que muestre los consumidores aledaños a una zona estipulada. Se deben tomar en cuenta los diferentes factores geográficos mediante una forma visual de mapas. La idea es que el usuario pueda crear distintas proyecciones con diferentes configuraciones del entorno. De esta manera, podrá elegir a su conveniencia el punto geográfico óptimo en base a las necesidades de consumo de dicha zona y la empresa.

Por otra parte, no debemos olvidar el factor económico. La aplicación dispondrá de herramientas para evaluar proyecciones a lo largo de los años; esto lo hará mediante la configuración de proyecciones a lo largo del tiempo. En este caso se entiende que el usuario hará distintas proyecciones sobre la misma zona en base a las variaciones que habrá alrededor de los años (dichas variaciones incluyen, por ejemplo, construcción de nuevos hospitales, zonas residenciales, fabricas etc.). La aplicación al final deberá sugerir la mejor zona tomando en cuenta estas proyecciones en el tiempo y que al final proveerán beneficios mayores a la empresa que la utilice.

Con el desarrollo de este software se pretende crear una solución eficaz y rápida, que tome en cuenta todos los factores previamente mencionados y encuentre la mejor ubicación geográficamente para una mejor optimización de gastos económicos en el tiempo. Para este desarrollo, se utilizará el Framework **ASP.Net Core MVC** ocupando ciertas librerías tales como **Google Maps API**, **Chart JS**, **Fabric.JS** y **SQL Server**.



IV. OBJETIVOS

IV.1 Objetivo general

Desarrollar un prototipo de sistema web que permita determinar la ubicación geográfica óptima de subestaciones eléctricas aplicando un modelo matemático soportado por funciones de Liapunov.

IV.2 Objetivos específicos

- 1) Analizar un modelo matemático basado en funciones de Liapunov que sea aplicable en la ubicación geográfica de subestaciones eléctricas.
- 2) Codificar un algoritmo que determine la ubicación geográfica óptima o alternativa más conveniente de SE.
- 3) Diseñar el sistema mediante herramientas de modelado y diagramas de UML⁶, usando la metodología UWE.
- 4) Programar la aplicación empleando una arquitectura MVC⁷, mediante la tecnología ASP.Net Core MVC y con apoyo de tecnologías de clientes tales como JavaScript, JQuery, Bootstrap, Google Maps Y Fabric.JS.

⁶ UML: Unified Modeling Language (Lenguaje unificado de modelado)

⁷ MVC: Modelo, Vista, Controlador.



V. SUBESTACIONES ELÉCTRICAS

V.1 Definición

En [3] se establece: “Las subestaciones eléctricas son estaciones de transformación y distribución de la energía eléctrica que se produce en los centros de generación y se transmite a través de las redes de transporte de los sistemas eléctricos de potencia.”

Una correcta ubicación de la subestación garantiza una máxima confiabilidad, flexibilidad en operaciones, alto nivel de continuidad de servicio y permite el rápido restablecimiento del servicio después que ocurra una falla. Sin omitir una reducción de costos en cuanto a material se refiere. Con la ubicación racional de las subestaciones eléctricas, los índices técnico-económicos del sistema de suministro resultan óptimos y, consecuentemente garantizan el mínimo de gastos anuales.

V.2 Localización de subestaciones eléctricas

A) Determinación del centro condicional de cargas eléctricas.

El algoritmo matemático implementado para la localización de las subestaciones eléctricas fue originalmente desarrollado por A. A. Feodorov y Eduardo Rodríguez López, los cuales escribieron el libro Suministro Eléctrico de Empresas Industriales que se tomará como referencia bibliográfica a lo largo de este documento. El problema de fundamental importancia en la proyección de los sistemas de suministro electro energético de empresas industriales lo constituye la determinación de la localización óptima de las subestaciones de alimentación.

La aplicación de criterios cualitativos en el análisis de este problema regularmente conduce hacia soluciones que no cuentan con el amparo de una adecuada fundamentación técnico-económica.

Feodorov afirma que la ubicación de S.E. no necesariamente tendrá que coincidir con un centro eventual de carga eléctrica en [4, p. 114]: “Por otra parte,



la suposición de que la subestación será correctamente ubicada cuando se le haga coincidir con el centro de cargas eléctricas, considerando éste como un punto estático en el que pudiera suponerse concentrada toda la carga, es también errónea. “

El determinar el centro de cargas de un conjunto de consumidores independientes, dará como resultado una diferente localización de éste para cada instante de tiempo considerado. Por ello, más que de un centro estático de cargas, debe hablarse de centros eventuales de carga, ya que dependerá de la unidad de tiempo considerada y cuya ocurrencia deberá ser analizada mediante el algoritmo a implementar.

La temática consultada [5] indica que las coordenadas (X_i, Y_i) de los centros eventuales de carga constituyen los valores numéricos que adoptan las variables aleatorias X , e Y . Éstas, a su vez forman lo que en teoría de las probabilidades se denomina una magnitud bidimensional o sistema de dos variables aleatorias. Un correcto análisis demuestra que, con una probabilidad confiable, las coordenadas (X_i, Y_i) ocurren dentro de una elipse, que tiene como coordenadas de su centro las dadas por la esperanza matemática de la magnitud bidimensional (X, Y) cuyos parámetros pueden ser evaluados.

La localización de la subestación dentro de la zona delimitada por la elipse, también denominada zona de dispersión del centro de cargas eléctricas estará siempre avalada por una mayor ventaja o conveniencia económica, a la vez que posibilita la construcción de un sistema de suministro más confiable desde el punto de vista técnico. Con ello, se reduce la longitud de los circuitos de voltaje secundario, se reducen las pérdidas de energía y desviaciones de voltaje, se reduce la zona de fallas, etcétera.

Desde un sentido algo más práctico, estas consideraciones pueden interpretarse como un conjunto n de consumidores independientes, que habrán de ser alimentados a través de una subestación principal reductora, la cual se quiere localizar óptimamente.



Para cada consumidor se conoce el gráfico de cargas (escalonado, con intervalos de una hora), determinado durante el día característico. Por analogía, es posible afirmar que las coordenadas de los centros eventuales de cargas (en el caso considerando, uno por cada hora del día) vendrán dadas por las expresiones siguientes:

$$x_i = \frac{\sum_{j=1}^n P_{ij} X_j}{\sum_{j=1}^n P_{ij}} \quad (1)$$

$$y_i = \frac{\sum_{j=1}^n P_{ij} Y_j}{\sum_{j=1}^n P_{ij}} \quad (2)$$

Donde:

P_{ij} = potencia de cada consumidor J , en la hora ($I = 1, 2, \dots, 24$);

X_i, Y_j = coordenadas de cada consumidor;

n = número total de consumidores.

Como lo demuestra la práctica, el cálculo de una tercera coordenada Z_i no resulta necesario. La ocurrencia de los centros eventuales de carga, sobre el plano general de la instalación, está sometida a una gran cantidad de factores que la hacen fortuita.

Según [6], [7] al tener en cuenta el teorema de Liapunov y como resultado de las investigaciones realizadas, puede afirmarse que la magnitud bidimensional X, Y constituida por las coordenadas X_i y Y_i de los centros eventuales de carga, está subordinada a la ley normal de distribución (ley de Gauss-Laplace).

En su forma más genérica, la ley normal de distribución para una magnitud bidimensional (X, Y) se define:

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-r^2}} e^{-\frac{1}{2(1-r^2)}\left[\frac{(x-a_x)^2}{\sigma_x^2} - \frac{2r(x-a_x)(y-a_y)}{\sigma_x\sigma_y} + \frac{(y-a_y)^2}{\sigma_y^2}\right]} \quad (3)$$



Donde:

A_x, A_y = esperanzas matemáticas de las coordenadas eventuales;

σ_x, σ_y = dispersiones de las coordenadas eventuales;

r = factor de correlación de las coordenadas eventuales.

La expresión (2) representa una superficie de distribución, con forma de colina cuya mayor altura se encuentra en el punto (A_x, A_y) (Fig. 1)

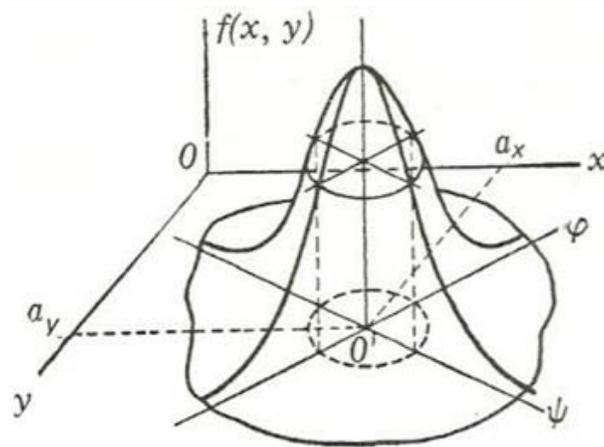


Ilustración 1 - Superficie de distribución normal

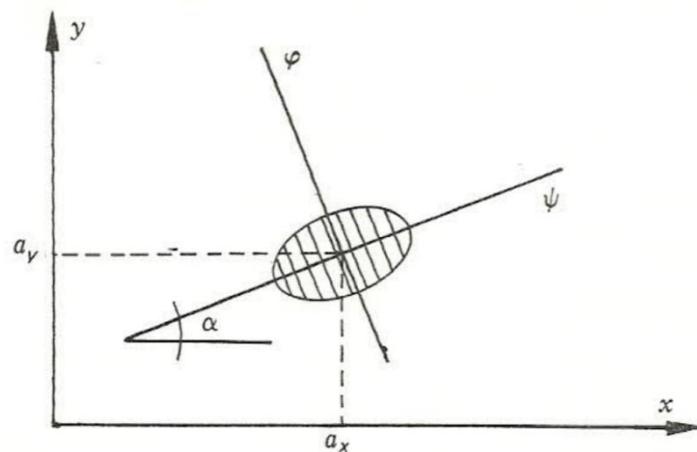


Ilustración 2 - Proyección del seccionamiento transversal de la superficie curva de distribución normal.



Con el seccionamiento de la superficie de distribución, según planos paralelos al eje $f(x, y)$, se obtienen curvas semejantes a las curvas normales de distribución. El seccionamiento de la superficie de distribución, por planos paralelos al plano xoy , proporcionará una familia de elipses, cuyo centro común se encuentra en el punto (A_x, A_y) véase Fig. 2.

Frecuentemente, durante la determinación de la ley teórica de distribución, las esperanzas matemáticas, dispersiones y factor de correlación resultan desconocidas y para su determinación se supone que ellas coinciden con los correspondientes valores de la distribución empírica. [4, p. 117]

Las características numéricas de la distribución empírica se determinan por las expresiones siguientes:

Esperanzas matemáticas:

$$a_x = \sum_{k=1}^n x_k p_{kx}; \quad a_y = \sum_{k=1}^n y_k p_{ky} \quad (4)$$

Donde:

P_{kx}, P_{ky} = probabilidad empírica de ocurrencia del valor x, y .

A simple vista se aprecia que la esperanza matemática puede ser interpretada como el valor medio de los valores empíricos de las magnitudes aleatorias, así:

$$a_x = \sum_{i=1}^n x_i / n \quad a_y = \sum_{i=1}^n y_i / n \quad (5)$$

Dispersión = Esperanza matemática del cuadrado de la desviación de la magnitud aleatoria respecto de su esperanza matemática.

$$D_x = \sum_{k=1}^n p_{kx} (x_k - a_x)^2; \quad D_y = \sum_{k=1}^n p_{ky} (y_k - a_y)^2 \quad (6)$$



O, lo que es lo mismo:

$$D_x = \sum_{i=1}^n \frac{(x_i - a_x)^2}{n}; \quad D_y = \sum_{i=1}^n \frac{(y_i - a_y)^2}{n} \quad (7)$$

Está probado, sin embargo, que cuando se trata de inferir resultados generales a partir de una pequeña muestra, como es el caso, las dispersiones obtenidas por medio de las expresiones **5** resultan ligeramente menores que las verdaderas. Se sugiere, entonces, introducir una modificación consistente en cambiar el divisor n por $n - 1$, resultando las expresiones siguientes:

$$D_x = \sum_{i=1}^n \frac{(x_i - a_x)^2}{n - 1}; \quad D_y = \sum_{i=1}^n \frac{(y_i - a_y)^2}{n - 1} \quad (8)$$

La dimensión de la dispersión es la de la magnitud aleatoria, elevada al cuadrado. Para facilitar el tratamiento dimensional, se define la variación cuadrática media, dada por:

$$\sigma_x = \sqrt{D_x} \quad \sigma_y = \sqrt{D_y} \quad (9)$$

De donde se definen las medidas de exactitud de las variables aleatorias:

$$h_x = \frac{1}{\sigma_x \sqrt{2}} \quad h_y = \frac{1}{\sigma_y \sqrt{2}} \quad (10)$$

Momento de correlación: esperanza matemática del producto de las desviaciones de las magnitudes aleatorias respecto de sus esperanzas matemáticas:

$$C_{xy} = \sum_{i=1}^n \frac{(x_i - a_x)(y_i - a_y)}{n} \quad (11)$$



En este caso resulta igualmente válida la corrección anterior, por lo que el momento de correlación vendrá dado por:

$$C_{xy} = \sum_{i=1}^n \frac{(x_i - a_x)(y_i - a_y)}{n - 1} \quad (12)$$

El momento de correlación caracteriza no sólo la dependencia de las magnitudes (X, Y) , sino también su dispersión. En la práctica, resulta preferible la consideración de una característica adimensional.

Denominada coeficiente de correlación r :

$$r = \frac{C_{xy}}{\sigma_x \sigma_y} \quad (13)$$

Una magnitud bidimensional será correlacionada siempre que $r \neq 0$; en caso contrario, ésta será no correlacionada. Puede afirmarse, además que para un sistema de dos variables aleatorias los conceptos de correlación y dependencia son equivalentes. Esto es, si $r = 0$, (X, Y) son independientes; si $r \neq 0$, (X, Y) son dependientes.

a. Determinación de la zona de dispersión de los centros eventuales de cargas [4, p. 118].

Después que ha sido encontrada la ley de distribución de las coordenadas de los centros eventuales de cargas, y determinadas sus características numéricas, se determina que la zona de dispersión de dichos centros de cargas. Para ello, seccionamos la superficie normal de distribución (fórmula 2, Ilustración 1), según un plano horizontal H , paralelo al xoy .

La proyección de la sección obtenida sobre el plano xoy estará limitada por la curva expresada por la fórmula:

$$H = Q * e^{-\frac{1}{2(1-r^2)} \left[\frac{(x-a_x)^2}{\sigma_x^2} - \frac{2r(x-a_x)(y-a_y)}{\sigma_x \sigma_y} + \frac{(y-a_y)^2}{\sigma_y^2} \right]} \quad (14)$$



$$\ln \frac{H}{Q} = -\frac{1}{2(1-r^2)} \left[\frac{(x-a_x)^2}{\sigma_x^2} - \frac{2r(x-a_x)(y-a_y)}{\sigma_x\sigma_y} + \frac{(y-a_y)^2}{\sigma_y^2} \right] \quad (15)$$

$$Q = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-r^2}} \quad (16)$$

Simbolizando $2(1-r^2) \ln H/Q = \lambda^2$, obtenemos la ecuación de una elipse, denominada elipse $-\lambda$:

$$\frac{(x-a_x)^2}{\sigma_x^2} - \frac{2r(x-a_x)(y-a_y)}{\sigma_x\sigma_y} + \frac{(y-a_y)^2}{\sigma_y^2} = \lambda^2 \quad (17)$$

La ecuación de la elipse $-\lambda$ puede ser analizada por los métodos convencionales de la geometría analítica. Aplicando éstos, llegamos a la conclusión de que, efectivamente, el centro de la elipse se encuentra en el punto con coordenadas (a_x, a_y) ; que los ejes de simetría de la elipse forman ángulos con el eje $0x$, determinados por la expresión

$$\tan 2\alpha = \frac{2r\sigma_x\sigma_y}{\sigma_x^2 - \sigma_y^2} \quad (18)$$

$$\alpha = \frac{1}{2} \tan^{-1} \frac{2C_{xy}}{D_x - D_y} \quad (19)$$

De esta ecuación son obtenidos dos valores de ángulos, desplazados entre sí 90° . De esta manera, la orientación de la elipse, con respecto a los ejes coordenados originales. Se halla en dependencia directa del coeficiente de correlación r del sistema (X, Y) . Si las magnitudes son no correlacionadas (es decir, independientes), los ejes de simetría de la elipse son paralelos a los ejes coordenados; en caso contrario, formarán algún ángulo con respecto a dicho ejes.

Es sabido que la ecuación de la elipse se adopta a su forma más simple, denominada forma canónica, si los ejes coordenados coinciden con los ejes de simetría de la elipse. Para pasar la ecuación de la elipse de dispersión a su forma canónica es suficiente trasladar el origen de coordenadas al punto (a_x, a_y) y rotar los ejes coordenados un ángulo α , determinando por la expresión (19).



Se obtiene, así, un nuevo sistema de ejes coordenados ψ o ϕ , para el que $r = 0$ o, lo que es lo mismo, para el que las nuevas variables o magnitudes aleatorias ψ y ϕ resultan independientes. La forma canónica de la ley normal en el plano viene dada, entonces por:

$$f(\psi, \phi) = \frac{1}{2\pi\sigma_\psi\sigma_\phi} e^{-\frac{1}{2}\left(\frac{\psi^2}{\sigma_\psi^2} + \frac{\phi^2}{\sigma_\phi^2}\right)} \quad (20)$$

Donde:

σ_ψ, σ_ϕ = desviaciones medio-cuadráticas de las coordenadas eventuales ψ, ϕ , en el sistema de coordenadas ψ o ϕ .

Las desviaciones medio-cuadráticas σ_ψ, σ_ϕ se expresan a través de las desviaciones medio-cuadráticas del sistema original de coordenadas, por las fórmulas:

$$\sigma_\psi^2 = \sigma_x^2 \cos^2 \alpha + r\sigma_x\sigma_y \operatorname{sen} 2\alpha + \sigma_y^2 \operatorname{sen}^2 \alpha \quad (21)$$

$$\sigma_\phi^2 = \sigma_x^2 \operatorname{sen}^2 \alpha - r\sigma_x\sigma_y \operatorname{sen} 2\alpha + \sigma_y^2 \cos^2 \alpha \quad (22)$$

La ecuación de la elipse de dispersión en la forma canónica adopta la forma siguiente:

$$h_\psi^2 \psi^2 + h_\phi^2 \phi^2 = \lambda^2 \quad (23)$$

Donde:

$$h_\psi = \frac{1}{\sqrt{2}} \sigma_\psi ; h_\phi = \frac{1}{\sqrt{2}} \sigma_\phi \quad (24)$$

Para la ecuación (13) resulta evidente que los semiejes vienen dados por:

$$R_\psi = \frac{\lambda}{h_\psi} ; R_\phi = \frac{\lambda}{h_\phi} \quad (25)$$



La probabilidad $P("λ")$ de ocurrencia de los centros eventuales de carga, dentro de la elipse, está dada por:

$$P(λ) = \iint_{s(λ)} f(ψ, φ) dψ dφ \quad (26)$$

Donde:

$s(λ)$ = área, limitada por la elipse - $λ$.

Resolviendo, se obtiene:

$$P(λ) = 1 - e^{-λ^2} \quad (27)$$

Es decir, la probabilidad de ocurrencia de los centros eventuales de cargas en la elipse de dispersión es una función de $λ$. Tomando en calidad de probabilidad confiable el valor $P(λ) = 0.95$ y resolviendo la ecuación (15), obtenemos:

$$0.95 = 1 - e^{-λ^2} \quad (28)$$

De donde:

$$λ^2 \approx 3$$

Con ello, se ha encontrado que el 95% de los centros eventuales de carga ocurren dentro de una elipse cuyos semiejes, según las expresiones (14), son iguales a:

$$R_ψ = \frac{\sqrt{3}}{h_ψ} ; R_φ = \frac{\sqrt{3}}{h_φ} \quad (29)$$

Obteniendo de esta manera la longitud para cada uno de los ejes de la elipse una vez que se haya establecido en el mapa la ubicación de su centro (esperanzas matemáticas) y así poder obtener la gráfica para su posterior visualización.



V.3 Sistema Informático

V.3.1 Definición

Un **sistema informático** es un conjunto de elementos que hace posible el tratamiento automático de la información. Las partes de un sistema informático son:

- Componente físico: está formado por todos los aparatos electrónicos y mecánicos que realizan los cálculos y el manejo de la información.
- Componente lógico: se trata de las aplicaciones y los datos con los que trabajan los componentes físicos del sistema.
- Componente humano: está compuesto tanto por los usuarios que trabajan con los equipos como por aquellos que elaboran las aplicaciones.

Un ejemplo de sistema informático sería uno o varios ordenadores personales o PC, junto con la persona que lo maneja, los programas que contiene y los periféricos que los envuelven (impresora, teclado, altavoces, entre otros).

V.3.2 Base de datos

El término base de datos surgió en 1963 según [8], en la informática una base de datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos de datos. En otras palabras, una base de datos no es más que un conjunto de información (un conjunto de datos) relacionada que se encuentra agrupada o estructurada.

Por otra parte, listaremos a continuación definiciones acerca de bases de datos que proveerán una idea más clara al lector.

1.- “Colección de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones” (Conference des Statisticiens Européens, 1977).



2.- “Es un conjunto exhaustivo de datos estructurados, fiables y homogéneos, organizados independientemente de su utilización y de su implementación en máquina, accesibles en tiempo real, compartibles por usuarios concurrentes que tienen necesidades de información diferentes y no predecibles en el tiempo” (Access, 2001).

V.3.3 Modelo Relacional

De acuerdo con [8], “en el modelo relacional se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas, y cada columna tiene un nombre único. El modelo relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se llaman así porque la base de datos se estructura en registros de formato fijo de varios tipos.”

Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro. El modelo relacional oculta detalles de implementación de bajo nivel para con los desarrolladores de bases de datos y a los usuarios.

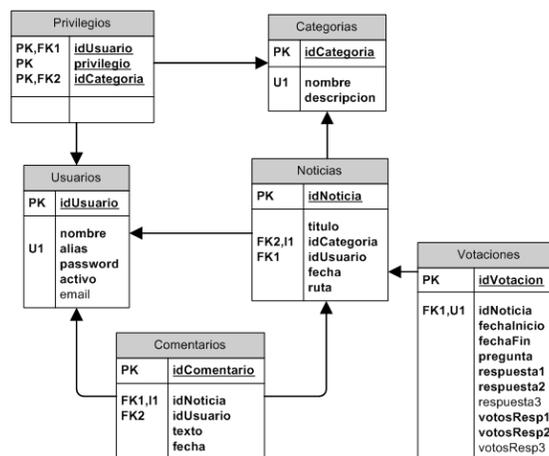


Ilustración 3 - Ejemplo de esquema del modelo relacional.

En las bases de datos relacionales los datos se almacenan en distintas tablas por asunto o tarea, pero están relacionados y se pueden combinar de las maneras que se especifique, de tal forma que se puede extraer y unir toda esta información siempre que se desee.



V.3.4 Aplicación Web

En la ingeniería software se denomina aplicación web según [9] a “aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, y en la que se confía la ejecución de la aplicación al navegador. Las aplicaciones web para Internet e Intranet presentan una serie de ventajas y beneficios con respecto al software de escritorio, con lo cual logrará aprovechar y acoplar los recursos de su empresa de una forma mucho más práctica que el software tradicional.”

Entre los beneficios que las aplicaciones desarrolladas para la web tienen respecto a las aplicaciones de escritorio se encuentran:

- El trabajo a distancia se realiza con mayor facilidad -Para trabajar en la aplicación web solo se necesita un computador con un buen navegador Web y conexión a internet.
- Las aplicaciones Web no necesitan conocimientos previos de informática. Con una aplicación Web tendrá total disponibilidad en cuanto a hora y lugar, podrá trabajar en ella en cualquier momento y en cualquier lugar del mundo siempre que tenga conexión a internet.
- Las aplicaciones Web le permiten centralizar todas las áreas de trabajo.

V.3.5 Modelo Cliente Servidor

El termino Cliente/Servidor en [10] se refiere a un modelo de aplicaciones de red donde los procesos juegan uno de los dos diferentes papeles: Un proceso servidor, también llamado servidor para abreviar se dedica a gestionar el acceso a algunos servicios de la red, mientras que los procesos clientes acceden al servidor para obtener un servicio de red.



Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor en base a lo expuesto en [11] como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. “En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta. En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras.”

La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PC's a través de interfaces gráficas de usuario.

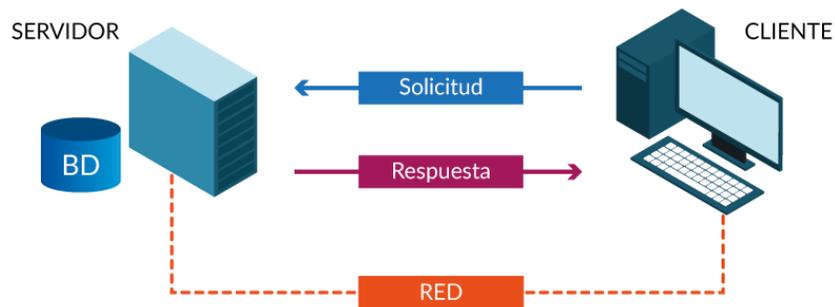


Ilustración 4 - Puesta en escena del modelo cliente/servidor.

Mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario (aunque esto puede variar).

En otras palabras, la arquitectura **Cliente/Servidor** es una extensión de programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento.



V.4 Metodología del desarrollo de Software

La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito.

Esta sistematización nos indica como dividiremos un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas, nos ayudan a definir entradas y salidas para cada una de las etapas y, sobre todo, normaliza el modo en el que se administrara el proyecto.

Una metodología para el desarrollo de software se compone de los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo del cual fue creado.
[12]

V.4.1 Ciclo de vida

Las principales diferencias entre distintos modelos de ciclos de vida en tres grandes visiones:

- El alcance del ciclo de vida, que depende de hasta donde deseamos llegar con el proyecto: solo saber si es viable el desarrollo de un producto, el desarrollo completo o el desarrollo completo más las actualizaciones y el mantenimiento.
- La caducidad y cantidad de las etapas en que dividiremos el ciclo de vida: según el ciclo de vida que adoptemos, y el proyecto para el cual lo adoptemos.
- La estructura y la sucesión de las etapas, si hay retroalimentación entre ellas y si tenemos libertad de repetirlas.

- **Ciclo de vida lineal**

Es el más sencillo de todos los modelos. Consiste en descomponer la actividad global del proyecto en etapas separadas que son realizadas de manera lineal, es decir, cada etapa se realiza una sola vez, a continuación de la etapa anterior y antes de la etapa siguiente. Con un ciclo de vida lineal es muy fácil de dividir las tareas, y prever los tiempos (sumando linealmente los de cada etapa). [12]

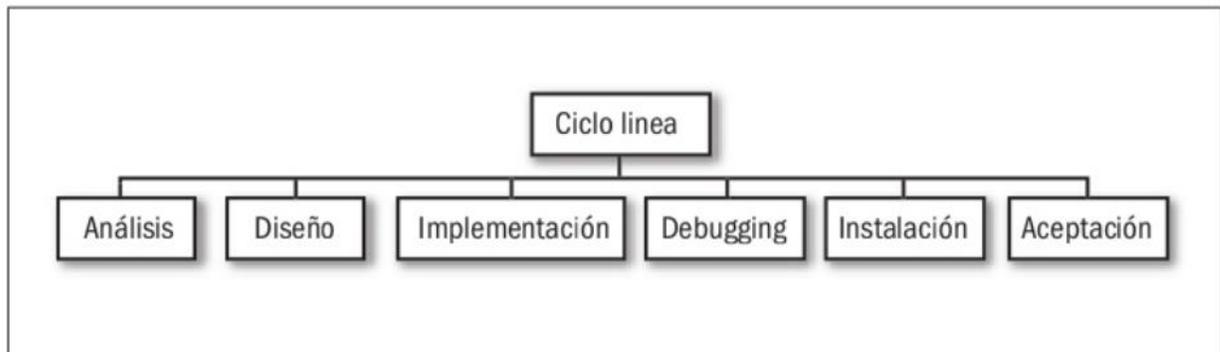


Ilustración 5 - Ciclo de vida Lineal.

- **Ciclo de vida en V**

Este ciclo fue diseñado por Alan Davis, y contiene las mismas etapas que el ciclo de vida en cascada puro. A diferencia de aquél, a éste se le agregaron dos subetapas de retroalimentación entre las etapas de análisis y mantenimiento, y entre las de diseño y depuración. [12]

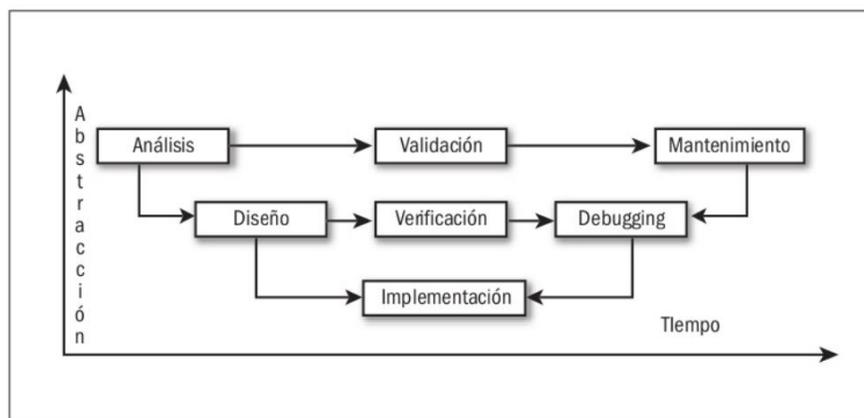


Ilustración 6 - Ciclo de vida en V.

- **Ciclo de vida en Espiral**

Es un modelo de ciclo de vida del software definido por primera vez por Barry Boehm en 1986. Las actividades de este modelo se conforman en una espiral. Cada bucle representa un conjunto de actividades.



En cada vuelta o iteración hay que tener en cuenta:

- Los objetivos: qué necesidad debe cubrir el producto.
- Alternativas: las diferentes formas de conseguir los objetivos de forma exitosa, desde diferentes puntos de vista como pueden ser:
 - Características.
 - Formas de gestión del sistema.
 - Riesgo asumido con cada alternativa.
- Desarrollar y Verificar: Programar y probar el software.

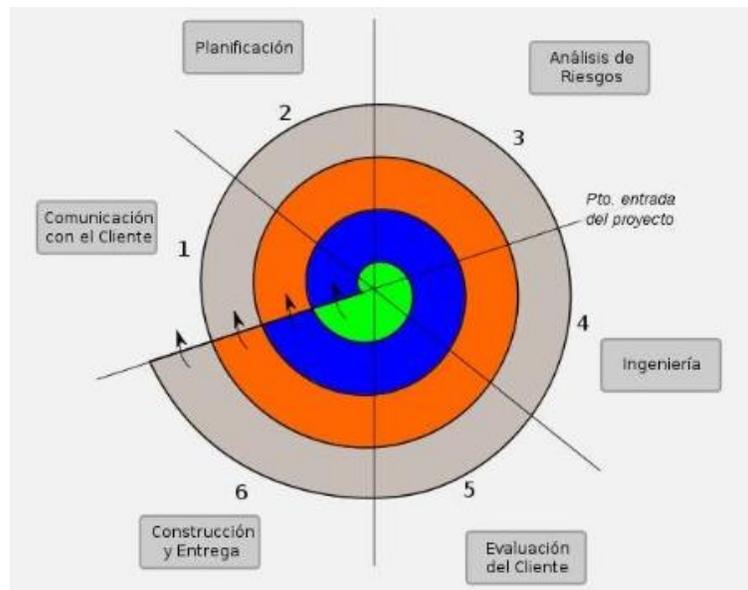


Ilustración 7 - Ciclo de vida en espiral.

V.4.2 UML

Según [13] “el lenguaje UML es un estándar OMG⁸ diseñado para visualizar, especificar, construir y documentar software orientado a objetos.”

Fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas.

⁸ OMG: Object Management Group.



En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene.

V.4.3 Metodología UWE

UWE es un:

... método de ingeniería del software para el desarrollo de aplicaciones web basado en UML. ¡Cualquier tipo de diagrama UML puede ser usado, porque UWE es una extensión de UML!, El enfoque UWE proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y soporte de herramientas para la ingeniería de aplicaciones Web. [14]



Ilustración 8 - Etapas de la metodología UWE.

UWE utiliza notación UML "pura" y tipos de diagramas UML siempre que es posible para el análisis y diseño de aplicaciones Web, es decir, sin extensiones de ningún tipo. Para las características específicas de la Web, como nodos y enlaces de la estructura de hipertexto, el perfil UWE incluye estereotipos, valores etiquetados y restricciones definidas para los elementos de modelado. La extensión UWE cubre la navegación, presentación, procesos de negocio y aspectos de adaptación. La notación UWE se define como una extensión "ligera" del UML.

Se ha considerado realizar la aplicación como un sistema web y hacer uso de la metodología UWE para su desarrollo pues es una herramienta para modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos).



Ciertamente existe una amplia gama de metodologías web, pero hemos decidido inclinarnos e implementar en nuestra tesis la metodología UWE, ya que esta es una propuesta basada en el proceso unificado y UML, pero adaptados a la web.

Una de las razones por las que utilizaremos esta metodología es que se estructura en requisitos que separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

Consiste en una notación y en un método, lo cual será muy útil en el momento del diseño del sistema ya que nos servirá de guía para el desarrollo de este.

El método consta de cuatro pasos, listados a continuación:

- 1) Modelo de casos de uso.
- 2) Modelo conceptual.
- 3) Modelo de espacio de navegación y modelo de estructura de navegación.
- 4) Modelo de presentación

Cada uno de estos pasos proveerá pautas que se tomarán en consideración para el diseño del sistema para que al momento de la creación del proyecto este sea eficiente y cumpla con el objetivo para el cual fue diseñado.

Modelo de casos de uso.

El objetivo del análisis de requisitos es encontrar los requisitos funcionales del sistema Web y representar estos requisitos como casos de uso. [15]

Modelo conceptual.

Para la creación de un modelo conceptual es indispensable tener en cuenta los requisitos capturados con los casos de uso. Las técnicas tradicionales orientadas a objetos se utilizan para construir el modelo conceptual, como encontrar clases y asociaciones y definir estructuras de herencia.

El modelo conceptual está representado por un diagrama de clases UML ordinario. [15]



Modelo de espacio y estructura de navegación.

Tomando como punto de partida el modelo conceptual, el método de navegación propone un conjunto de pautas para construir un modelo de navegación que represente el espacio y la estructura de navegación agregando elementos que se pueden usar para navegar en el sistema.

El método incluye un conjunto de elementos de modelado estereotipados UML para el diseño de la navegación, como índices, visitas guiadas, consultas y menús. [15]

Modelo de presentación

El modelado de presentación tiene como objetivo el diseño de interfaces de usuario abstractas y el diseño de la interacción del usuario con la aplicación web. [15] Este consta de dos pasos:

1. En el diseño de la presentación define las vistas de la interfaz de usuario que esbozan el contenido y la apariencia de los nodos. Estas vistas de la interfaz de usuario se pueden combinar con escenarios de guiones gráficos.
2. El segundo paso se centra en la dinámica de la presentación representada con diagramas de secuencia UML.

V.4.4 Arquitectura MVC

En [16] se define:

MVC (Modelo-Vista-Controlador) es un patrón para crear aplicaciones web que es usado en casi todos los marcos de desarrollo web (ejemplos populares son Ruby on Rails y Express), adicionalmente marcos de trabajo del lado de cliente con JavaScript como Angular. Las aplicaciones móviles sobre iOS y Android también usan una variante de MVC.

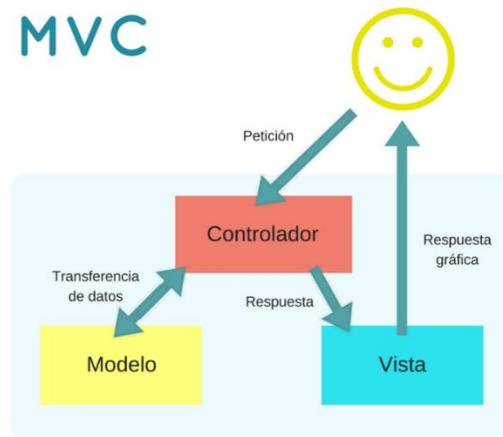


Ilustración 9 - Arquitectura MVC

V.5 Herramientas de desarrollo

V.5.1 HTML5

HTML5 es considerado el producto de la combinación de HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5. HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y JavaScript hace el resto que (como veremos más adelante) es extremadamente significativo.

Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones.

Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como los fundamentos de un edificio.

Para trabajar y crear sitios webs y aplicaciones con HTML5, necesitamos saber primero cómo esa estructura es construida. En [17] “Crear fundamentos fuertes nos ayudará más adelante a aplicar el resto de los componentes para aprovechar completamente estas nuevas tecnologías.”



V.5.2 CSS

CSS “Cascading Style Sheets” Es un intento por reducir el uso de código JavaScript y para estandarizar funciones populares, CSS3 no solo cubre diseño y estilos web sino también forma y movimiento. La especificación de CSS3 es presentada en módulos que permiten a la tecnología proveer una especificación estándar por cada aspecto involucrado en la presentación visual del documento.

Desde esquinas redondeadas y sombras hasta transformaciones y reposicionamiento de los elementos ya presentados en pantalla, cada posible efecto aplicado previamente utilizando JavaScript fue cubierto. En [17] explica que este nivel de cambio convierte CSS3 en una tecnología prácticamente inédita comparada con versiones anteriores.

V.5.3 JavaScript

JavaScript es definido por [17] como:

... un lenguaje interpretado usado para múltiples propósitos, pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio.

Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje JavaScript como la mejor opción para la web. Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandido en relación con portabilidad e integración.

A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores.



“La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, facilitando la creación de programas útiles para la web.” [17]

V.5.4 JQuery

Las aplicaciones web se sitúan en un entorno dinámico, actualmente ricos en términos de presentación visual y otras funciones, basadas en gestión de la información. Siendo framework libre y open source del lado del cliente, que se centra en la interacción entre el DOM, JavaScript, AJAX y HTML; teniendo como objetivo aplicar los comandos comunes siendo el lema de JQuery "Escribir menos para hacer más". [18].

JQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. En [19] menciona algunas de sus características, como el desplazamiento y la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript.

V.5.5 Bootstrap

En cuanto a [20] Bootstrap es un kit de herramientas de código abierto para desarrollar con HTML, CSS y JS, realizando rápidamente prototipos, creando aplicaciones con variables y mixins Sass, sistema de cuadrícula sensible, componentes precompilados extensos y complementos potentes creados en jQuery.

De igual manera, en [21] asegura que Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios a pie, en coche, bicicleta (beta) y transporte público y un navegador GPS, Google Maps Go.

V.5.6 Fabric JS

Fabric.js , una potente biblioteca de JavaScript que facilita el trabajo con HTML5 Canvas .



Fabric proporciona un modelo de objeto faltante para el lienzo, así como un analizador SVG, una capa de interactividad y un conjunto completo de otras herramientas indispensables. Es un proyecto de código abierto [22], con licencia del MIT, con muchas contribuciones a lo largo de los años.

De acuerdo con [22] Fabric comenzó alrededor de 2010, después de descubrir los inconvenientes de trabajar con la API de lienzo nativa. El autor original estaba creando un editor de diseño interactivo para printio.ru ,su startup que permite a los usuarios diseñar su propia indumentaria. El tipo de interactividad que querían solo existía en las aplicaciones Flash en esos días. Incluso ahora, muy pocos se acercan a lo que se hizo posible con Fabric.



VI. DISEÑO METODOLÓGICO

El presente capítulo detalla un estudio de prefactibilidad y las fases de desarrollo cómo lo son: captura, análisis y especificación de requerimientos, diseño de sistema, codificación del software y pruebas. Cabe recalcar que el sistema GLOCSE, constará de cinco módulos, los cuales son: Mapa, Gráficas, Reportes, Proyecciones y Configuración.

Para el desarrollo del software se decidió hacer uso de la metodología en espiral, para la constante revisión de las diferentes fases del sistema y su corrección temprana.

A continuación, se muestra la arquitectura general del sistema.

DISEÑO ARQUITECTÓNICO

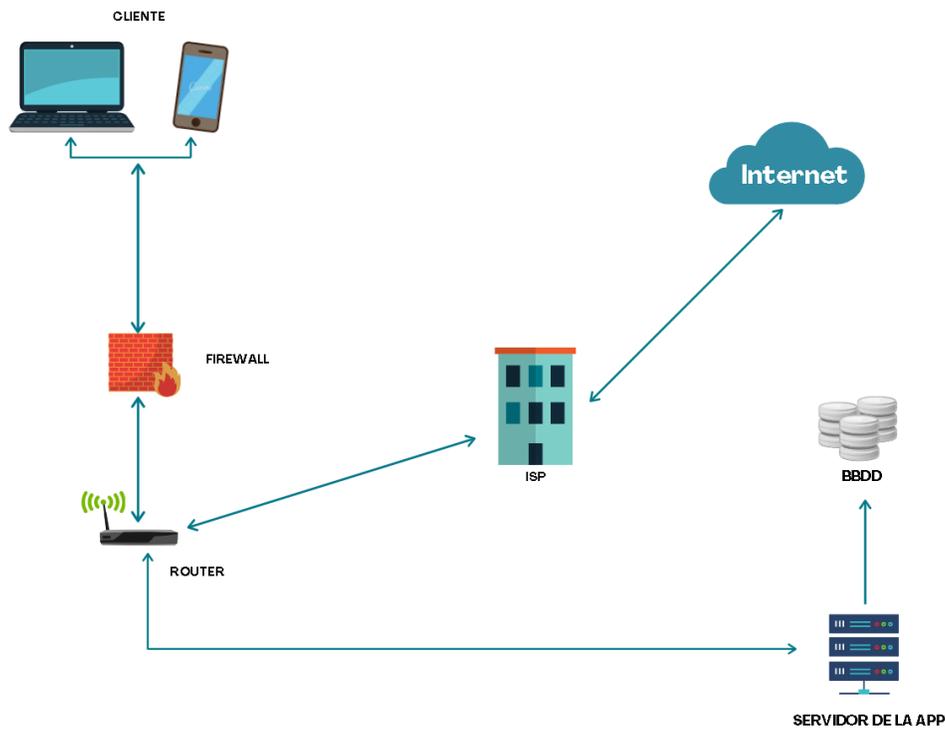


Ilustración 10 - Estructura del sistema



VI.1 Estudio de Prefactibilidad

Este estudio de factibilidad se basa en el análisis de los requerimientos necesarios para llevar a ejecución este proyecto (prototipo de sistema web para determinar y visualizar la geolocalización de subestaciones eléctricas en Nicaragua) a fin de determinar la viabilidad técnica, operativa y económica que conllevará la realización de este. De manera que sirva este estudio para proveer la información necesaria para la toma de decisión de realizar este proyecto o no.

VI.1.1 Factibilidad Técnica

Para desarrollo del sistema web GLOCSE es necesario contar con ciertos recursos de hardware y de software. (Ver Tabla 1 - Aspectos técnicos de hardware. y Tabla 2 - Aspectos técnicos de software.)

Se presentan aquellos elementos que serán fundamentales para este prototipo del sistema a desarrollar, para que este opere óptimamente.

Hardware

- Computador laptop
 - Uso: Los computadores serán necesarios para el desarrollo del software, realización de pruebas, análisis de reportes, visualización de gráficas de consumidores y trazo de la elipse en el mapa del sistema.
 - Características: Procesador Intel(R) Core (TM) i7-5600U, 1 TB HDD, 8 GB RAM.
- Internet
 - Uso: Navegación en la web para consumo de API de Google Maps y pruebas correspondientes.
 - Características: El internet a utilizar es de los más económicos, estables y de buena calidad que asegura un funcionamiento óptimo del sistema.



Software

- Sistema operativo Windows 10
 - Uso: Es en donde se ejecutarán los programas a utilizar para el desarrollo de GLOCSE.
 - Características: Es el intermediario entre el usuario y el hardware. Necesario para el funcionamiento de todos los computadores, tabletas y teléfonos móviles. Otorga seguridad y protege a los programas y archivos del ordenador. Está diseñado para ser amigable con el usuario y fácil de usar. Permite interactuar con varios dispositivos.

- GitHub
 - Uso: Sirve como un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores.
 - Características: Provee una wiki y una página web para cada proyecto. Ofrece gráficas para ver cómo los desarrolladores trabajan en sus repositorios y bifurcaciones del proyecto.

- Visual Studio 2019 Edition Community.
 - Uso: Tiene como objetivo crear aplicaciones modernas para Windows, Android e iOS, además de aplicaciones web y servicios en la nube.

 - Características: Es flexible, aumenta la productividad y posee gran gama de lenguajes de programación. Ofrece una instalación ligera y modular, conexión a la nube, herramientas de codificación y aplicaciones para dispositivos.



- SQL Server 2019 Developer.
 - Uso: Desarrollo de base de datos y pruebas necesarias.
 - Características: Escalabilidad y rendimiento innovadores para mejorar la estabilidad y el tiempo de respuesta de la base de datos. Alta disponibilidad de aplicaciones, almacenes de datos y lagos de datos críticos, ahora con soporte incorporado para macrodatos.
- Azure SQL Database
 - Uso: Se encarga de la mayoría de las funciones de administración de bases de datos, como actualizar, aplicar revisiones, crear copias de seguridad y supervisar sin intervención del usuario.
 - Características: Encriptación, grupos de disponibilidad, auditorías, certificados y claves asimétricas, transacción entre base de datos.
- Visual Paradigm
 - Uso: Propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.
 - Características: Permite hacer clic y arrastrar para construir formas y líneas en un flujograma con una guía de alineación para ajustar todo en su lugar. Cuenta con videos tutoriales para los recién iniciados. Integración con **Eclipse**, **VisualStudio** y **Net Beans**. Puede generar y compartir diagramas a través de formatos como PNG, JPG, GIF, SVG y PDF.



- App Service
 - Uso: Creación de APIs y aplicaciones web en la nube.
 - Características: Plataforma totalmente administrada que cuenta con funcionalidad integrada de mantenimiento de la infraestructura, aplicación de revisiones de seguridad y escalado. Integración con redes virtuales y capacidad para ejecutarse en un entorno de App Service aislado y dedicado. Rigor en cuanto a la seguridad y el cumplimiento normativo.

HARDWARE		
Cantidad	Recurso	Descripción
2	Computadora laptop.	Procesador Intel(R) Core (TM) i7-5600U, 1 TB HDD, 8 GB RAM.
1	Internet	Servicio de internet, 5MB

Tabla 1 - Aspectos técnicos de hardware.

SOFTWARE		
Cantidad	Recurso	Descripción
2	Sistema operativo	Windows 10 Home.
2	Software	Visual Studio 2019 Edition Community.
2	GitHub	Usuarios de GitHub.
2	SQL Server	SQL Server 2019 Developer.
1	Azure SQL Database	Base de datos única, modelo de compra DTU, nivel Basic Tier, B: 5 DTU, 2 GB de almacenamiento incluido por base de datos, 1 bases de datos x 1 Mes, 1, retención GB.
1	Visual Paradigm	Visual Paradigm proporciona modelado de notación más formal y capacidad de dibujo casual. Los diagramas en categorías de modelado como UML.
1	App Service	Nivel Gratis; 1 F1 (0 núcleos, 1 GB de RAM, 1 GB de almacenamiento) x 730 Horas; Sistema operativo Windows.

Tabla 2 - Aspectos técnicos de software.



VI.1.2 Factibilidad Económica

En este apartado, se mencionan los gastos que involucra el proyecto como tal, estos gastos serán de costos de hardware, de software, de desarrollo y costos totales.

Se ha realizado las investigaciones correspondientes para determinar cuáles serán los costos más factibles y viables para hacer posible el desarrollo del prototipo web GLOCSE.

Se provee una lista detallada de estos gastos. (Ver Tabla 3 - Costos de hardware., Tabla 4 - Costos de software., Tabla 5 - Costos de desarrollo. y Tabla 6 - Costos totales.)

COSTOS DE HARDWARE			
Cantidad	Descripción	Precio Unitario	Total
2	Procesador Intel(R) Core (TM) i7-5600U, 1 TB HDD, 8 GB RAM.	\$600.00	\$1,200.00
6	Servicio de internet, 5 MB por mes.	\$20.00	\$120.00
Total			\$1,320.00
<i>Nota: Computadores e internet no fueron parte de costos totales, ya que se contaban con dichos recursos.</i>			

Tabla 3 - Costos de hardware.

COSTOS DE SOFTWARE			
Cantidad	Descripción	Precio Unitario	Total
2	Windows 10 Home	\$199.00	\$398.00
2	Visual Studio 2019 Edition Community.	\$0.00	\$0.00
2	Usuarios de GitHub	\$0.00	\$0.00
2	SQL Server 2019 Developer	\$0.00	\$0.00
6	Azure SQL Database	\$4.90	\$29.38
6	Visual Paradigm	\$6.00	\$36.00
1	App Service	\$0.00	\$0.00
Total			\$463.38
<i>Nota: El uso de Azure SQL Database y Visual Paradigm está contemplado a 6 meses (Precio unitario mensual). Licencias de Windows 10 Home no fueron necesarias, ya que los computadores poseían licencias OEM (de fábrica).</i>			

Tabla 4 - Costos de software.



COSTOS DE DESARROLLO				
Cantidad	Descripción	Precio Unitario	Meses	Total
2	Desarrollador web junior	\$700.00	\$7.00	\$9,800.00
Total				\$9,800.00

Tabla 5 - Costos de desarrollo.

COSTOS TOTALES		
Descripción	Total (con recursos existentes)	Total (adquiriendo todos los recursos)
Costos de Hardware	\$0.00	\$1,320.00
Costos de Software	\$29.38	\$463.38
Costos de Desarrollo	\$0.00	\$9,800.00
Total	\$29.38	\$11,583.38

Tabla 6 - Costos totales.

VI.1.3 Factibilidad Operativa

La factibilidad operativa permite conocer lo urgente de implementar un proceso y la posible aceptación de este por parte de los usuarios que utilizarán el sistema.

Los aspectos para tomar en cuenta son que por lo general el uso de un sistema puede ser intimidante para personas que no estén familiarizadas con el uso de programas de computadora, es por eso por lo que se tomará en cuenta una interfaz minimalista, sencilla y fácil de usar, de tal manera que la navegación dentro de este no sea complicada. Esto será facilitado mediante la diagramación UML.

La implementación de este sistema permitirá a usuarios del sistema GLOCSE realizar cálculos tediosos y complejos automáticamente, anteriormente se podría encontrar el error humano y este es un factor muy común y grave al momento de realizar los cálculos de forma manual. Aún más dado la estimación de costos al momento de la construcción de las SE.



Dado a lo anterior dicho, nosotros denotamos que el sistema es viable a nivel operativa debido a que con la implementación de este se solucionarían muchos problemas, como el manejo de cargas eléctricas por consumidores, ubicación de los consumidores, proyecciones en periodos de tiempos predefinidos por el usuario, lo que en su defecto incluye a un uso drástico de tiempo para realizar las operaciones matemáticas pertinentes.

VI.1.4 Factibilidad Legal

En esta sección definiremos los parámetros legales en que este prototipo de proyecto se basará de manera que podamos dejar en claro los requerimientos legales para su operación a futuro, licencias para un sistema informático de manera auténtica, con la finalidad de no tener inconvenientes legales a futuro.

VI.1.4.1 Requerimientos legales

Según la Ley de Derechos de Autor y Conexos, el proyecto final (segunda fase del prototipo) deberá inscribirse en el registro según el artículo 55 Derecho por Inscripción y Servicios, con un costo de \$CA100.00

VI.1.4.2 Las licencias para el software

De acuerdo con el software que utilizaremos para el funcionamiento del sistema:

- Azure SQL Database: Este será necesario para el alojamiento de la base de datos.
- App Service: Cuenta con funcionalidad integrada de mantenimiento de la infraestructura, aplicación de revisiones de seguridad y escalado para que la aplicación opere en la nube.

VI.1.4.3 Propiedad Intelectual

Los desarrolladores se reservan el derecho de autor del código fuente, siendo ellos únicos creadores del sistema, basado en el art. 132 inciso b) de la Ley No. 380, Ley de Marcas y Otros Signos Distintivos.



Conforme los artículos 22 y 23 de la Ley Derechos de Autor y Conexos, los desarrolladores tienen el derecho exclusivo de realizar o autorizar las traducciones, así como las adaptaciones, arreglos y otras modificaciones de su obra, teniendo potestad para realizar ajustes necesarios al sistema.

Sin embargo, debido a la naturaleza de este documento, por ser un trabajo de carácter monográfico, según la Normativa para los trabajos monográficos, en el Título V, capítulo único, en su art. 48, “todos los trabajos monográficos se constituyen parte integrante del patrimonio de la Universidad y de las Facultades.” [23] Por lo cual se ceden los derechos de autor a la Universidad Nacional de Ingeniería.

VI.2 Captura, análisis y especificación de requerimientos

VI.2.1 Actores

Los actores que entran en juego en este caso serán los usuarios que inician sesión en la plataforma e interactúan con los espacios de trabajos que contienen las proyecciones de cada proyecto.

Estos actores en su mayoría serán ingenieros eléctricos o en un perfil académico y/o profesional de conocimiento similar al necesario para la comprensión de los procesos que realiza el sistema GLOCSE.

VI.2.2 Requerimientos Funcionales

Los requerimientos funcionales del sistema van de acuerdo con las necesidades expresadas por especialista en el tema, al igual que los requerimientos no funcionales. Se determinarán los elementos claves necesarios para el cálculo de la ubicación de la SE y se adaptarán en el prototipo de sistema forma óptima y sistematizada.



El prototipo del sistema web deberá realizar:

- Recibimiento de credenciales de usuarios y su verificación para el proceso de Inicio de Sesión.
- Contar con diferentes módulos según el área a cubrir. (Mapa, Gráficas, Reportes, Proyecciones y Configuración)
- Permitir la creación de nuevos espacios de trabajos.
- Almacenamiento de datos en la Base de datos referente a los módulos definidos previamente.
- Visualización de datos almacenados en la base de datos.
- Permitir modificaciones en los datos de las proyecciones.
- Cargar los espacios de trabajos guardados.
- Mostrar las gráficas de consumidores que alimentan las proyecciones.
- Visualizar el valor de las variables que construyen la elipse y cada proyección de esta misma.
- Generación de reportes con datos globales del sistema por espacio de trabajo en uso.

VI.2.3 Requerimientos No Funcionales

VI.2.3.1 Reglas del negocio

- Se deberá acceder a la aplicación desde cualquier navegador y cualquier dispositivo, esto debido a su diseño responsivo.
- El sistema deberá permitir inicios de sesión simultáneos para diferentes usuarios.

VI.2.3.2 Requerimientos de seguridad.

- Se tendrán tres intentos de inicio de sesión incorrectos, al cuarto intento el usuario se bloqueará y deberá ser desbloqueado por un administrador.
- Todos los usuarios tendrán acceso a todos los módulos.
- En el caso de inactividad del usuario después de una hora, se procederá a cerrar la sesión de este por motivos de seguridad.



VI.2.3.3 Del Producto

Usabilidad:

RN-1 El diseño de la aplicación deberá ser amigable y sencillo de usar para el usuario.

RN-2 La aplicación deberá tener un diseño adaptable, es decir deberá soportar múltiples resoluciones.

RN-3 La aplicación deberá ser clara y concisa cuando el usuario se equivoque advirtiéndolo a este último mediante mensajes de advertencia o de error.

RN-4 La aplicación deberá ser amigable para el usuario, de tal forma que la navegación sea fluida y rápida.

RN-5 Realizar máscaras de validación a los archivos .csv que lo ameriten, para evitar el ingreso de valores inadecuados, según el contexto de la funcionalidad al momento de cargar los datos.

Confiabilidad:

RN-6 El sistema deberá poseer un sistema de autenticación de usuarios, de tal forma que sólo usuarios registrados puedan acceder a los módulos.

RN-7 Facilidades y controles para permitir el acceso a la información a los usuarios autorizado a través de Internet, con la intención de consultar y subir información.



Desempeño:

RN-8 Los tiempos de respuesta de las vistas, deberán ser de entre 500MS a 1 Segundos, para que la navegación por el sistema no se vea afectada.

RN-9 El rendimiento del sistema no se debería ver afectado si hay múltiples usuarios usando el sistema.

Soportabilidad:

RN-10 El sistema deberá ejecutarse óptimamente, en una gran gama de hardware con distintas especificaciones técnicas.

Documentación:

RN-11 Todas las vistas, wiki y manual de usuario deben tener los signos de acentuación y ortográficos pertinentes.

RN-12 Utilizar patrones de diseño homogéneos en los textos, botones, cajas de texto y otros controles gráficos.

VI.2.3.4 Del Ambiente

Ético:

RN-13 Los cálculos del sistema para el trazo de ubicación de la elipse deberán ser exactos y estar definidos por los algoritmos previamente definidos. Es decir, no se debe alterar ningún valor.

VI.2.4 Requerimientos de Interfaz

VI.2.4.1 Interfaces de Hardware

- Computador portátil: Mediante este computador el usuario podrá abrir un navegador, ejecutar la aplicación y hacer uso de la misma.



- Modem con internet: Para acceso al internet, de esta manera podrá acceder a recursos del Mapa, tales como consumidores cercanos en la zona dónde se ubicará la subestación eléctrica.

VI.2.4.2 Interfaces de Software

- API de Google Maps: Esta API aportará significativamente para la ubicación de las SE gráficamente, por las bondades que ofrece en sus Mapas para ubicar pines de referencia y gráficas 2D. Mediante el consumo de esta API, podremos obtener información acerca de Mapas, métodos de geolocalización y obtención de nombres de lugares que podrían ser consumidores potenciales para la SE.

VI.2.4.3 Interfaces de Comunicación

La interfaz de comunicación está dada entre el servidor de base de datos y la API por ADO.net que sirve de interfaz entre ambos, a su vez la aplicación WEB estará conectada a la API de Google Maps mediante la red, la comunicación entre ambos está dada mediante peticiones HTTP del cliente al servidor.

VI.2.5 Restricciones de Diseño

El sistema debe estar estructurado bajo la arquitectura modelo, vista y controlador siendo este codificado en C#, esto permitirá que el proyecto sea escalable de cara al futuro permitiendo que crezca sin tener problemas de implementación a largo plazo.



VI.2.6 Requerimientos de Licencia

Se requiere obtener las licencias de software para las diferentes herramientas a utilizar el proceso de análisis, diseño, desarrollo y pruebas del prototipo de sistema web GLOCSE.

Se utilizará una licencia gratuita de SQL Developer Edition debido al alto costo de las ediciones superiores.

- Visual Paradigm
- Visual Studio
- Microsoft Office Home and Student (PowerPoint, Excel, Word y Outlook)
- Base de datos SQL Server.

VI.3 Diseño del Sistema

VI.3.1 Modelo de casos de uso

VI.3.1.1 Diagrama de casos de uso

Paul Kimmel expresa en [24] que los diagramas de casos de uso son responsables principalmente de documentar los macrorrequisitos del sistema. Y se puede pensar en los diagramas de casos de uso como la lista de las capacidades que debe proporcionar el sistema. A continuación, se muestran y describen los casos de uso para el prototipo de sistema web GLOCSE.



VI.3.1.1.1 Diagrama de caso de Uso - Graficar Elipse

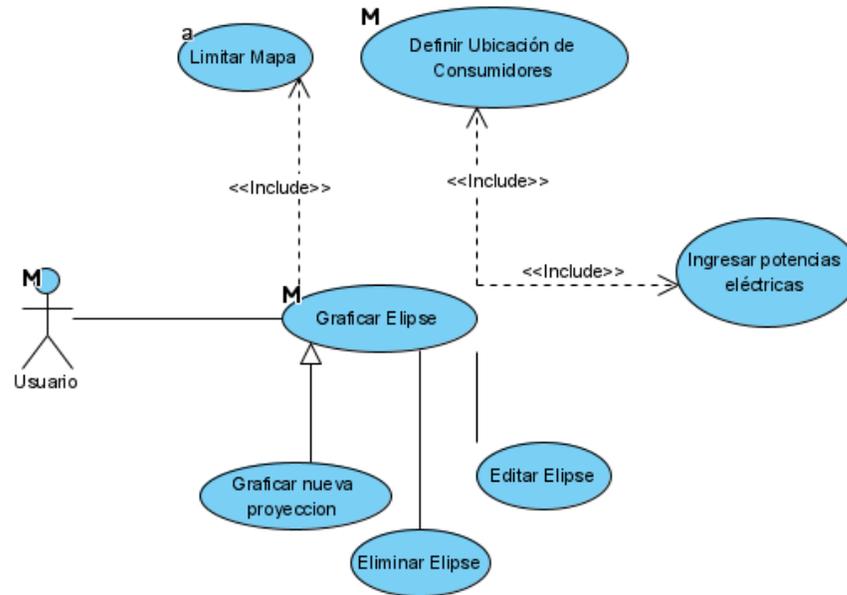


Ilustración 11 - DCU - Graficar Elipse

Título	Graficar Elipse
Inicios del caso de uso	Este caso de uso inicia cuando se ha limitado el mapa, se ubican consumidores y cargan las potencias eléctricas.
Funciones primarias	Mostrar visualmente la elipse en el mapa, actualización de la elipse en caso de cambio de posición de los consumidores. Eliminar elipse.
Necesidades no funcionales	Se debe mostrar la gráfica de la elipse en un periodo no mayor a 10 segundos.
Finalizaciones de los casos de uso	El caso de uso finaliza una vez que se muestran los centros eventuales de carga y la elipse en el mapa gráficamente.

Tabla 7 - Descripción de CU - Graficar Elipse



VI.3.1.1.2 Diagrama de caso de Uso - Espacio de trabajo

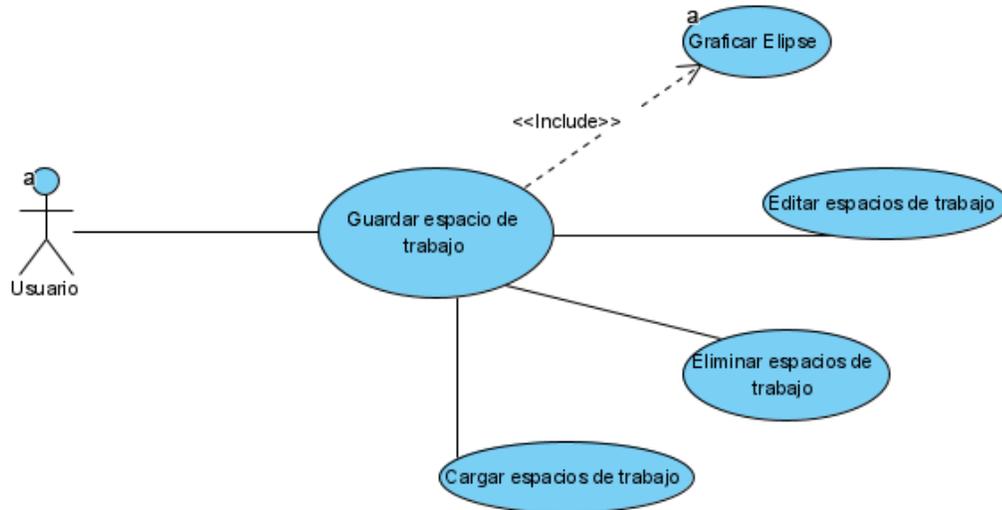


Ilustración 12 - DCU - Guardar espacio de trabajo

Título	Guardar espacio de trabajo
Inicios del caso de uso	Este caso de uso inicia cuando el usuario desea guardar los cambios realizados en las elipses realizadas.
Funciones primarias	Almacenar todos los cambios realizados en las distintas proyecciones en un proyecto de trabajo para ser cargados posteriormente. Estos espacios guardados pueden ser restaurados, eliminados y editados.
Necesidades no funcionales	Se estima que el tiempo de guardado no debe exceder los 5 segundos. Se puede visualizar los espacios de trabajos guardados en la sección Proyecciones.
Finalizaciones de los casos de uso	Una vez guardado el espacio de trabajo, el usuario puede continuar con el uso del sistema normalmente.

Tabla 8 - Descripción de CU - Guardar espacio de trabajo



VI.3.1.1.3 Diagrama de caso de Uso - Cargar datos masivamente

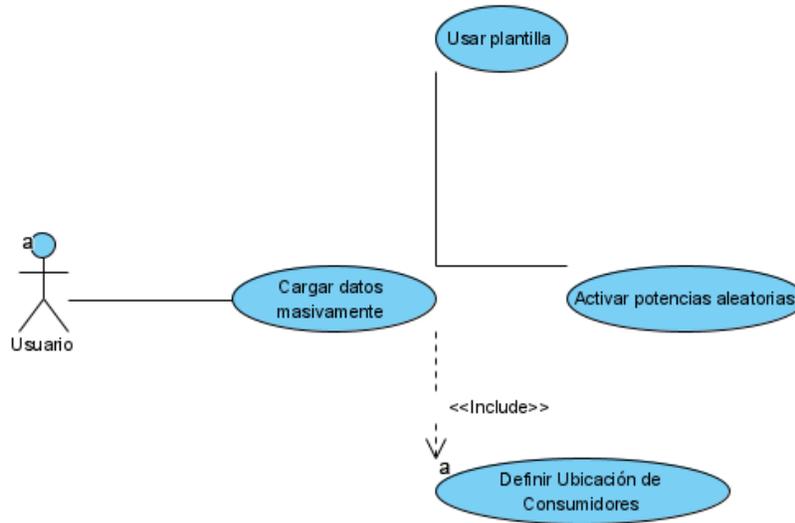


Ilustración 13 - DCU - Cargar datos masivamente

Título	Cargar datos masivamente
Inicios del caso de uso	Para iniciar este caso de uso, es necesario tener limitado una zona en el mapa, y todos los consumidores ubicados en el mapa. Y el usuario asigna cargas eléctricas a cada usuario.
Funciones primarias	Se carga al sistema un archivo de Excel con las cargas eléctricas a cada consumidor ubicado en el mapa para graficar la elipse en una proyección. Se activar potencias aleatorias para generación de cargas eléctricas.
Necesidades no funcionales	La carga de datos masiva no deberá demorarse más de 2 segundos. Archivo de Excel a subir debe usar formato proveído.
Finalizaciones de los casos de uso	Finaliza una vez que los usuarios ya posean cargas eléctricas asignadas.

Tabla 9 - Descripción de CU - Cargar datos masivamente



VI.3.1.1.4 Diagrama de caso de Uso - Generar superficie normal

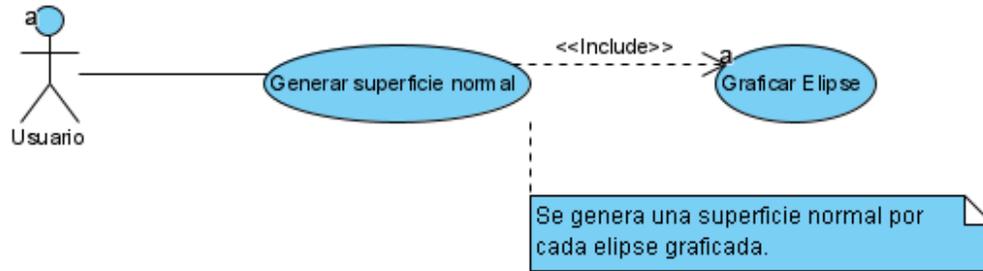


Ilustración 14 - DCU - Generar superficie normal

Título	Generar superficie normal
Inicios del caso de uso	Este caso de uso comienza cuando el usuario desea visualizar la superficie normal, que es una columna de consumo energético en 3D
Funciones primarias	Trazo de la superficie normal en base a la gráfica de la elipse. Manipulación de la gráfica visualmente.
Necesidades no funcionales	La gráfica no demorará más de 10 segundos para graficarse.
Finalizaciones de los casos de uso	Finaliza una vez todas las superficies han sido graficadas.

Tabla 10 - Descripción de CU - Generar superficie normal



VI.3.1.1.5 Diagrama de caso de Uso - Generar gráfica de consumidores

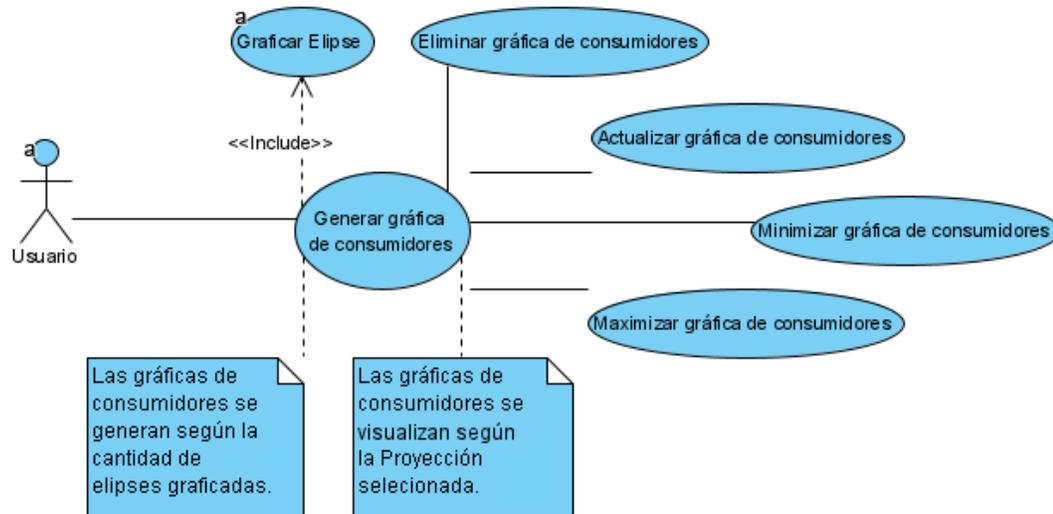


Ilustración 15 - DCU - Generar gráfica de consumidores

Título	Generar gráfica de consumidores
Inicios del caso de uso	Este verifica primero si ya hay elipses graficadas, para poder mostrar datos relacionados a la carga eléctrica de los consumidores.
Funciones primarias	Mostrar el consumo energético por hora de cada consumidor. Se pueden minimizar, maximizar, actualizar y eliminar las gráficas.
Necesidades no funcionales	Las gráficas serán inmediatamente disponibles una vez las elipses hayan sido graficadas.
Finalizaciones de los casos de uso	Siendo las gráficas visibles, el usuario podrá utilizar la información según su necesidad.

Tabla 11 - Descripción de CU - Generar gráfica de consumidores



VI.3.1.1.6 Diagrama de caso de Uso - Descargar plantilla de carga de datos



Ilustración 16 - DCU - Descargar plantilla de carga de datos

Título	Descargar plantilla de carga de datos
Inicios del caso de uso	Esta tiene como utilidad facilitar al usuario una plantilla donde él pueda ingresar los campos necesarios de carga eléctrica a subir en el sistema.
Funciones primarias	Proveer un esquema predeterminado para ingreso de cargas eléctricas sistemáticamente a los consumidores.
Necesidades no funcionales	La descarga de la plantilla no tiene restricción según el navegador.
Finalizaciones de los casos de uso	Finaliza una vez que el archivo haya sido descargado y el usuario pueda visualizarlo y abrirlo.

Tabla 12 - Descripción de CU - Descargar plantilla de carga de datos

VI.3.1.1.7 Diagrama de caso de Uso - Limitar Mapa

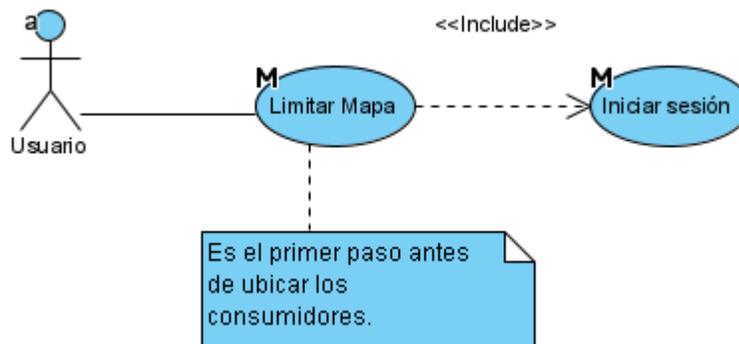


Ilustración 17 - DCU - Limitar Mapa



Título	Limitar Mapa
Inicios del caso de uso	Una vez el usuario inicie sesión y ubicar en el mapa el área donde se colocarán los consumidores.
Funciones primarias	Limitar la zona de trabajo para la ubicación de consumidores en el área a trabajar.
Necesidades no funcionales	La limitación se ejecuta instantáneamente para posterior ubicación de consumidores. Una vez los consumidores ya han sido establecidos en el mapa, la limitación no puede removerse.
Finalizaciones de los casos de uso	El caso de uso se completa una vez se visualiza una franja roja en el borde del mapa.

Tabla 13 - Descripción de CU - Limitar Mapa

VI.3.1.1.8 Diagrama de caso de Uso - Remover limitación de Mapa

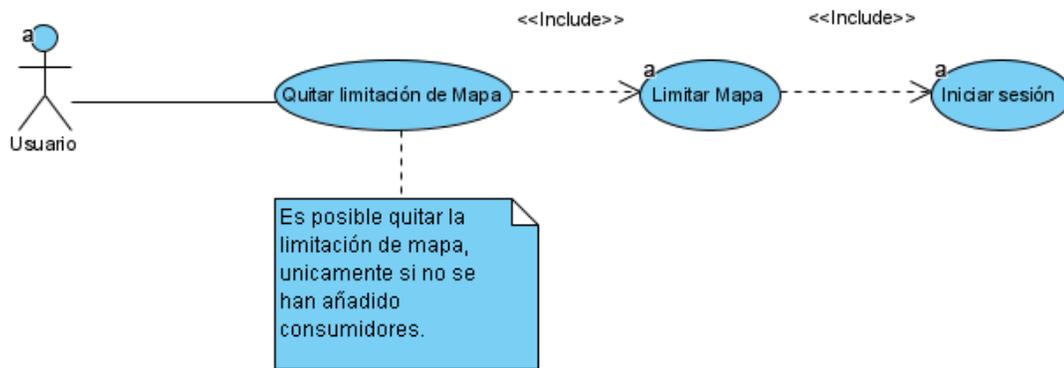


Ilustración 18 - DCU - Quitar limitación de Mapa

Título	Remover limitación de Mapa
Inicios del caso de uso	Es requisito que el mapa se encuentre previamente limitado.
Funciones primarias	Eliminar la restricción del área de trabajo.
Necesidades no funcionales	Esta se realiza inmediatamente en caso de no haber consumidores en el mapa.
Finalizaciones de los casos de uso	Termina el caso de uso al remover la limitación para que el usuario.

Tabla 14 - Descripción de CU - Remover limitación de Mapa



VI.3.1.1.9 Diagrama de caso de Uso - Reiniciar sistema

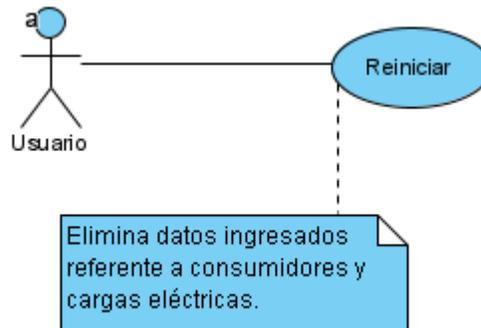


Ilustración 19 - DCU - Reiniciar

Título	Reiniciar
Inicios del caso de uso	Este caso de uso inicia cuando el usuario desea iniciar el sistema desde cero.
Funciones primarias	Restaurar valores de fábricas de todas las variables del sistema y gráficos para empezar a trabajar desde cero.
Necesidades no funcionales	Al efectuar el reinicio no deberá demorar más de 3 segundos para que el usuario pueda trabajar nuevamente.
Finalizaciones de los casos de uso	El sistema se visualizará como si este fuese cargado desde cero, a la espera de ingreso de valores.

Tabla 15 - Descripción de CU - Reiniciar

VI.3.1.1.10 Diagrama de caso de Uso - Poner pin de prueba

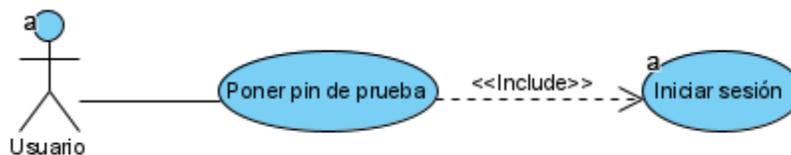


Ilustración 20 - DCU - Poner pin de prueba



Título	Poner pin de prueba
Inicios del caso de uso	El usuario puede poner un pin de prueba, a modo de visualizar dónde es deseada la ubicación, y dónde es sugerida por el sistema.
Funciones primarias	Este ha sido diseñado para temas geográficos de ubicación.
Necesidades no funcionales	Este necesita ser establecido una vez el mapa ya haya sido limitado.
Finalizaciones de los casos de uso	El Pin de prueba puede manipularse a lo largo de las proyecciones de elipses en el mapa.

Tabla 16 - Descripción de CU - Poner pin de prueba

VI.3.1.1.11 Diagrama de caso de Uso - Configurar vista de Mapa

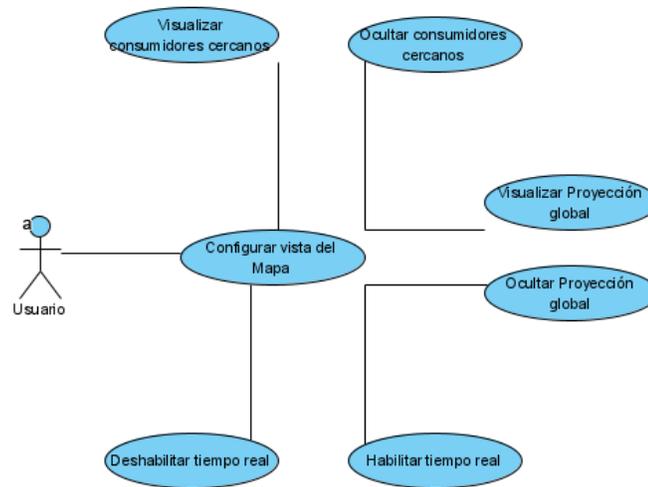


Ilustración 21 - DCU - Configurar vista del Mapa

Título	Configurar vista del mapa
Inicios del caso de uso	Este caso inicia, cuando el usuario desea modificar lo que visualiza en el mapa.
Funciones primarias	Se puede visualizar y ocultar a los consumidores cercanos en la zona, proyección global, y habilitación de cambios en tiempo real.
Necesidades no funcionales	Los cambios se aplican en menos de 1 segundo.
Finalizaciones de los casos de uso	Finaliza cuando el cliente configura la vista según su necesidad.

Tabla 17 - Descripción de CU - Configurar vista del mapa



VI.3.1.1.12 Diagrama de caso de Uso - Configurar vista proyección

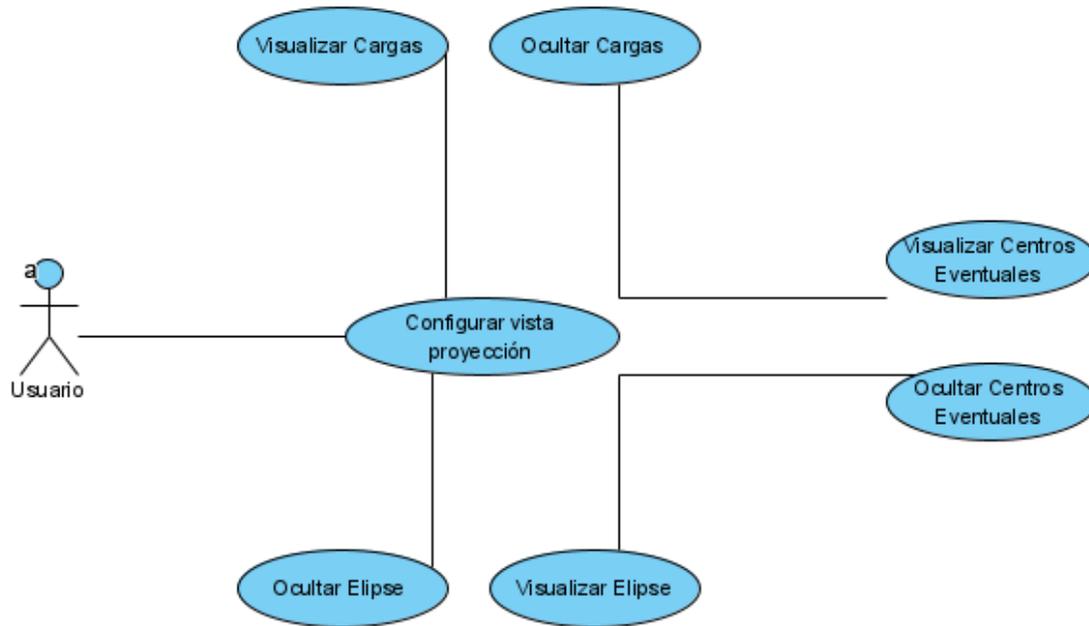


Ilustración 22 - DCU - Configurar vista proyección

Titulo	Configurar vista proyección
Inicios del caso de uso	Este caso inicia, cuando el usuario desea modificar la vista de la proyección actual.
Funciones primarias	Se puede visualizar y ocultar cargas, centros eventuales de cargas y la elipse de la proyección.
Necesidades no funcionales	Los cambios deberán aplicarse en menos de 1 segundo.
Finalizaciones de los casos de uso	Finaliza cuando el cliente configura la vista según su necesidad.

Tabla 18 - Descripción de CU - Configurar vista proyección



VI.3.1.1.13 Diagrama de caso de Uso - Configurar vista global

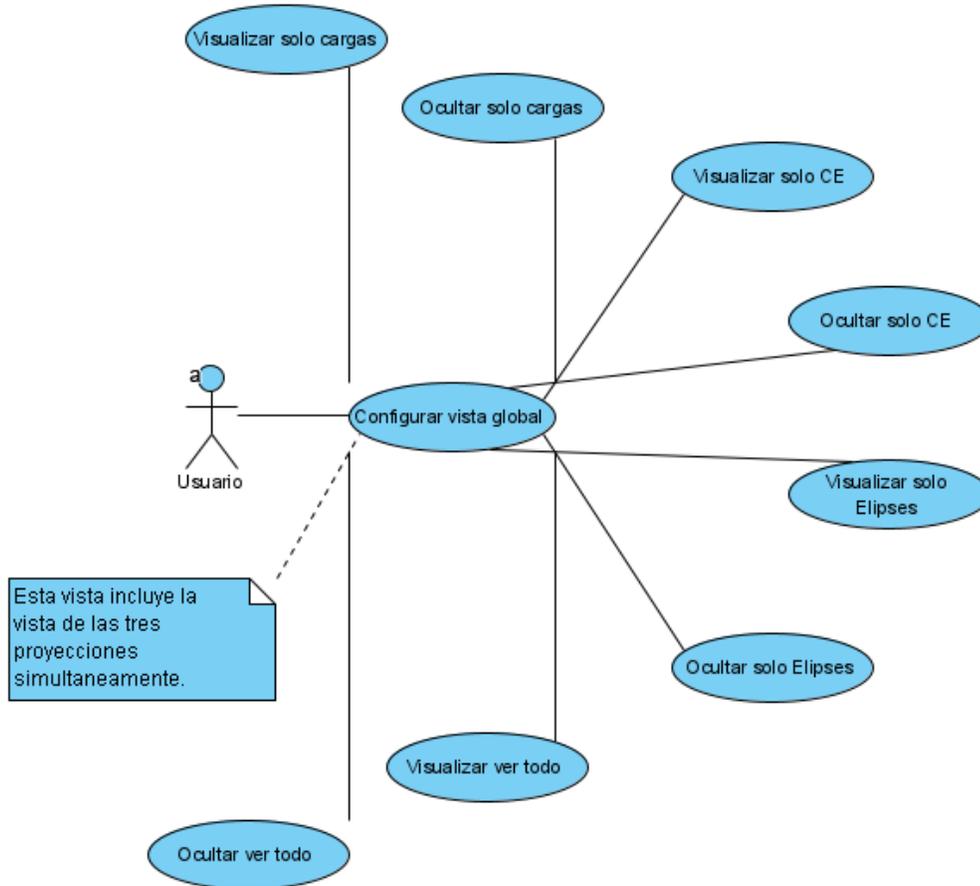


Ilustración 23 - DCU - Configurar vista global

Título	Configurar vista global
Inicios del caso de uso	Este caso inicia, cuando el usuario desea modificar la vista de la proyección global en el mapa.
Funciones primarias	Visualizar y ocultar las cargas de los consumidores, los centros eventuales de cargas, las elipses, y todos los elementos en el mapa.
Necesidades no funcionales	Se aplican cambios en tiempo real, siendo visualizado en el mapa.
Finalizaciones de los casos de uso	Finaliza cuando el cliente configura la vista según su necesidad.

Tabla 19 - Descripción de CU - Configurar vista global



VI.3.1.1.14 Diagrama de caso de Uso - Generar reportes

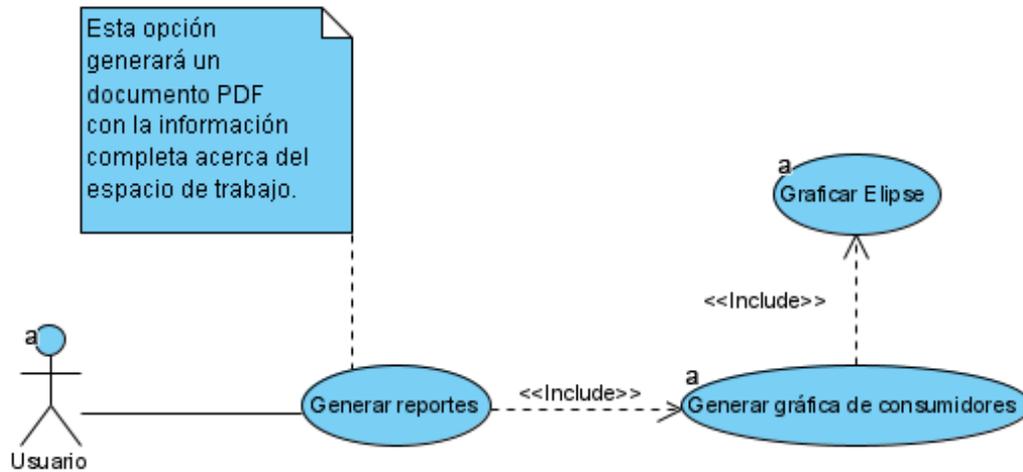


Ilustración 24 - DCU - Generar reportes

Título	Generar reportes
Inicios del caso de uso	Para la generación de reportes es necesario tener una gráfica de consumidores y elipses graficadas.
Funciones primarias	Recopilación de los datos mostrados en el sistema, entre ellos, variables, consumidores con cargas eléctricas, y proyecciones.
Necesidades no funcionales	La información es mostrada en un archivo pdf y se descarga automáticamente localmente.
Finalizaciones de los casos de uso	El caso termina cuando el usuario puede acceder al archivo pdf descargado.

Tabla 20 - Descripción de CU - Generar reportes



VI.3.1.1.15 Diagrama de caso de Uso - Visualizar reportes

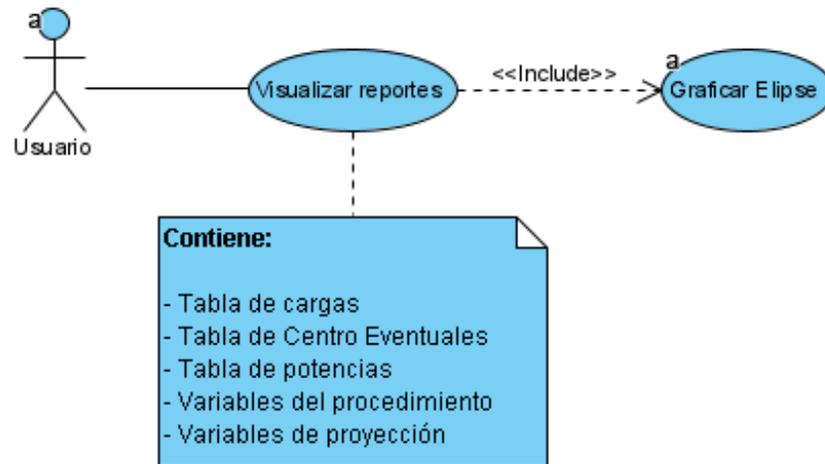


Ilustración 25 - DCU - Visualizar reportes

Título	Visualizar reportes
Inicios del caso de uso	Este caso de uso comienza con la necesidad del usuario de visualizar en el sistema los datos dinámicamente. Como requisito, este solicita el trazo de elipses.
Funciones primarias	Presentar un resumen de información en tablas respecto a cargas, centros eventuales, potencias, variables de procedimiento, y variables de proyección.
Necesidades no funcionales	Estos son cargados en tiempo real en la sección de Reportes.
Finalizaciones de los casos de uso	Una vez el usuario visualiza los reportes este caso de uso finaliza.

Tabla 21 - Descripción de CU - Visualizar reportes



VI.3.1.1.16 Diagrama de caso de Uso - Visualizar Proyecciones

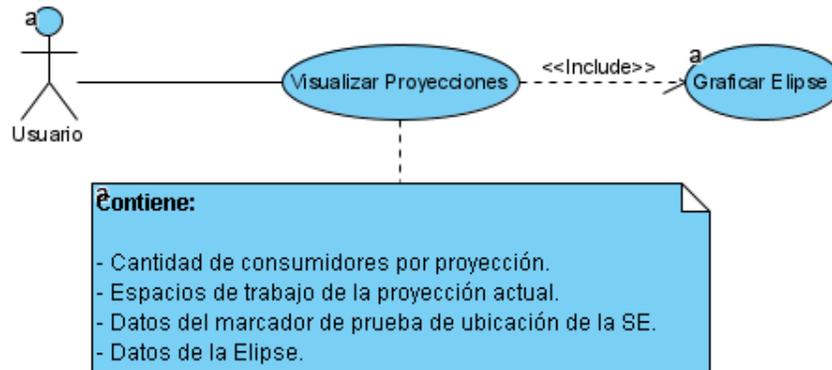


Ilustración 26 - DCU - Visualizar proyecciones

Título	Visualizar Proyecciones
Inicios del caso de uso	Este caso de uso inicia cuando el usuario desea visualizar más información respecto a las proyecciones de las elipses.
Funciones primarias	Conocer valores de la cantidad de consumidores, espacios de trabajo y su restauración, datos del pin de prueba y su coincidencia dentro de la zona limitada del mapa.
Necesidades no funcionales	Carga automática de los datos en la sección de proyecciones, esta información es obtenida una vez las proyecciones son realizadas.
Finalizaciones de los casos de uso	Termina el caso de uso cuando el usuario visualiza los datos en el sistema.

Tabla 22 - Descripción de CU - Visualizar Proyecciones



VI.3.1.1.17 Diagrama de caso de Uso - Configurar el sistema

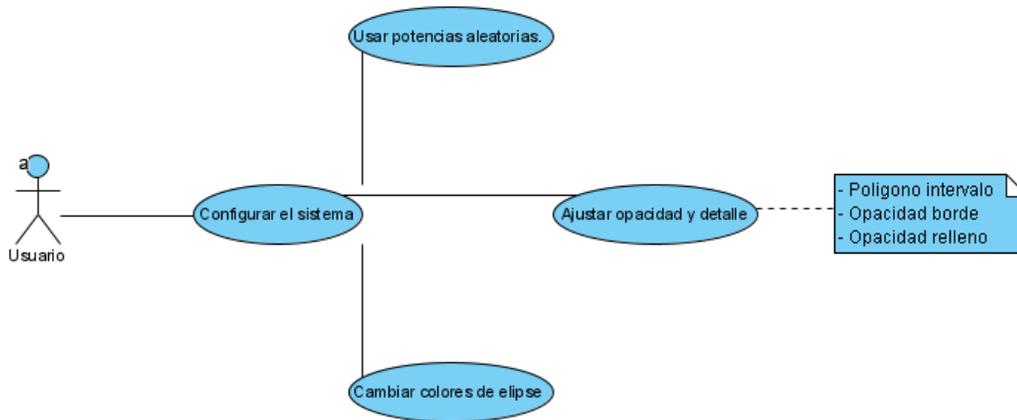


Ilustración 27 - DCU - Configurar el sistema

Título	Configurar el sistema
Inicios del caso de uso	El caso de uso toma lugar al momento que el usuario desea personalizar la vista en las proyecciones.
Funciones primarias	Seleccionar entre potencias aleatorias o uso de carga de datos. Modificar la opacidad, detalle y colores de componentes de la elipse.
Necesidades no funcionales	Cambios se aplican inmediatamente.
Finalizaciones de los casos de uso	Este finaliza cuando el usuario desea aplicar cambios en el sistema.

Tabla 23 - Descripción de CU - Configurar el sistema

VI.3.2 Modelo conceptual

Los principales elementos de modelado UML utilizados en el modelo conceptual son: clase, asociación y paquete. Estos se representan gráficamente mediante la notación UML (1999). [15] Podemos expresar estos diagramas como una representación gráfica que sirve para representar la estructura de un sistema que será implementado utilizando un lenguaje orientado a objetos.



VI.3.2.1 Diagrama de clases de sistema web GLOCSE

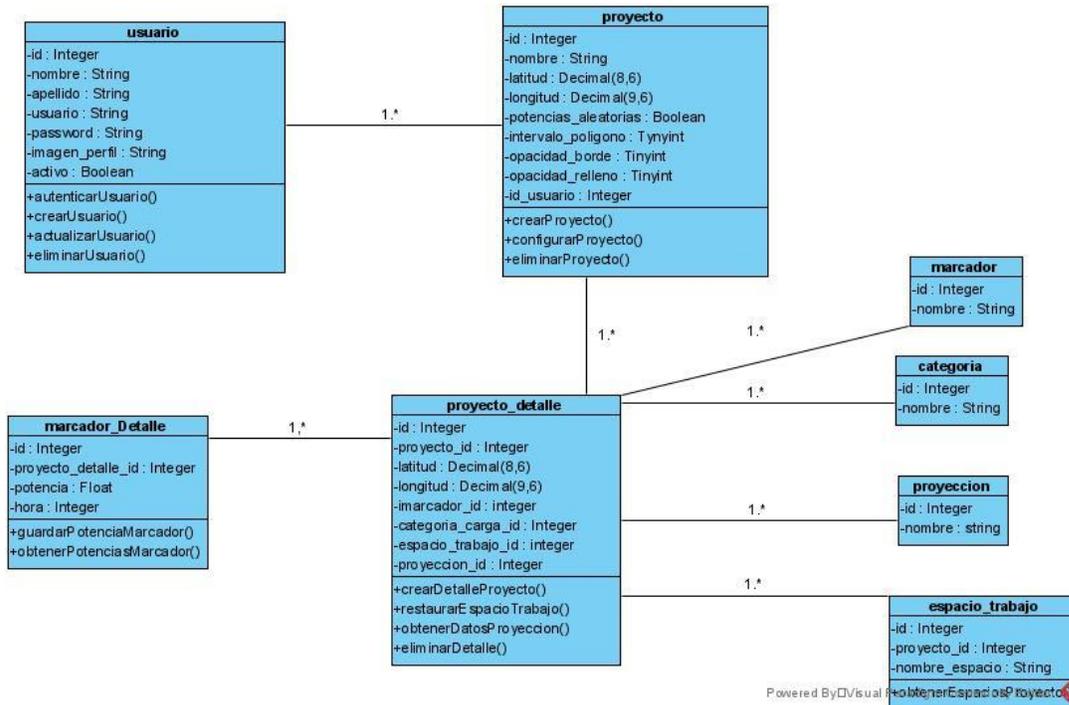


Ilustración 28 - DC – GLOCSE

Descripción: El diagrama de clases del sistema está compuesta únicamente por 8 clases, guardando únicamente los valores necesarios para que cada usuario que se registre y autentique en la aplicación pueda guardar y dar seguimiento a sus proyectos sin tener que estar creando el proyecto de 0 cada vez que usa la aplicación.

VI.3.3 Modelo de espacio de navegación y modelo de estructura de navegación

VI.3.3.1 Diagramas de navegación o de actividades

Los diagramas de navegación, también conocidos como diagramas de actividades o de estados, constituyen un medio a través del cual podemos captar la comprensión de las personas a las que llamamos “expertos del dominio”.

Estos diagramas son buenos para captar detalles en un sentido general y de una manera en que los expertos del dominio pueden examinar, aclarar y mejorar. [24]



Estos diagramas son conocidos por proveer un panorama de cómo será el espacio de navegación que compondrá al sistema web GLOCSE y cómo será la experiencia del usuario.

VI.3.3.1.1 Diagrama de navegación – Módulo Mapa

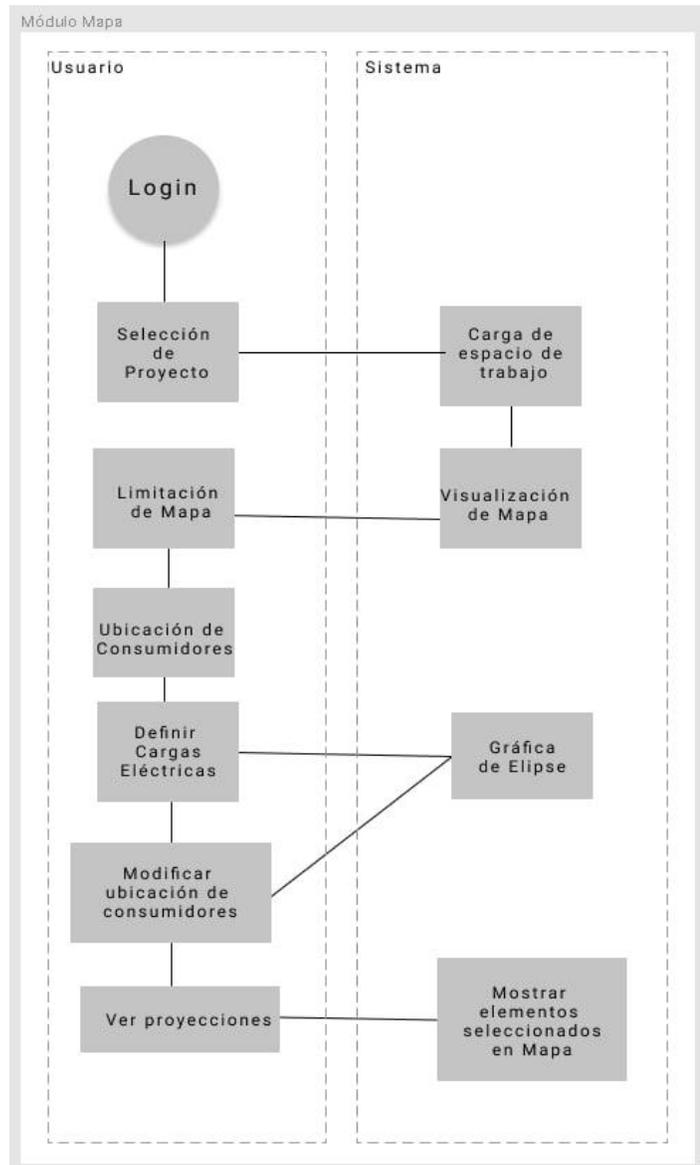


Ilustración 29 - DN - Módulo Mapa

Descripción: Diagrama de navegación que presenta las opciones que un usuario tiene habilitadas al momento de acceder al módulo Mapa.



VI.3.3.1.2 Diagrama de navegación – Módulo Gráficas

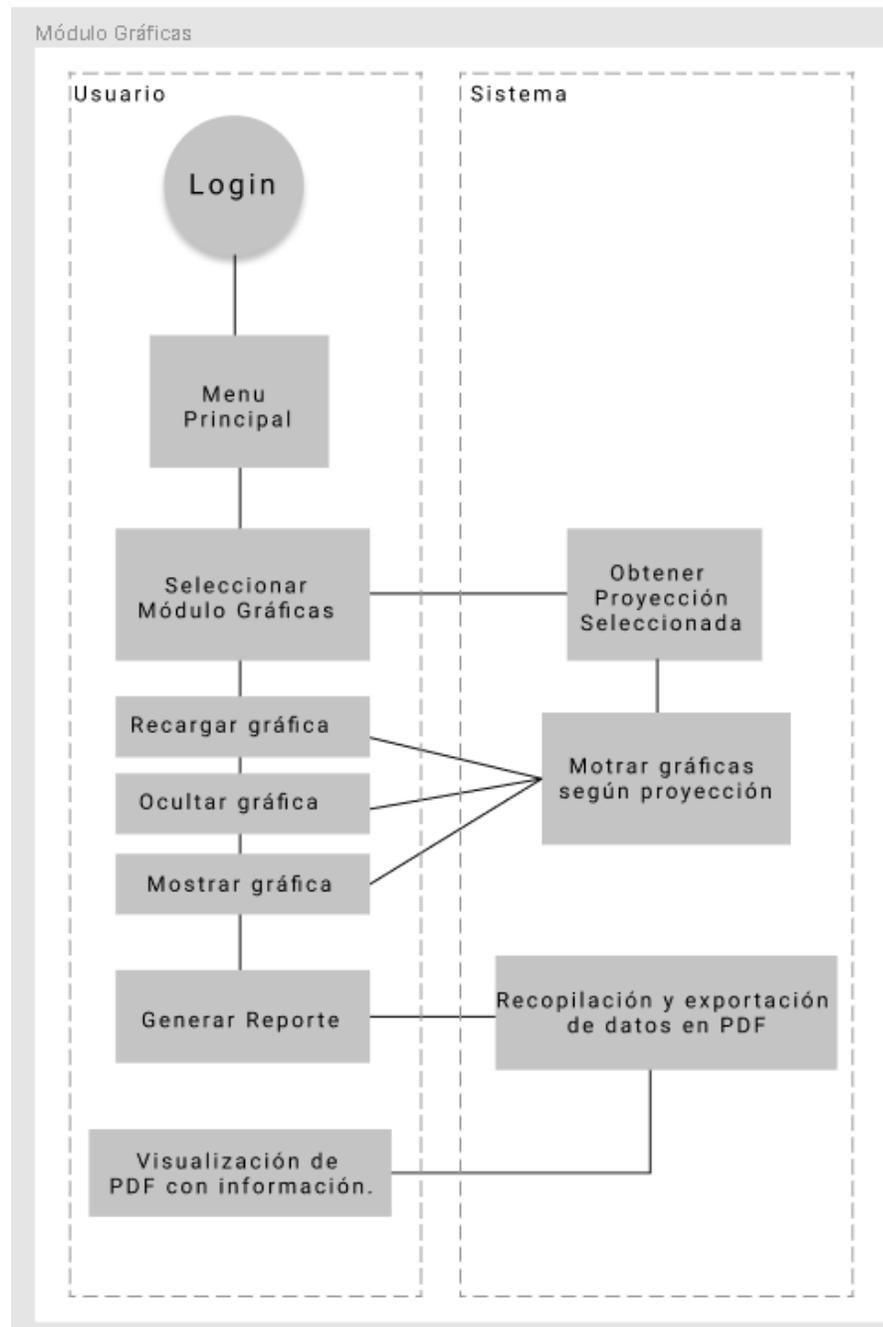


Ilustración 30 - DN - Módulo Gráficas

Descripción: Diagrama de navegación del módulo Gráficas, que provee al usuario de visualización y manipulación de las gráficas según la proyección.



VI.3.3.1.3 Diagrama de navegación – Módulo Reportes

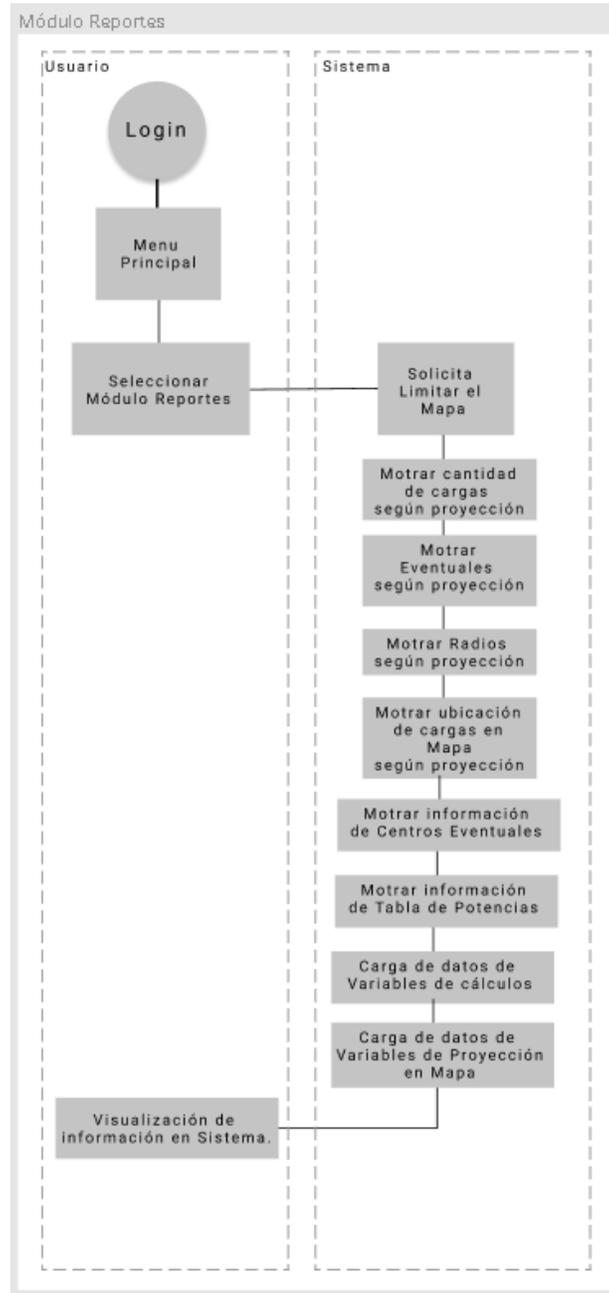


Ilustración 31 - DN - Módulo Reportes

Descripción: Diagrama de navegación del módulo Reportes, que provee al usuario de visualización de datos específico de una proyección según la selección realizada.



VI.3.3.1.4 Diagrama de navegación – Módulo Proyecciones

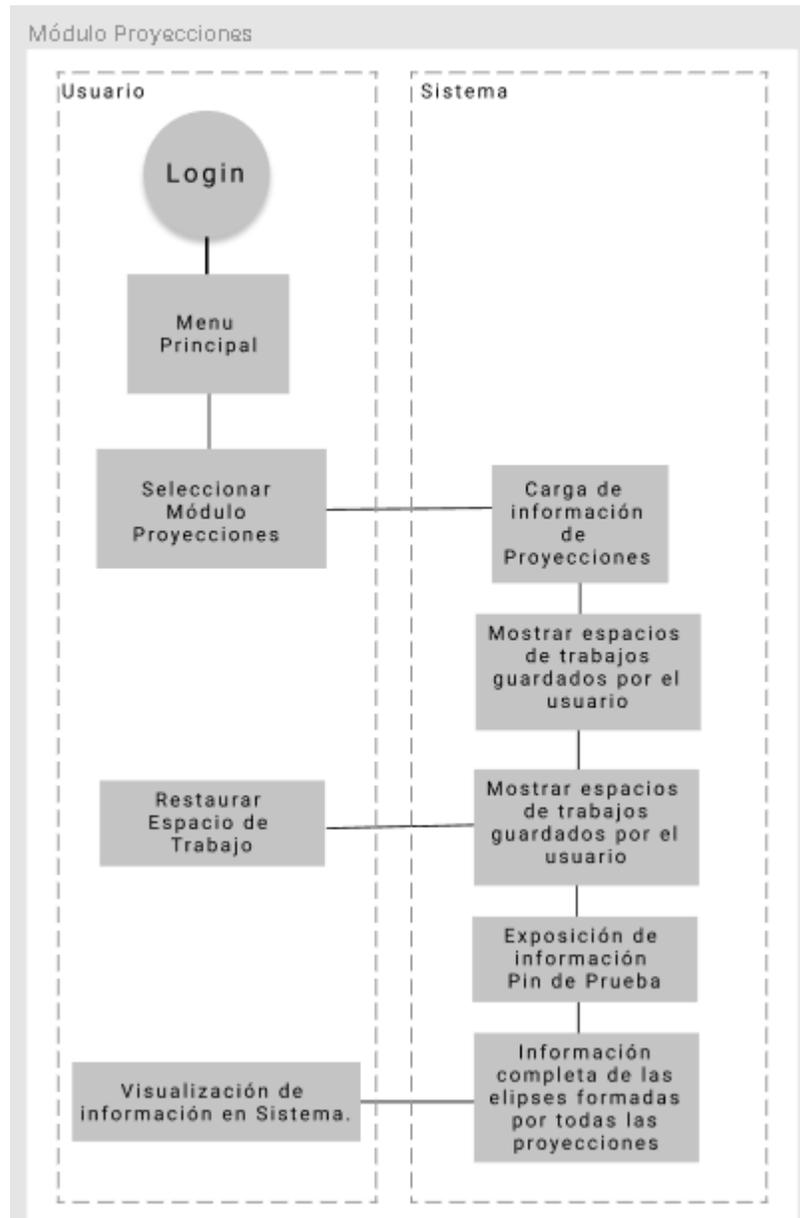


Ilustración 32 - DN - Módulo Proyecciones

Descripción: Diagrama de navegación del módulo Proyecciones, que provee al usuario de visualización de datos generales de las proyecciones generadas en el sistema.



VI.3.3.1.5 Diagrama de navegación – Módulo Configuración

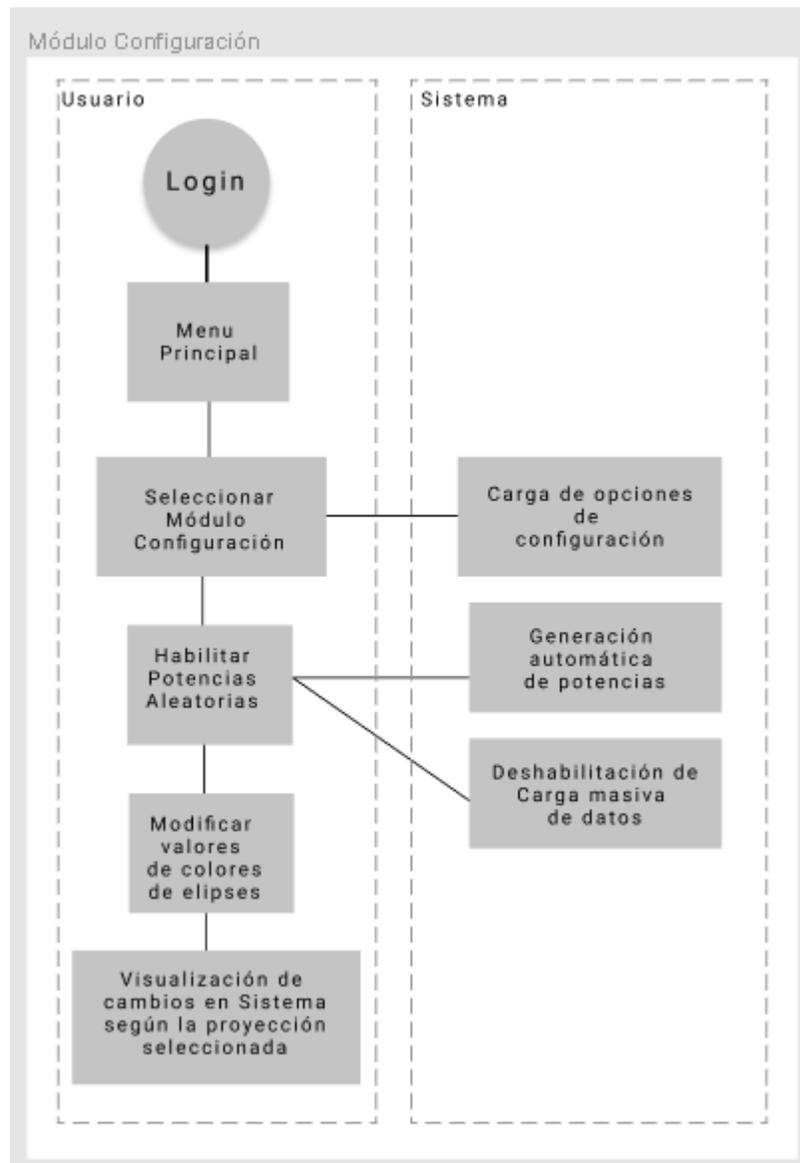


Ilustración 33 - DN - Módulo Configuración

Descripción: Diagrama de navegación del módulo Configuración, en dónde se da opción al usuario de modificar el estilo de las proyecciones, modificando opacidad, colores y habilitando potencias aleatorias.



VI.3.3.2 Diagrama de secuencia

El diagrama de secuencias consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical. [25] Estas interacciones entre objetos muestran una estructura de navegación que representa al sistema web GLOCSE.

VI.3.3.2.1 Diagrama de secuencia - Autenticación de Usuario

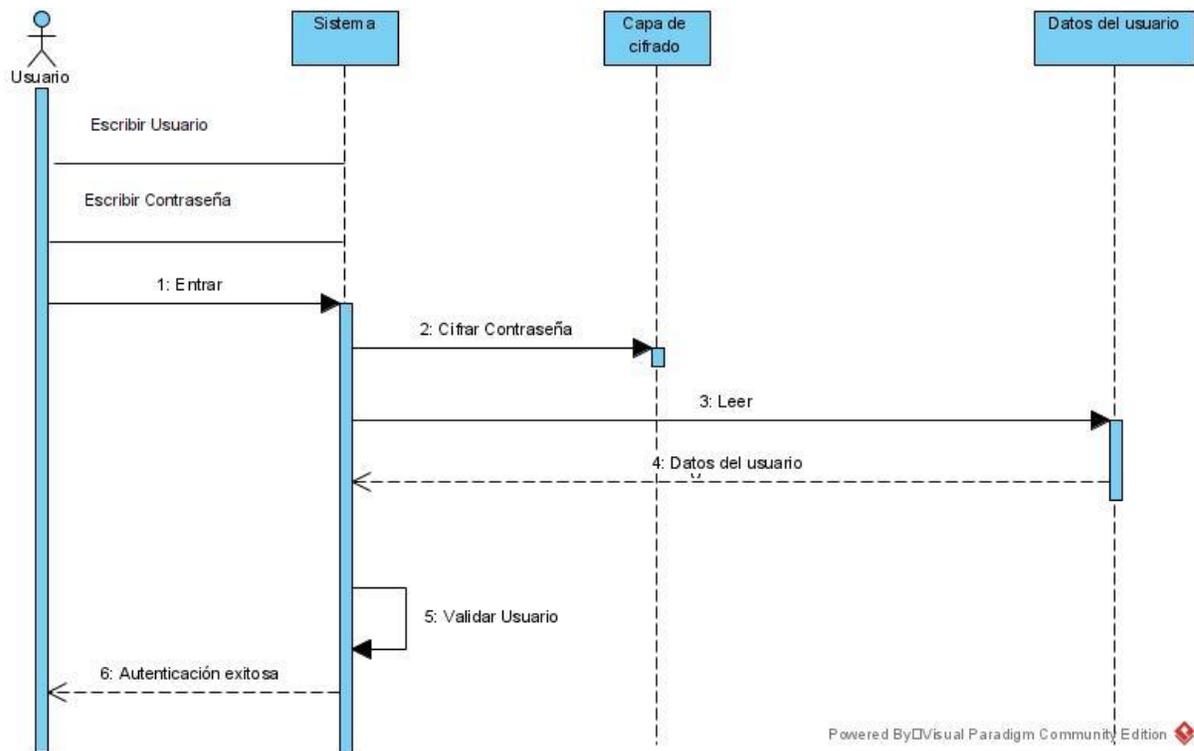


Ilustración 34 - DS - Autenticación de Usuario

Descripción: Diagrama de secuencia del proceso de autenticación de Usuario, el diagrama muestra la interacción entre los objetos del usuario y el sistema al momento que este se autentica en el sistema.



VI.3.3.2 Diagrama de secuencia - Creación de proyectos

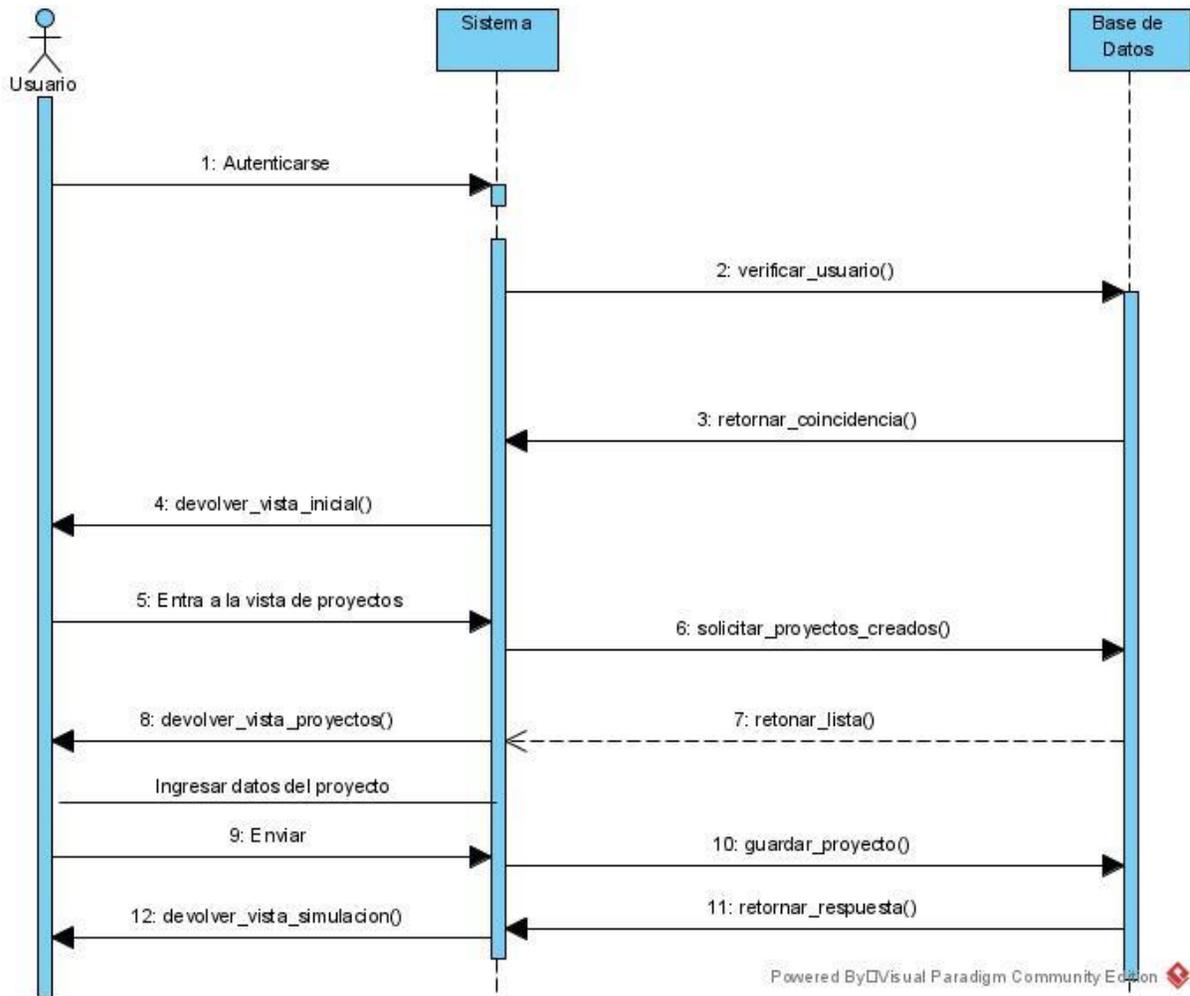


Ilustración 35 - DS - Creación de proyectos

Descripción: Diagrama de secuencia del proceso de creación de proyectos, el usuario se encarga de crear y gestionar sus propios proyectos al terminar de crear o cargar el proyecto el sistema lo redirecciona al simulador.



VI.3.3.2.3 Diagrama de secuencia - Creación de proyección

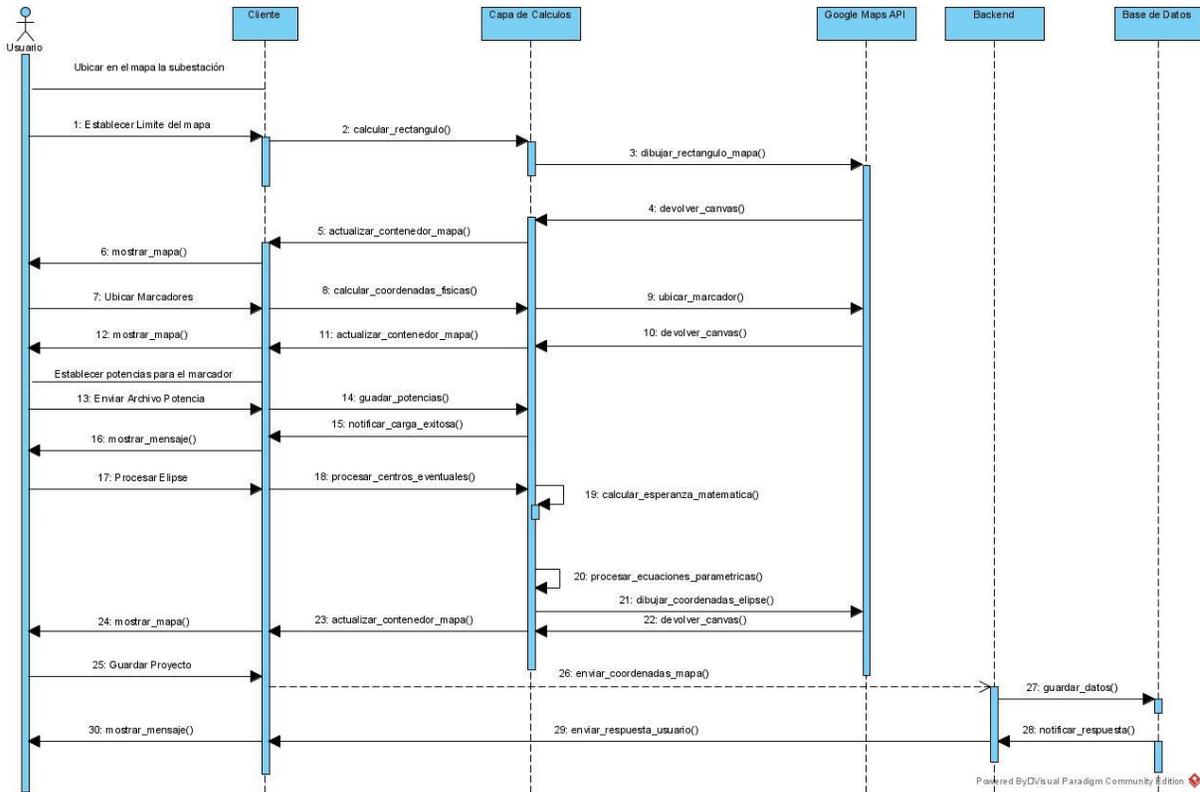


Ilustración 36 - DS - Creación de proyección

Descripción: Diagrama de secuencia del proceso de creación de una proyección, una vez que el usuario se encuentre en la vista de simulación, se procede a limitar el mapa, escoger la ubicación de los marcadores de potencia, subir la plantilla de cargas y finalmente a generar la elipse.



VI.3.3.2.4 Diagrama de secuencia - Creación de múltiples proyecciones

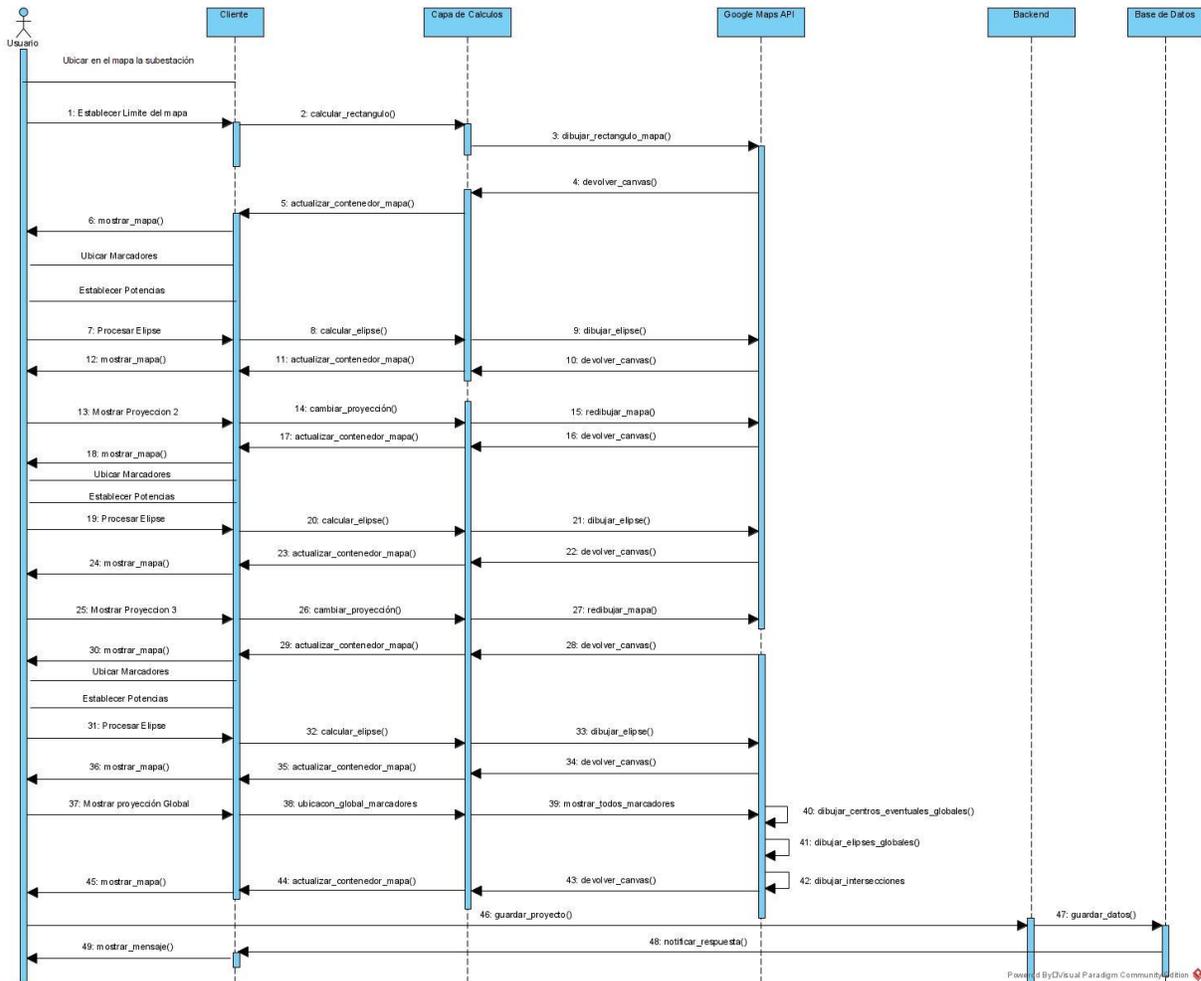


Ilustración 37 - DS - Creación de múltiples proyecciones

Descripción: Diagrama de secuencia del proceso de creación de múltiples proyecciones, el usuario puede trabajar con un total de hasta 3 proyecciones en el mismo proyecto dichas proyecciones se adecuan de acuerdo con las varianzas de cargas en los próximos 30 años, luego de procesar la proyección global se muestra el polígono resultante de la intersección de las 3 elipses.



VI.3.3.2.5 Diagrama de secuencia - Generación de reporte

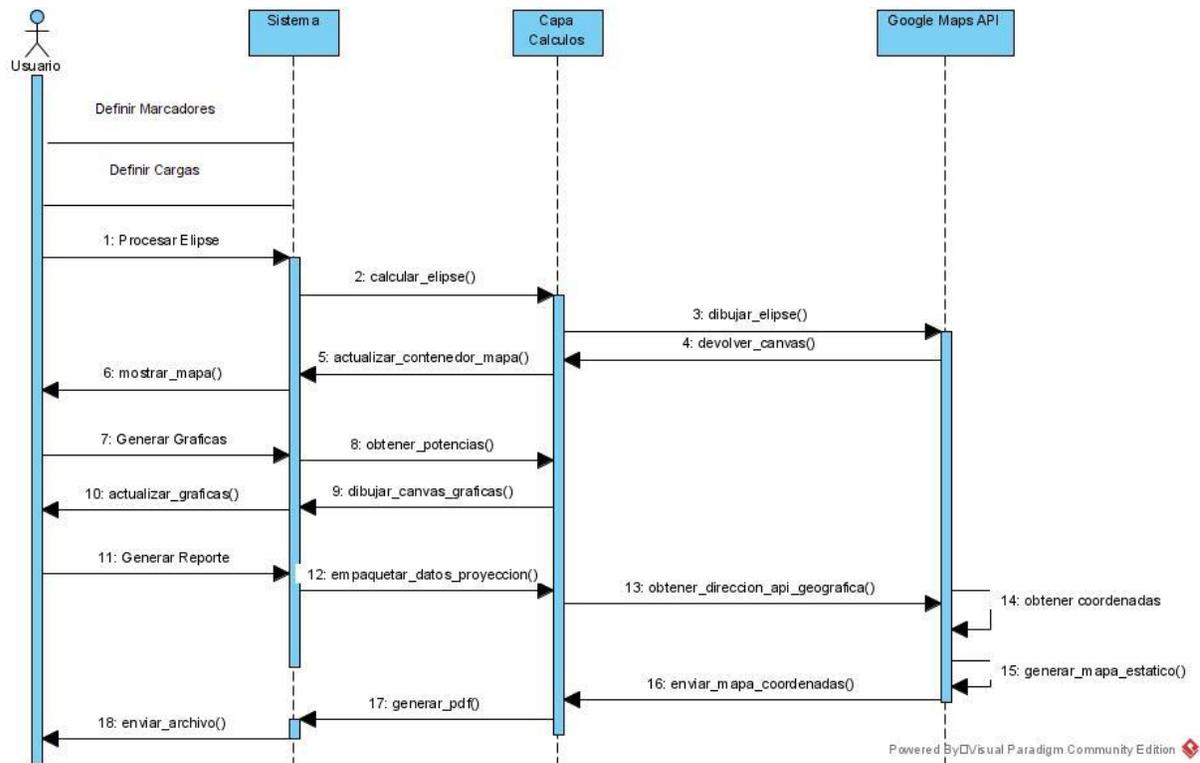


Ilustración 38 - DS - Generación de reporte

Descripción: Diagrama de secuencia del proceso de generación de reporte, al procesar la elipse el usuario cuenta con un reporte completo con los resultados de los cálculos de todas las variables involucradas en el proceso.



VI.3.4 Modelo de presentación

VI.3.3.1 Diagramas de presentación

A continuación, se presentarán los bosquejos de diseño de cada vista que compondrá al prototipo de sistema web GLOCSE.

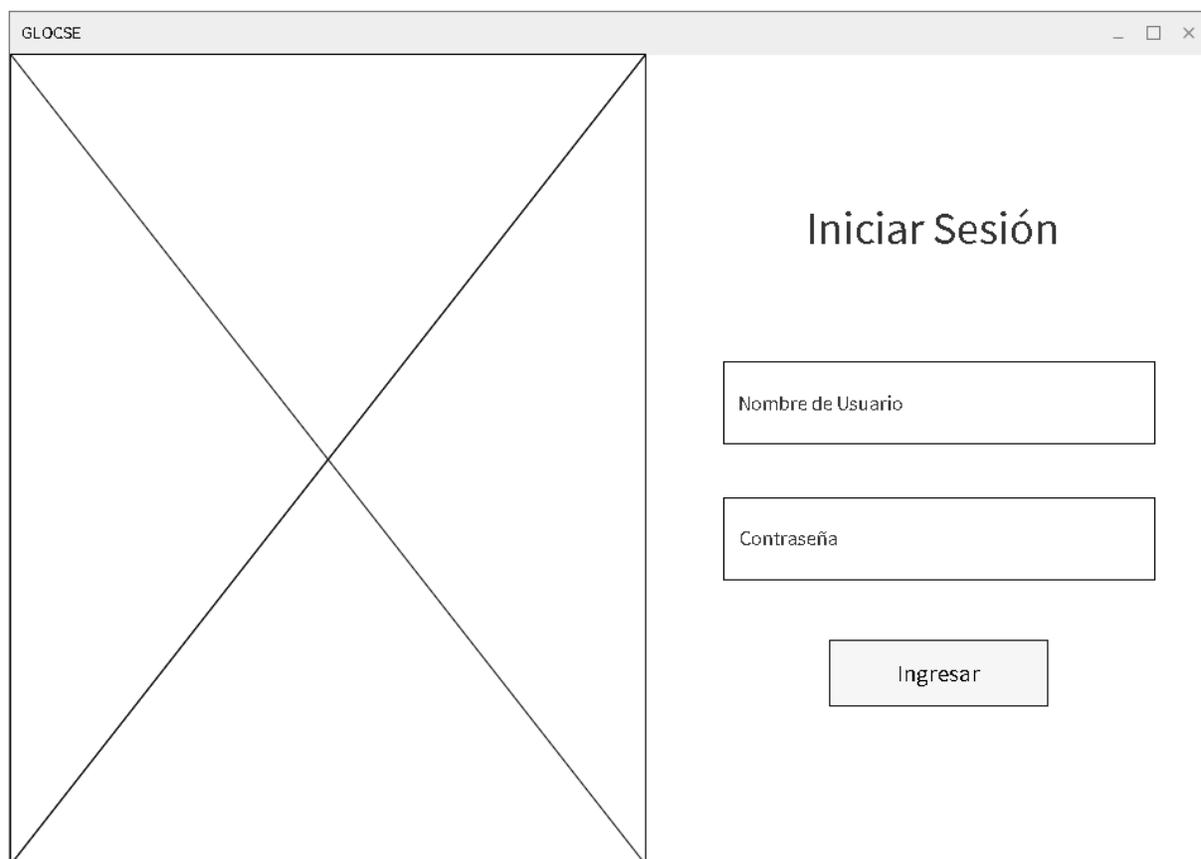


Ilustración 39 - Vista Inicio de sesión

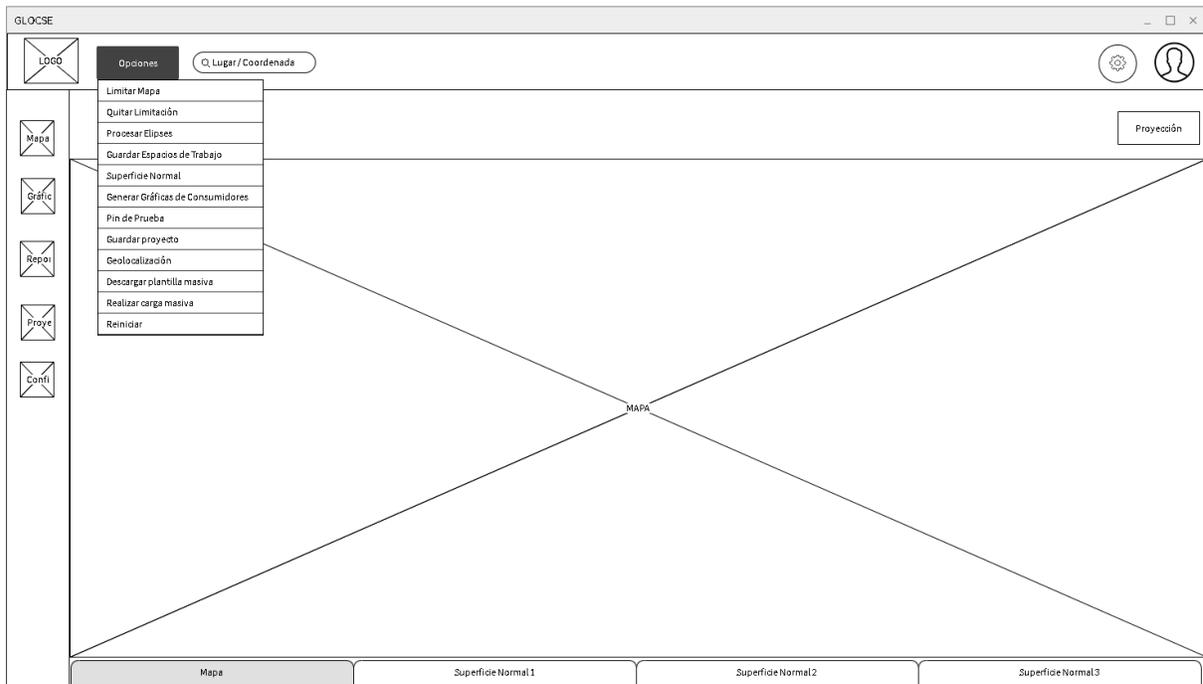


Ilustración 40 - Vista Menú Principal

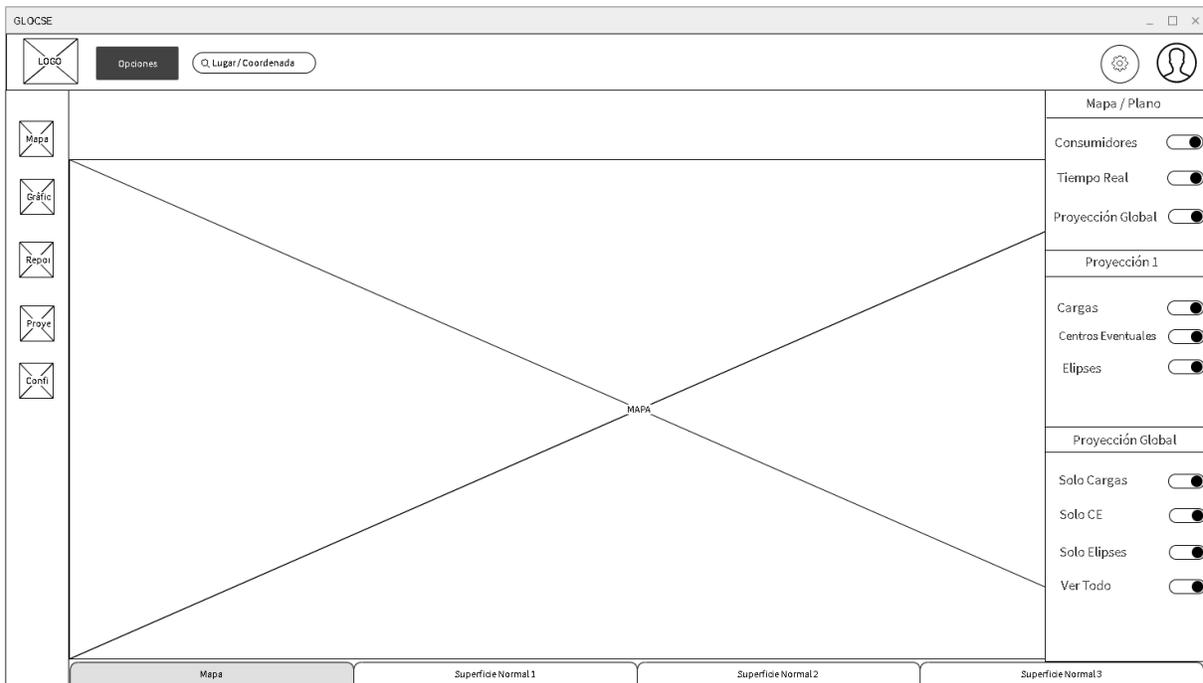


Ilustración 41 - Vista Configuración Menú Principal

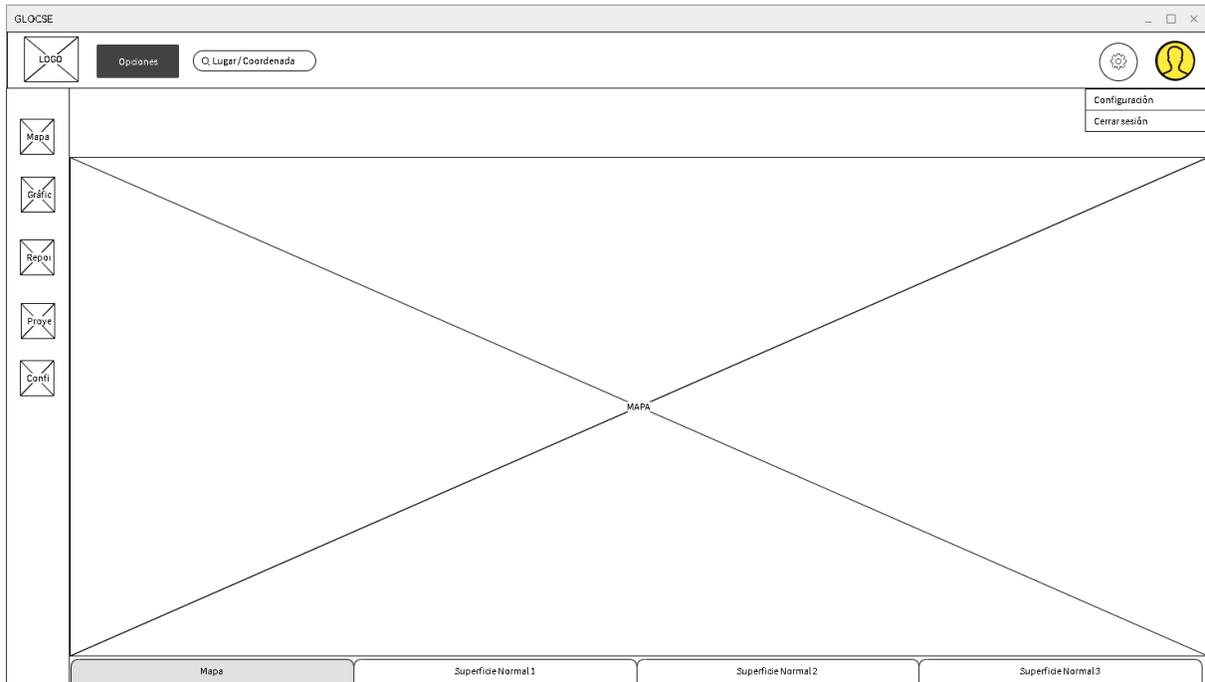


Ilustración 42 - Vista Opciones de Usuario

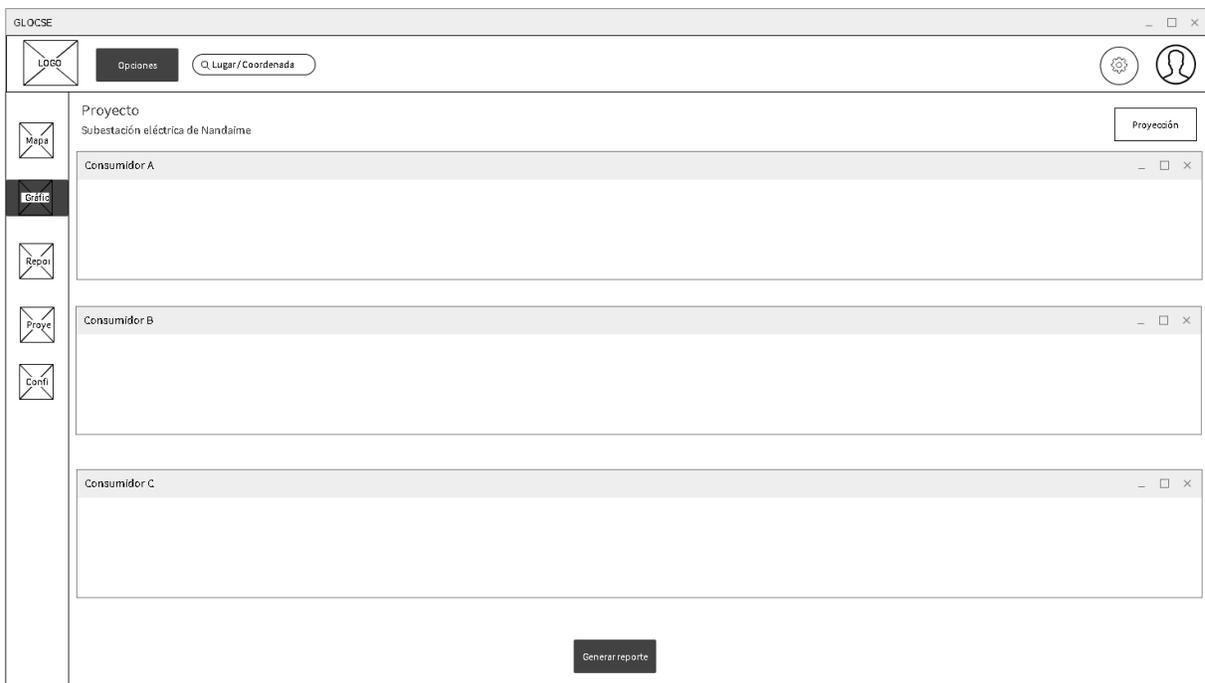


Ilustración 43 - Vista Módulo Gráficas



GLOCSE

LOGO Opciones Q. Lugar / Coordenada

Proyecto
Subestación eléctrica de Nandaime Proyección

Tabla de cargas

Carga	Categoría	Cx	Cy	Latitud	Longitud	Mapa

Tabla de Centros Eventuales

Centro Eventual	Cx	Cy	Latitud	Longitud	Mapa

Tabla de Potencias

Hora	A	B	C	D	E	F	G	H	I	J	Sumatoria	Minimo	Media	Maximo

Variables del Procedimiento

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

Variables de la Proyección

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

Ilustración 44 - Vista Módulo Reportes

GLOCSE

LOGO Opciones Q. Lugar / Coordenada

Proyecto
Subestación eléctrica de Nandaime Proyección

0/10 Proyección 1 Consumidores | 0/10 Proyección 2 Consumidores | 0/10 Proyección 3 Consumidores | 0/10 Proyección Final

ESPACIOS DE TRABAJO DE LA PROYECCIÓN ACTUAL

Identificador	Restaurar

MARCADOR DE PRUEBA DE UBICACIÓN DE LA SUBESTACIÓN ELECTRICA

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

DATOS DE LAS ELIPSES

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6
- Item 7

Ilustración 45 - Vista Módulo Proyecciones

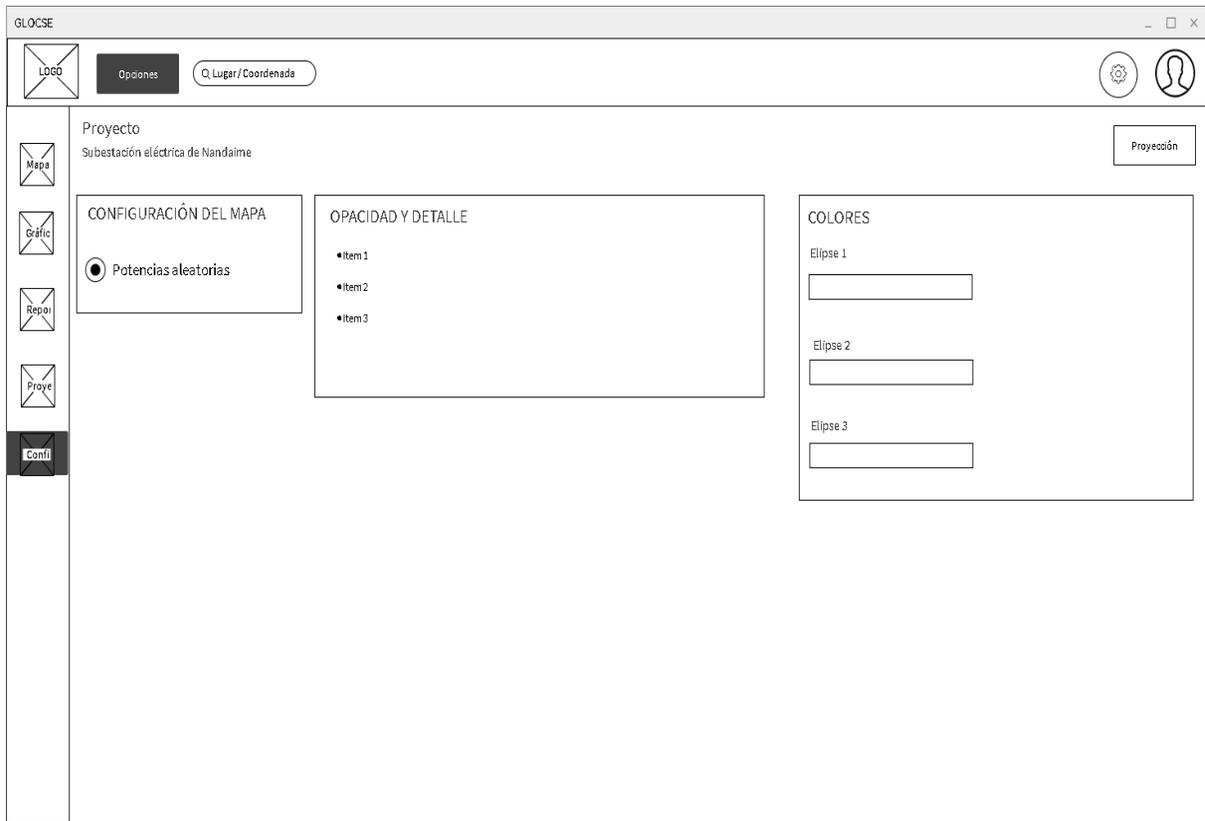


Ilustración 46 - Vista Módulo Configuración



VI.3.3.1 Modelo arquitectónico

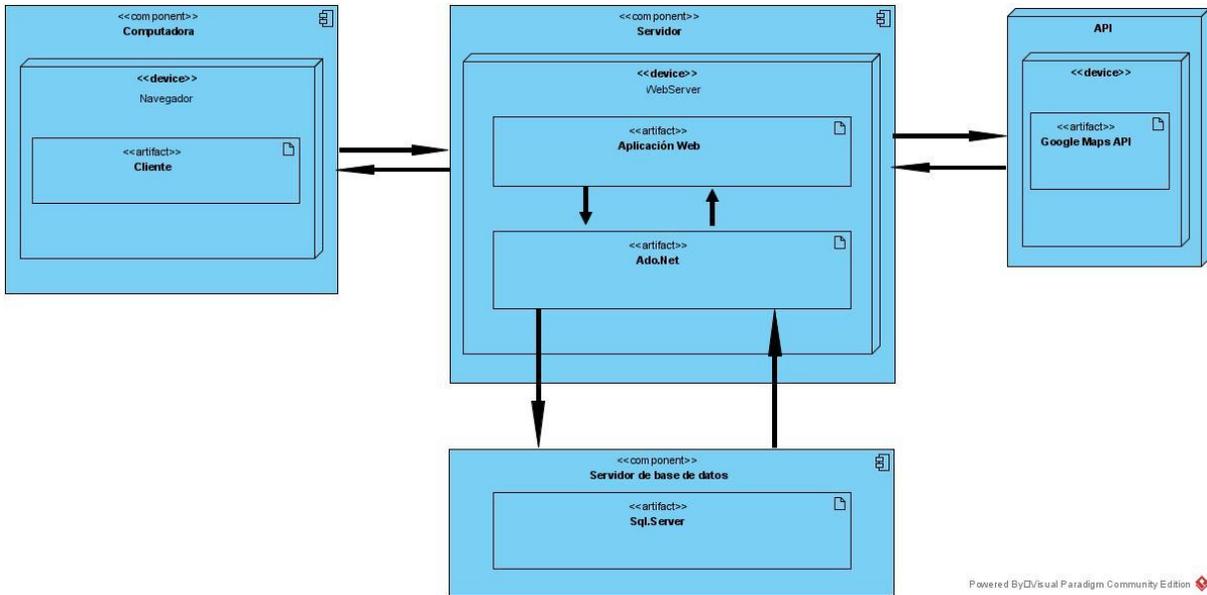


Ilustración 47 - Modelo arquitectónico

Descripción: Diagrama de modelo arquitectónico de la aplicación, la aplicación consta de una arquitectura sencilla y fácil de implementar.

VI.4 Codificación del software

En esta fase se procederá con la codificación del software de información web considerando lo especificado en la etapa de diseño del sistema, esto se llevará a cabo utilizando las herramientas de desarrollo web de .NET (MVC Net Core), JQuery, Bootstrap 4, SQL server como sistema gestor de base de datos, GitHub para el control de versiones del código fuente, generando un sistema de información web que cuenta con una aplicación cliente multiplataforma y responsiva, un BackEnd que proporcione mecanismos de seguridad en el acceso y comunicación segura con la base de datos.



Para el desarrollo de este software se utilizó el IDE Visual Studio 2019, Git para la administración de cambios realizados al sistema, para la revisión y modificación del código del sistema.

VI.4.1 Lenguaje de programación y frameworks

El lenguaje de programación para la creación del Web Server es C# con Visual Studio 2019 Community Edition, De lado cliente se ocupó Bootstrap 4, JQuery y Ajax para brindar una experiencia agradable y funcional al usuario.

VI.4.1.1 Instalación de Visual Studio

Para realizar la instalación de Visual Studio es necesario ingresar a su página oficial <https://visualstudio.microsoft.com/es/>, luego procedemos a realizar la descarga de la versión Community y luego ejecutamos el instalador.

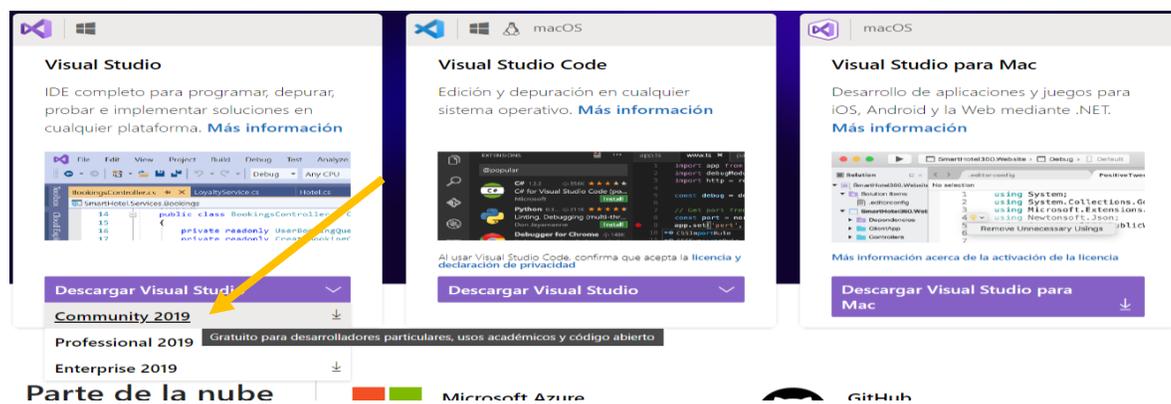


Ilustración 48 - Descargar Visual Studio Community

```
orivera04 @ DESKTOP-8D6COP4 in ~ | 21:04:48
$ cd Github/
/home/orivera04/Github
orivera04 @ DESKTOP-8D6COP4 in ~/Github | 21:04:49
$ mkdir Glocse
```

Ilustración 49 - Creación de directorio con nombre de Proyecto



Una vez que haya terminado la instalación procedemos a inicializar el repositorio que contendrá nuestro proyecto, primeramente, creamos el directorio.

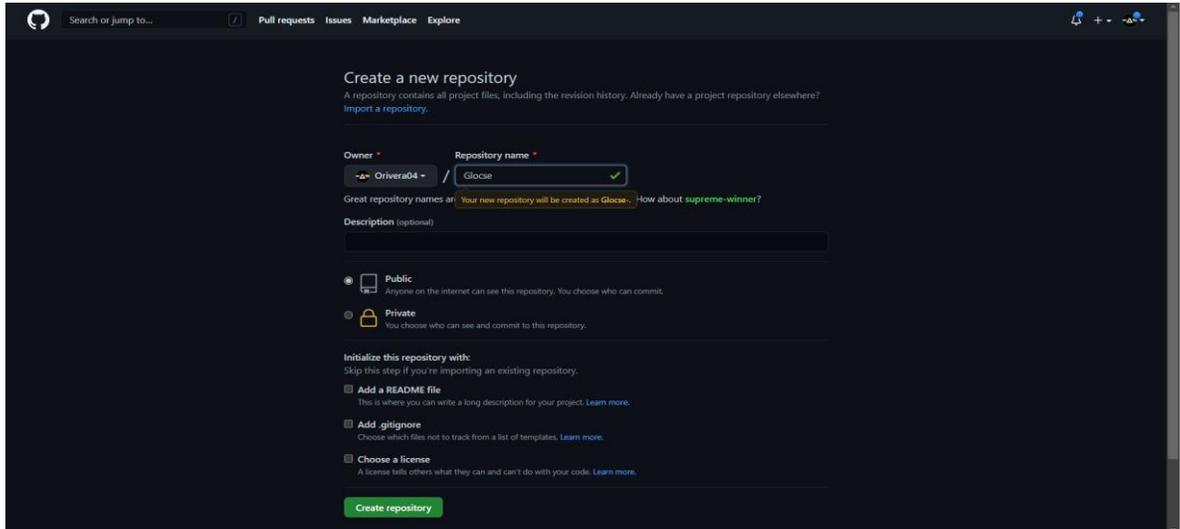


Ilustración 50 - Creación de un repositorio

Accedemos con nuestra cuenta de GitHub y creamos un nuevo repositorio como se muestra en la imagen.

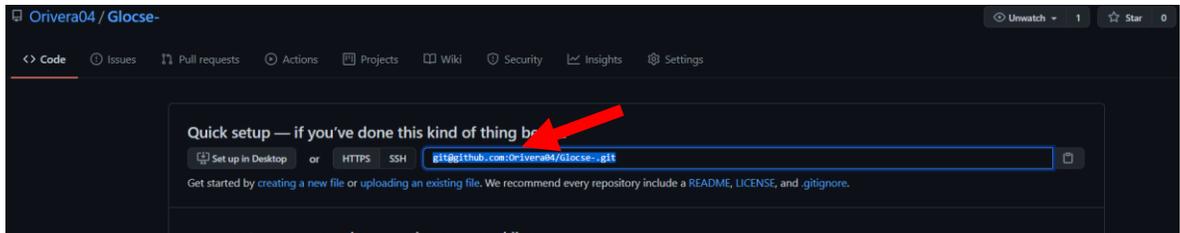


Ilustración 51 - Dirección del repositorio creado

Luego de haberlo creado copiamos esta dirección para poder clonar el repo en nuestra carpeta que creamos previamente.



```
gorivera04 @ DESKTOP-8D6COP4 in ~/.../Glocse/.git | 21:11:36 |master ✓|
$ git clone git@github.com:Orivera04/Glocse-.git
Cloning into 'Glocse-'...
Enter passphrase for key '/home/orivera04/.ssh/id_ed25519':
warning: You appear to have cloned an empty repository.
orivera04 @ DESKTOP-8D6COP4 in ~/.../Glocse/.git | 21:11:43 |master ✓|
$ |
```

Ilustración 52 - Clonación de proyecto

Lo clonamos y ya tendremos inicializado nuestro proyecto con Git con la rama por defecto **Máster**.

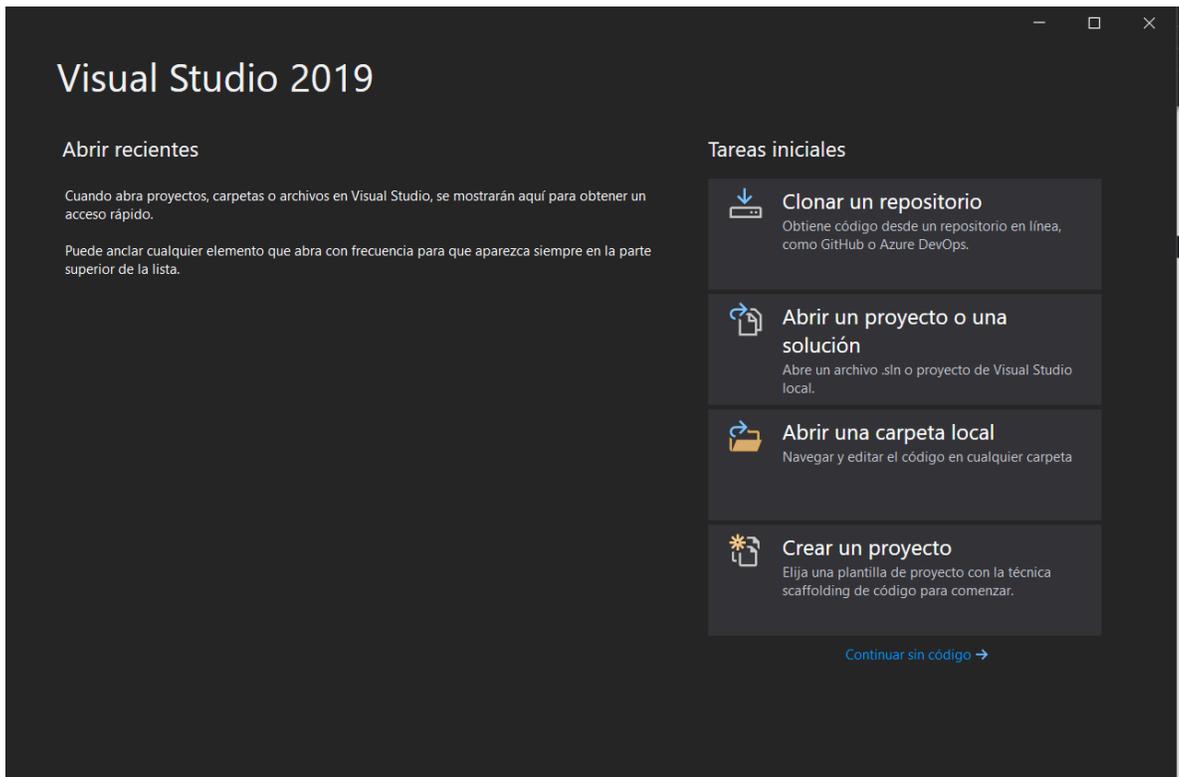


Ilustración 53 - Creación de proyecto en Visual Studio



En **Visual Studio** escogemos la opción de crear un proyecto.

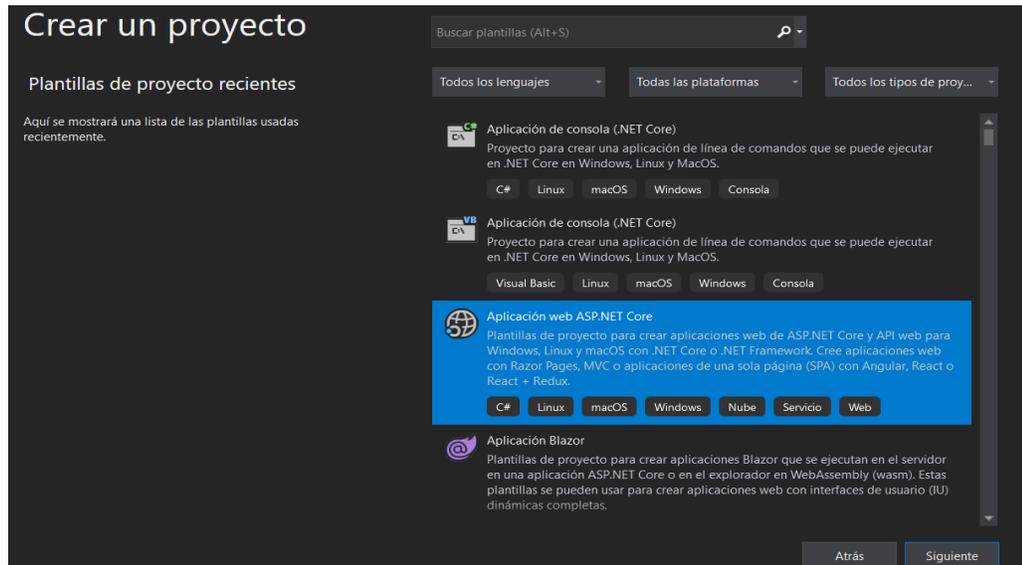


Ilustración 54 - Selección tipo de proyecto en Visual Studio

Elegimos “Aplicación web **ASP.Net Core**” y damos clic al botón “*Siguiente*”.

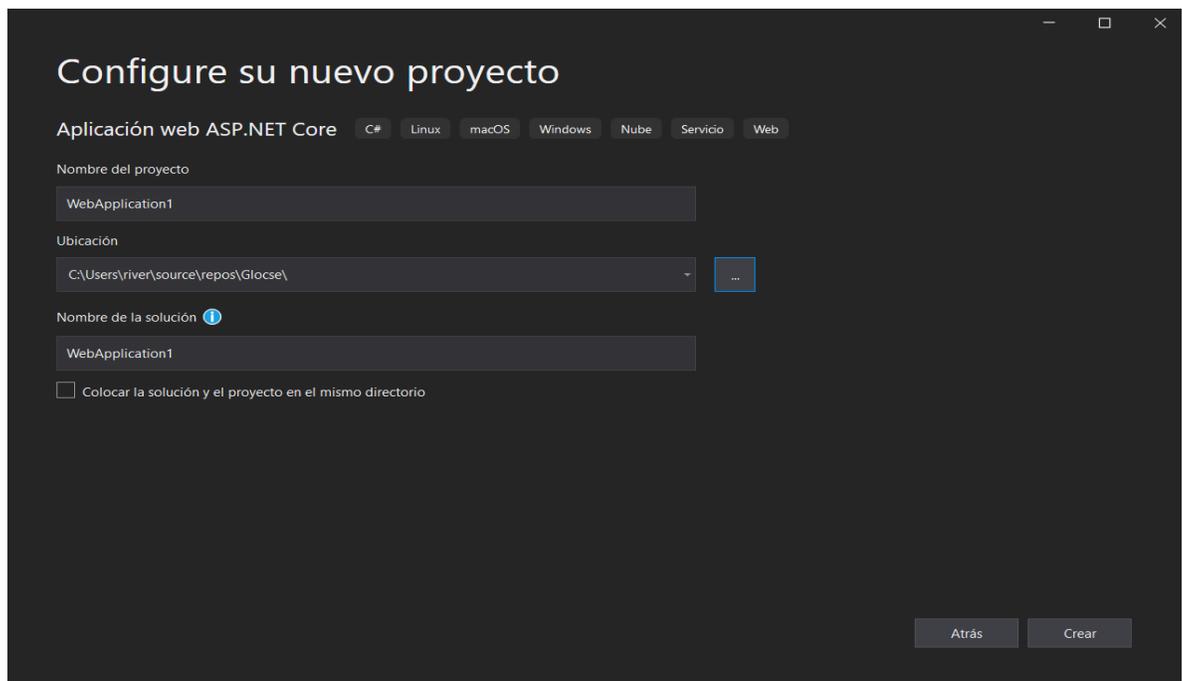


Ilustración 55 - Configuración de proyecto en Visual Studio



En ubicación escogemos la carpeta que previamente habíamos inicializado como nuestro repositorio.

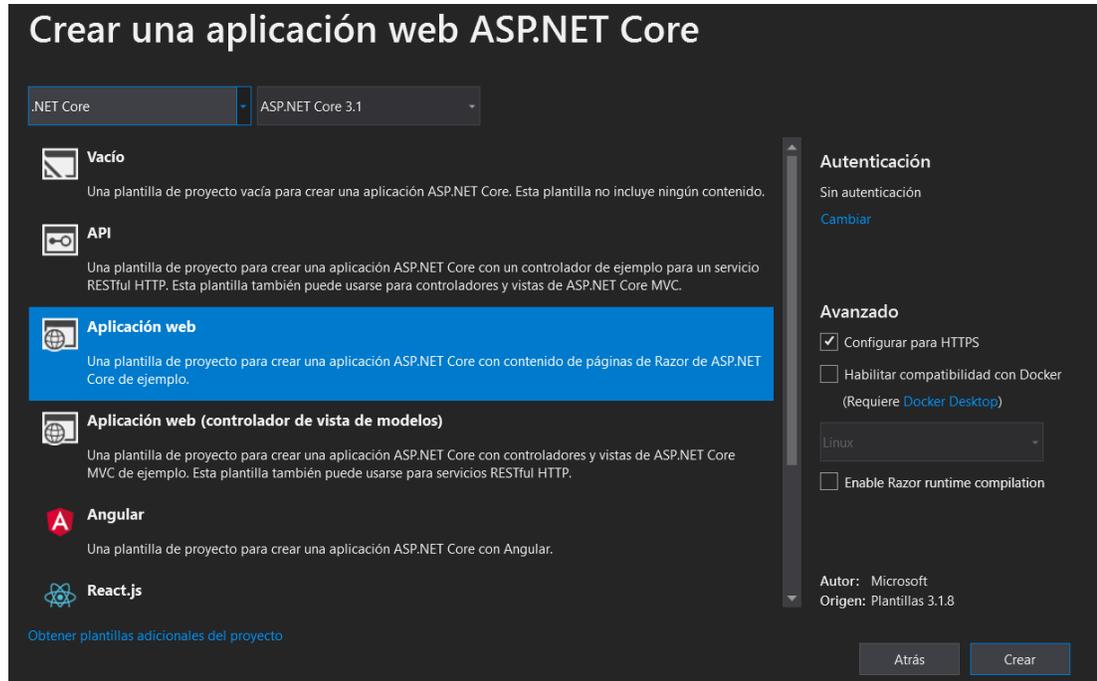


Ilustración 56 - Selección de Aplicación Web

Seleccionamos *Aplicación Web*.

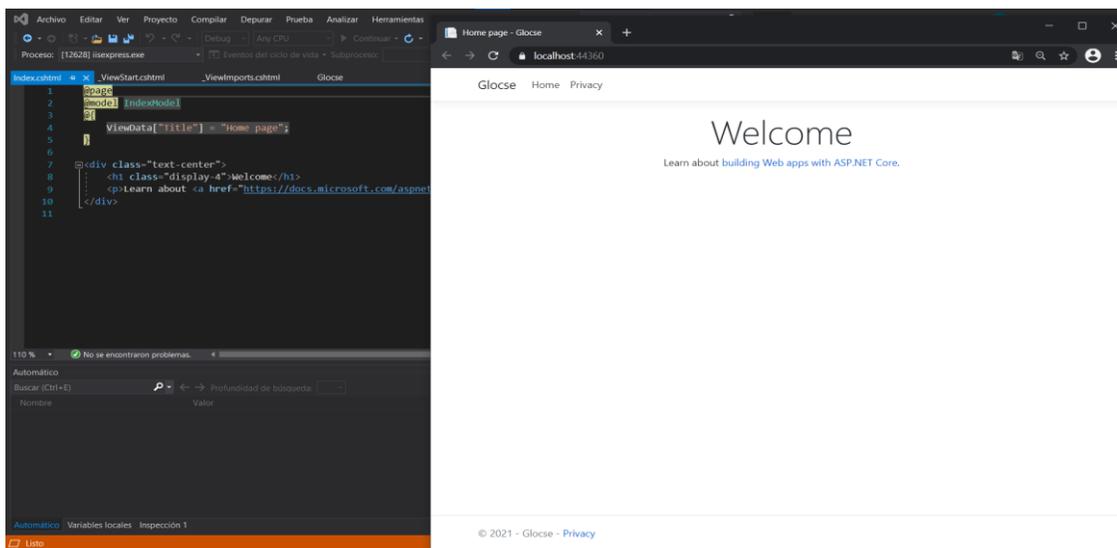


Ilustración 57 - Proyecto creado satisfactoriamente



Con esto ya hemos creado el proyecto estamos listo para iniciar con el desarrollo.

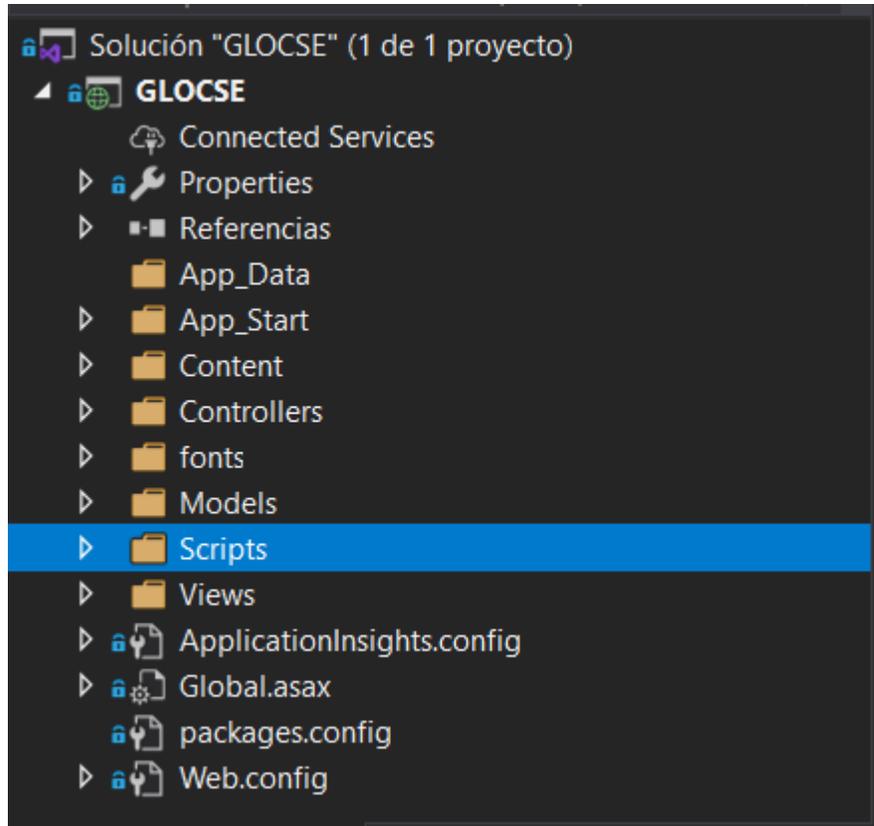


Ilustración 58 - Vista de los principales componentes de un proyecto ASP.net MVC

VI.4.1.2 Descripción de los principales componentes de un proyecto ASP.net MVC

- **AppStart:**

La carpeta App_Start contiene los archivos de clases que se ejecutarán cuando se inicie la aplicación.

- **Content:**

Contiene archivos estáticos como hojas de estilo (css, sass, less), imágenes y archivos de iconos.



- **Controllers:**

Controladores maneja las peticiones de los usuarios y devuelve una respuesta.

- **Script:**

La carpeta Scripts contiene archivos JavaScript para la aplicación.

- **Views:**

Contiene los diferentes métodos de acción, por lo que la carpeta views contiene una carpeta separada para cada controlador.

- **Fonts:**

Contiene las fuentes utilizadas en el sistema.

VI.4.2 Base de datos

La base de datos empleada para el almacenamiento de la información fue Microsoft SQL Server Developer en su versión 2019.

VI.4.3 Seguridad

Para protección y seguridad de la aplicación se utilizaron Middlewares que verifican que el usuario este autenticado en cada petición realizada al servidor de esta forma nos aseguramos de que no se realicen solicitudes de datos fraudulentas, se implementó protección CSRF⁹, El cual es un método por el cual un usuario malintencionado intenta hacer que sus usuarios, sin saberlo, envíen datos que no quieren enviar. Afortunadamente, los ataques CSRF se pueden prevenir añadiendo un token CSRF a los formularios.

⁹ CSRF: Cross-site request forgery

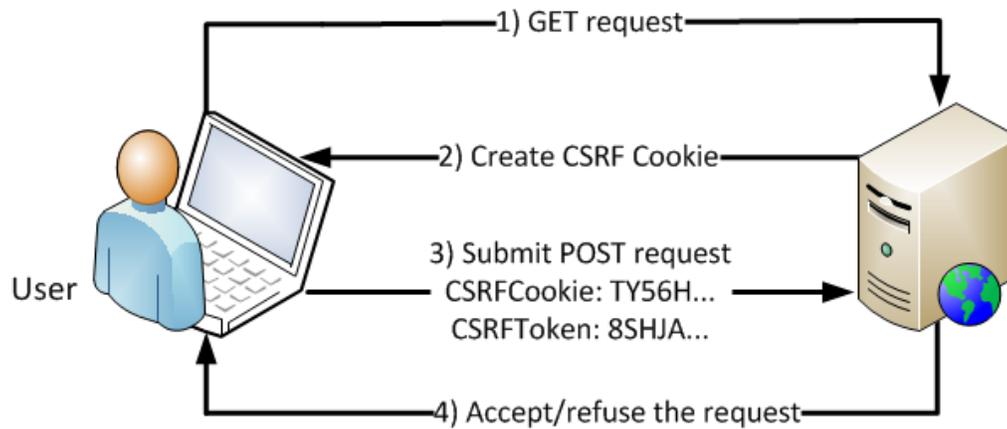


Ilustración 59 - Protección CSRF implementada

VI.4.3.1 Seguridad para entidades y control de excepciones

El almacenamiento de los datos de la cadena de conexión y la configuración de acceso al servidor de base datos se encuentran protegidos en el archivo *web.config* del proyecto web api.

```
<system.web>
<connectionStrings>
  <add name="dbconnection" connectionString="Data Source=glocse_db;Integrated Security=true;Initial Catalog=glocse" Password="*****" providerName="System.Data.SqlClient" />
</connectionStrings>
</system.web>
```

Ilustración 60 - Cadena de conexión a Base de Datos

Las acciones de los controladores que requieran uso de la base de datos se encapsularon en bloques try-catch para poder manejar las excepciones del sistema.

```
0 referencias
public ActionResult Index(int id_proyecto)
{
    var bd = new bd();
    ViewBag.Titulo = "GLOCSE - Subestación";
    try
    {
        ViewBag.proyecto = bd.obtenerInformacionProyecto(id_proyecto);
    }
    catch(Exception ex)
    {
        return BadRequest(ex.InnerException.Message);
    }
    return View();
}
```

Ilustración 61 - Manejo de excepciones en el sistema



VI.5 Pruebas

VI.5.1 Evolución del sistema

En esta etapa se procederá con la elaboración de pruebas de cada uno de los módulos realizados, se realizaron pruebas de funcionalidad y pruebas unitarias con el objetivo de validar que los cálculos de las proyecciones sean correctos.

VI.5.1.1 Modelo en espiral

El modelo de desarrollo de software seleccionado fue el de espiral estudiado en la asignatura de ingeniería de software I, se escogió este modelo de desarrollo debido a que hace pleno énfasis en la mitigación de riesgos, el modelo en espiral demanda una consideración continua de los riesgos técnicos en todas las fases del proyecto y si se emplea pertinentemente debe reducir los riesgos antes de que se conviertan en problemas, dada la naturaleza extracurricular de nuestro sistema fue la metodología perfecta para el desarrollo. Se desarrollaron los siguientes prototipos:

VI.5.1.1.1 Prototipo 1

- Se creó el esquema de navegación inicial de la aplicación.
- Se integro Google Maps.
- Se implemento un menú lateral para configurar el mapa de manera sencilla.
- Se implementó una barra de búsqueda de lugares para que el usuario pudiera elegir la ubicación de la subestación eléctrica.
- Se creó la opción de ingresar marcadores en el mapa, el usuario simplemente tiene que dar clic derecho en cualquier ubicación dentro del mapa y se ubicará el marcador en dicha zona.

VI.5.1.1.2 Prototipo 2

- Implementación de la funcionalidad de limitar mapa; se crean validaciones para que el usuario no pueda ubicar marcadores si el mapa no ha sido limitado, se implementa un bloqueo al mapa para no se desplazase fuera del rectángulo de los bordes que se han limitado.



- Se crea la opción de quitar limitación del mapa en el caso de que el usuario lo haya limitado en una zona equivocada.
- En el menú lateral se crea un interruptor llamado consumidores, al presionarlo se mostrarán en el mapa todos aquellos consumidores (Tiendas, hospitales, empresas, etc.) que se encuentren sobre la limitación establecida con el usuario, de acuerdo con su categoría se pondrán con un icono diferente para facilitar la toma de decisiones.

VI.5.1.1.3 Prototipo 3

- Se crea un contenedor informativo para consumidor este se muestra al dar clic en cada marcador, en este contenedor se mostrará información importante acerca de este consumidor tales como su categoría, ubicación geográfica, sus coordenadas físicas (X, Y) y una tabla de potencias acerca de los consumos realizados por este en las 24 horas del día.
- Se crean validaciones para que el usuario no pueda ingresar más de 10 marcadores en el mapa.
- Se implementan nuevos botones para procesar la elipse, Guardar espacios de trabajo, creación de la superficie normal de la proyección, las gráficas de consumo de los consumidores, geolocalización para ubicar el centro del mapa en la ubicación actual del usuario.
- Se creó una plantilla de carga masiva, el usuario solo tiene que descargarla del sistema e ingresar sus datos para facilitar la generación de la proyección.
- Mediante el botón de generar elipse se procesa toda la información suministrada por el usuario y se ubican en el mapa los centros eventuales generados, y se dibuja la elipse ubicando un pin en el centro para denotar la ubicación óptima para la construcción de la subestación.



VI.5.1.1.4 Prototipo 4

- Se creó el módulo de reportes en donde se muestran los datos suministrados por el usuario para crear la proyección, se crearon tablas de cargas, centros eventuales, potencias, variables de procedimiento y de proyección.
- Se creó un botón de proyecciones el cual sirve para ubicarse entre las 3 proyecciones posibles para el proceso de ubicación de la subestación, cada proyección tiene sus propios marcadores y cargas. Cuando el usuario tiene las 3 proyecciones realizadas se creó un polígono el cual es el resultado de la intersección de las 3 elipses, se ubicaron marcadores en cada centro de cada polígono para que sirva de guía visual para el usuario.
- En el menú lateral se crearon nuevas opciones para facilitar la visualización, de acuerdo con la proyección que este activa se puede escoger mediante interruptores si el usuario quiere ver los centros eventuales, los consumidores entre otros.
- Se creó un PDF en donde se expone las gráficas de los consumidores y todos los datos calculados por el sistema.
- En el apartado de proyecciones se crearon espacios de trabajo de acuerdo con la proyección que este usando el usuario cada proyección tiene sus propios espacios de trabajo con valores diferentes.
- Se creó un apartado de configuración en él se puede decidir si usar potencias aleatorias (Generadas automáticamente por el sistema) con el objetivo de realizar pruebas, colores de la elipse generada en cada proyección entre otras.



VI.5.1.1.5 Prototipo 5 y final

- Se creó la página de inicio de sesión.
- Se creó la página de creación y administración de proyectos en donde cada usuario registrado en el sistema puede manejar sus proyectos.
- Se creó la opción de exportar los datos de la proyección en Excel.
- Se creó la posibilidad de guardar los proyectos creados en el sistema.

VI.5.2 Control de versiones

Se utilizó GitHub para la creación y registro de los cambios realizado en el código a lo largo de la confección de los prototipos expuestos anteriormente.

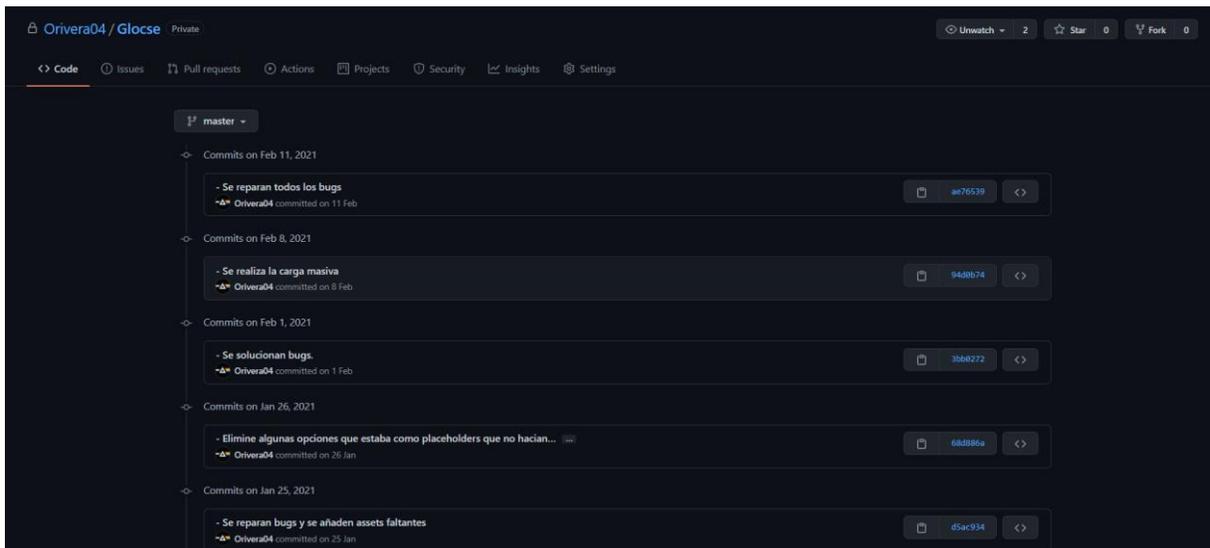


Ilustración 62 - Se muestran algunos cambios realizados en la elaboración del prototipo final

VI.5.3 Pruebas unitarias

Para la creación de pruebas unitarias se utilizó la herramienta **JUnit** a continuación se muestran las pruebas creadas y los resultados. En dónde se logra constatar que los tiempos de respuestas y validación de métodos son eficientes y satisfactorios para un desempeño óptimo del sistema.



Glocse		
<input type="checkbox"/> Hide passed tests	<input type="checkbox"/> Check for Globals	<input type="checkbox"/> No try-catch
Filter:	Go	Module: All modules
QUnit 2.10.2-pre; Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36 Edg/89.0.774.63		
15 tests completed in 17 milliseconds, with 0 failed, 0 skipped, and 0 todo. 766 assertions of 766 passed, 0 failed.		
1. Limitar mapa (28) Rerun		1 ms
2. Quitar limitación del mapa (8) Rerun		1 ms
3. Carga masiva por proyección (33) Rerun		0 ms
4. Conversión de coordenadas cartesianas a ubicación geografica (35) Rerun		1 ms
5. Geolocalización (7) Rerun		0 ms
6. Generación de elipse (168) Rerun		2 ms
7. Creación de superficie normal (105) Rerun		1 ms
8. Cambio de proyeccion (21) Rerun		1 ms
9. Intersección de proyecciones (4) Rerun		0 ms
10. Intersección de proyecciones (77) Rerun		1 ms
11. Carga y guardado de espacios de trabajo (56) Rerun		1 ms
12. Mostrar consumidores cercanos (35) Rerun		0 ms
13. Validar archivo PDF (91) Rerun		1 ms
14. Validar archivo Excel (56) Rerun		1 ms
15. Guardar Proyecto (42) Rerun		1 ms

Ilustración 63 - Pruebas Unitarias

En total se crearon 15 test que validan todo el proceso, tomando en consideración aquellos procesos claves para el funcionamiento del sistema, siendo:

- Limitar Mapa
- Quitar Limitación
- Carga masiva por proyección
- Conversión de coordenadas cartesianas a ubicación geográfica
- Geolocalización
- Generación de elipse
- Creación de superficie normal
- Cambio de proyección
- Intersección de proyecciones
- Carga y guardado de espacios de trabajo
- Mostrar consumidores cercanos
- Validar archivo PDF
- Validar archivo Excel
- Guardar Proyecto



VI.6 Caso de Prueba

Para propósitos de prueba, se ha tomado como referencia un estudio realizado en el departamento de Estelí, para la determinación de subestaciones eléctricas, proveyendo los datos a continuación.

En el sistema, se utilizarán las mismas potencias eléctricas ocupadas en el documento, y las coordenadas equivalentes más próximas en nuestro Mapa, ya que tienen referencias diferentes respecto a dimensiones de escala diferentes.

VARIANTE 9													
POTENCIA, 100KVA													
CARGAS										S			
HORAS		B	K	R	X	L	E	M	T	Σ	MAX	MEDIA	MINIMA
	1	10	35	5	51	49	30	60	43	283	60	35,38	5
	2	10	37	5	51	51	30	65	44	293	65	36,63	5
	3	11	36	15	55	51	35	65	45	313	65	39,13	11
	4	11	36	15	59	55	35	68	55	334	68	41,75	11
	5	12	38	26	58	59	37	68	52	350	68	43,75	12
	6	12	42	26	57	58	36	68	17	316	68	39,50	12
	7	15	47	38	56	57	36	72	18	339	72	42,38	15
	8	16	47	39	58	56	38	72	12	338	72	42,25	12
	9	16	13	24	59	55	42	47	14	270	59	33,75	13
	10	17	14	39	60	52	47	47	15	291	60	36,38	14
	11	18	15	39	68	25	47	39	16	267	68	33,38	15
	12	12	15	43	62	37	39	41	16	265	62	33,13	12
	13	14	15	43	64	37	41	45	15	274	64	34,25	14
	14	15	15	38	55	48	45	51	12	279	55	34,88	12
	15	16	15	57	52	48	51	40	39	318	57	39,75	15
	16	16	11	56	25	35	40	31	41	255	56	31,88	11
	17	15	11	58	37	30	31	31	45	258	58	32,25	11
	18	12	10	59	37	32	31	31	51	263	59	32,88	10
	19	12	11	60	48	29	31	29	40	260	60	32,50	11
	20	10	10	68	11	28	29	30	31	217	68	27,13	10
	21	10	11	58	68	30	30	31	45	283	68	35,38	10
	22	10	10	59	68	32	30	31	51	291	68	36,38	10
	23	8	11	60	72	29	30	29	40	279	72	34,88	8
	24	8	10	68	51	28	30	30	31	256	68	32,00	8
W		306	515	998	1282	1011	871	1121	788	6892			
S,100KVA	Maximo	18	47	68	72	59	51	72	55	350			
	Medio	12,750	21,458	41,583	53,417	42,125	36,292	46,708	32,833	287,167			
	Minimo	8	10	5	11	25	29	29	12	217			

Ilustración 64- Tabla de Cargas eléctricas de Caso de Prueba



Coordenadas			
Cargas	Categorías	Coordenadas en X (mts)	Coordenadas en Y(mts)
A	1	3140.7012	305.8245
B	3	3148.7425	2366.2595
C	3	2854.8344	3962.2545
D	3	1326.4557	3608.9138
E	2	3735.9796	4017.354
F	3	1887.0224	2391.3654
G	3	2888.8953	3127.7181
H	3	2007.8154	4659.811
I	1	3140.7012	305.8245
J	3	3148.7425	2366.2595

Tabla 24 - Coordenadas en Mapa para consumidores

Una vez que estos valores han sido procesados por los algoritmos correspondientes, se muestra según el caso de prueba una ilustración de la siguiente manera:

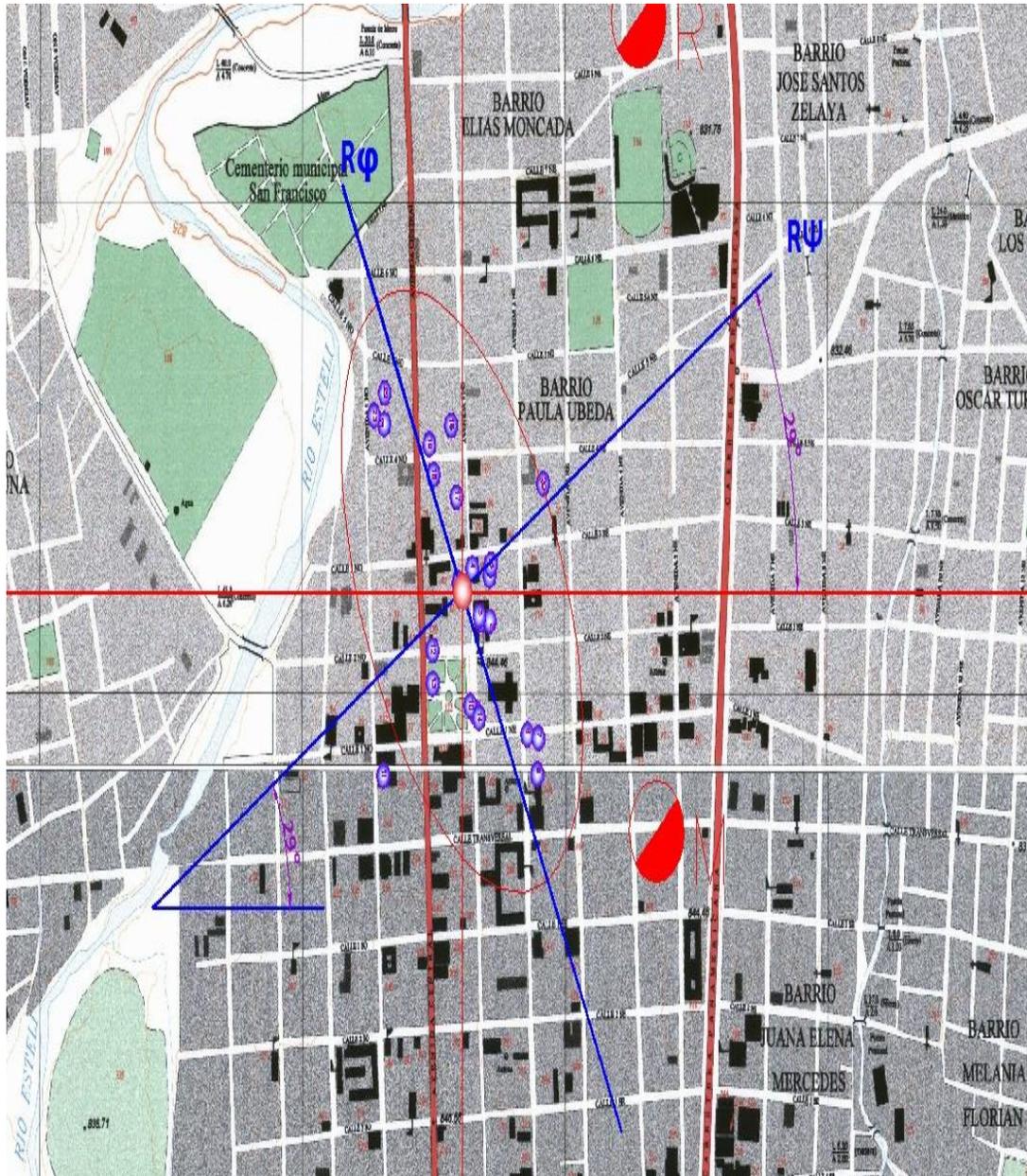


Ilustración 65 - Ubicación estimada de la elipse

En el prototipo de sistema web GLOCSE, se limita una zona próxima para definir el área a trabajar. Posterior a esto se ubican los consumidores en el Mapa, los cauels cargarán las cargas eléctricas masivamente mediante el uso de un archivo excel que comprende estos datos.

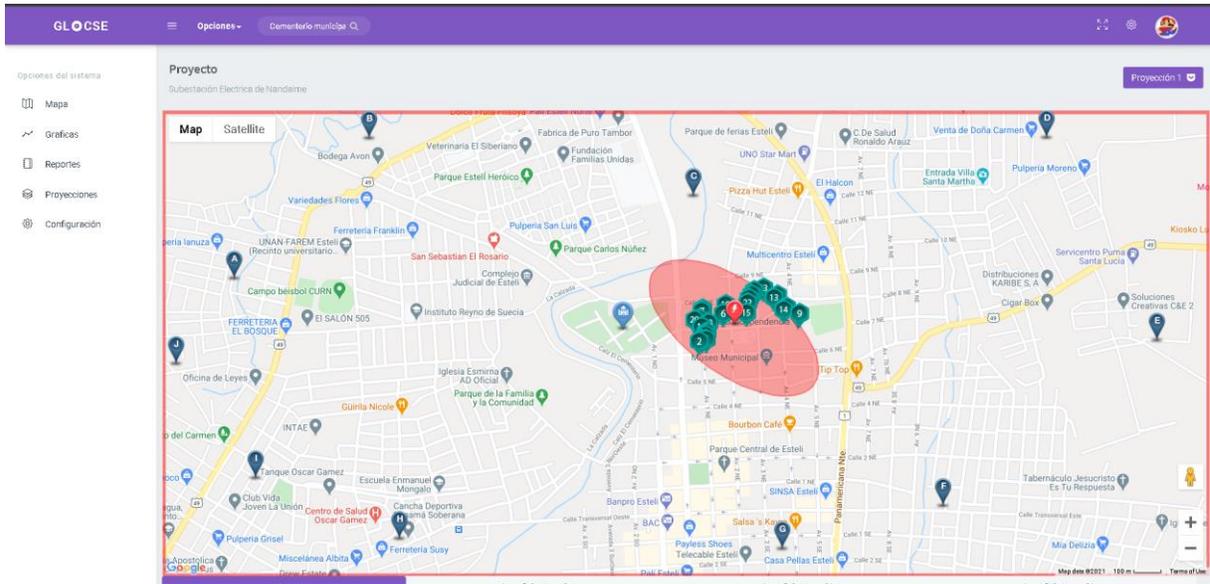


Ilustración 66 - Gráfica de Elipse con datos proporcionados

A cómo es apreciable, se logra determinar en la imagen, las coordenadas más próximas, siendo la gráfica muy acertada, respecto al caso de prueba.

VI.7 Evaluación de resultados

A modo de prueba, se realizarán evaluaciones de la manera convencional que el proceso se realiza (estas pueden ser manualmente, o usando una hoja de cálculo de Excel).

Actualmente, es usado por profesionales en hojas de cálculo en dónde ellos tienen que ingresar los datos manualmente en fórmulas que ya poseen diseñadas en Excel, lo que puede encontrarse un poco tedioso.

VI.7.1 Desempeño

La medida del desempeño es el tiempo de respuesta que se requiere hacer para una determinada tarea o procedimiento.

- Tiempo de respuesta (T): tiempo requerido para generar un reporte, una función, y diferentes funcionalidades del sistema. $T_{nuevo}, T_{anterior}$



- Razón de rendimiento (R): Expresa en qué medida el uso del nuevo sistema supera en rendimiento el método anterior.

Fórmula 1: razón de rendimiento

$$R = \frac{T_{nuevo}}{T_{anterior}} = \frac{T_{anterior}}{T_{nuevo}}$$

En donde, si el resultado es mayor a 1, entonces el sistema es mejor que el proceso antiguo. Si es igual a 1 entonces el sistema y el proceso son iguales. Y si es menor que 1 el sistema es inferior en rendimiento.

Fórmula 2: ventaja neta porcentual de rendimiento

$$n\% = (n-1) * 100\%$$

Siendo n% el porcentaje de expresión en la medida que es mejor el proceso actual el sistema.

VI.7.2 Facilidad de uso

La facilidad de uso implica la sencillez del uso de una herramienta en este caso el sistema, que solo se puede medir mediante pregunta directa al usuario en función de su percepción al momento de usarla.

- Esfuerzo: cantidad de pasos necesarios para lograr una tarea.
- Simplicidad: grado de simplificación y adecuación al diseño que facilite la interacción con el sistema.

VI.7.3 Proceso de evaluación de pruebas

El proceso de evaluación cuantitativa seleccionado fue el método de pruebas para desempeño y facilidad de uso. Los ejecutores de las pruebas fuimos los desarrolladores del sistema. Se realizaron 2 casos de pruebas.

- Prueba 1: Ejecución manual de los cálculos, sin procesos de automatización, versus el prototipo de sistema haciendo que el usuario ingrese las cargas eléctricas manualmente.



- Prueba 2: Realización de una hoja de Excel con fórmulas que recepciona los cálculos, y se ejecutan simultáneamente. Ingreso de datos de cargas manuales.

Se cronometró el tiempo y la cantidad de pasos que tomar realizar estas tareas, así como la simplicidad de uso que los usuarios perciben. Los resultados obtenidos de las pruebas se tabularán para comparar y verificar si el sistema es viable.

VI.7.4 Evaluación de desempeño

La tabla a continuación provee un resume de los resultados de evaluación de la razón de rendimiento (n) y mejora neta porcentual de rendimiento (n%):

Evaluación de desempeño				
Caso de prueba	Método convencional (ts)	Sistema GLOCSE (ts)	n	n%
Prueba 1	5,700	560	10.179	917.86%
Prueba 2	2,220	145	15.310	1431.03%
Global	7,920	705	12.744	1174.45%

Tabla 25 - Evaluación de desempeño

Los resultados prueban que el uso del sistema GLOCSE tiene mejor desempeño que el método anterior de trabajo, debido a que el tiempo requerido para completar la misma operación varía a favor del sistema GLOCSE, además el indicador de mejora del rendimiento global muestra un 1174.45% mejora neta porcentual de rendimiento, reduciendo significativamente el tiempo el proceso de ejecución de las operaciones.

VI.7.5 Evaluación de facilidad de uso

Para medir la facilidad de uso para los distintos usuarios fue necesario hacer un conteo de la cantidad de esfuerzo requerido (número de pasos) para realizar las pruebas asignadas y se tomó en cuenta la percepción que tuvo el usuario según el nivel de complejidad de cada caso.



Evaluación de facilidad de uso					
Caso de prueba	Esfuerzos (Cantidad de pasos)		Mejora de facilidad	Simplicidad	
	Método anterior	Sistema		Método anterior	Sistema
Prueba 1	32	6	81.25%	Complicado	Simple
Prueba 2	14	6	57.14%	Normal	Simple

Tabla 26 - Evaluación de facilidad de uso

Los resultados obtenidos demuestran que el sistema GLOCSE requiere menos pasos de procedimientos para completar las operaciones a diferencia del método anterior. El índice de mejora de facilidad de uso fue de 81.25 % para el primer caso, y para el segundo se obtuvo una mejora de 57.14 %. En cuanto a la percepción de la evaluación se demostró que para el caso 1 pasó de ser de un proceso complejo a uno normal para el caso 2 se pasó de proceso normal a uno simple.

VI.8 Implementación y despliegue

Una vez concluida la etapa de pruebas y de evaluación de los resultados estamos listos para implementar el sistema en un servidor, procederemos a realizar la implementación en el servicio de computación en la nube **Microsoft Azure**, el proceso es muy sencillo de realizar basta con que tengamos una cuenta registrada y activada la facturación.

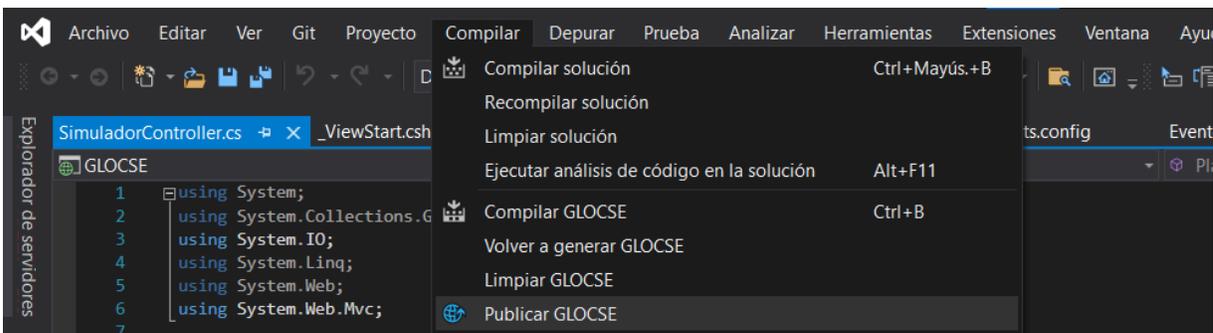


Ilustración 67 - Proceso para publicar una aplicación

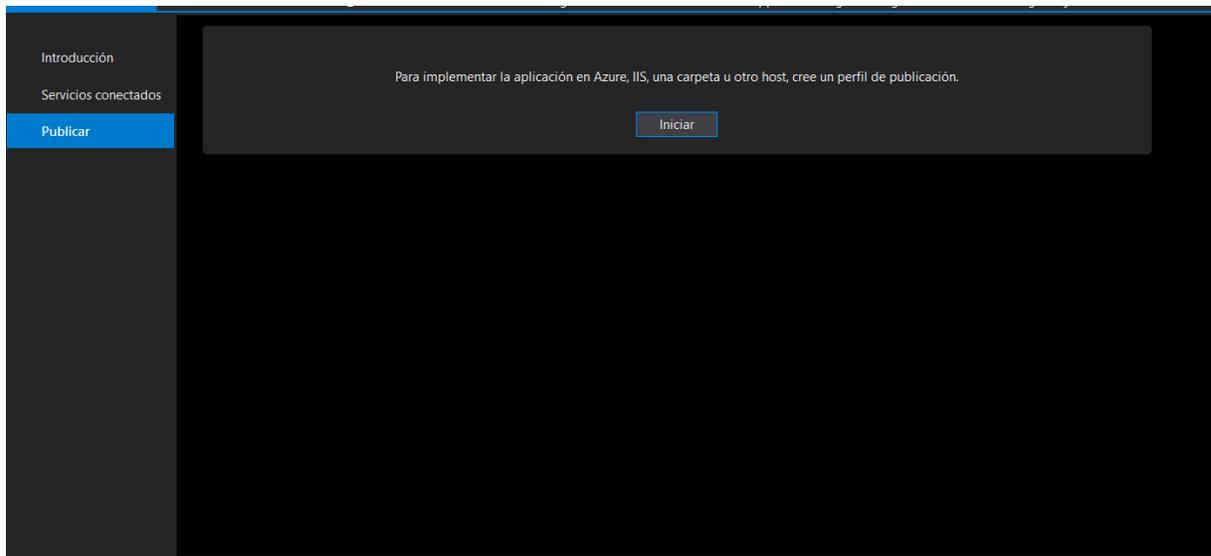


Ilustración 68 - Sección de implementación de Visual Studio

Desde Visual Studio accedemos a **Compilar -> Publicar GLOCSE** luego damos clic en el botón **Iniciar**.

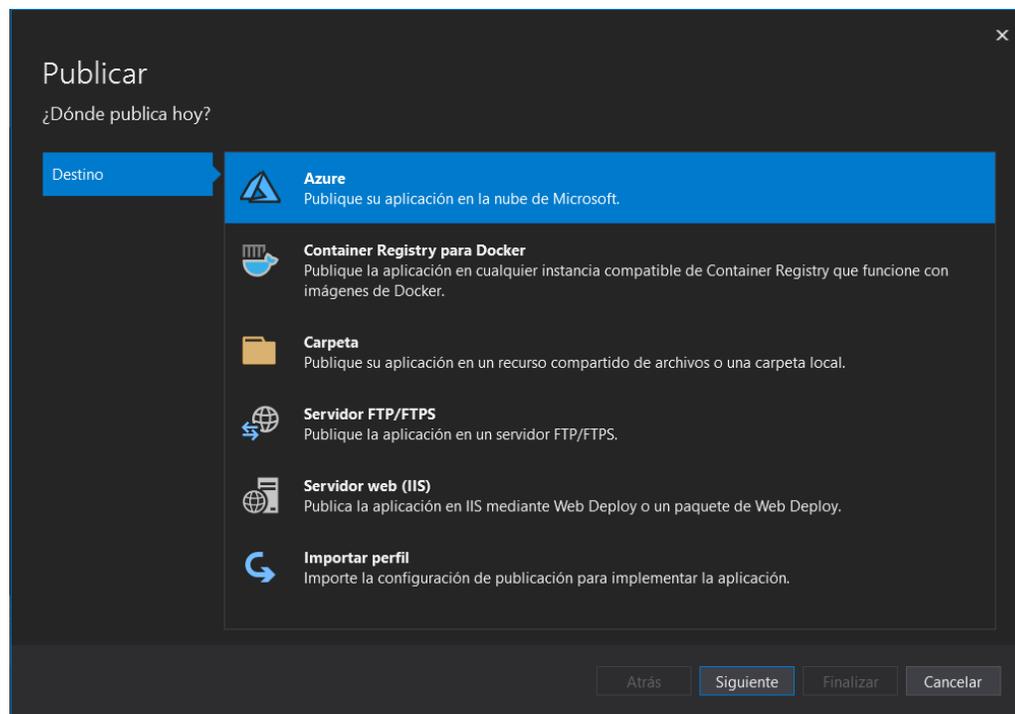


Ilustración 69 - Destinos de implementación que nos brinda Visual Studio para nuestras aplicaciones

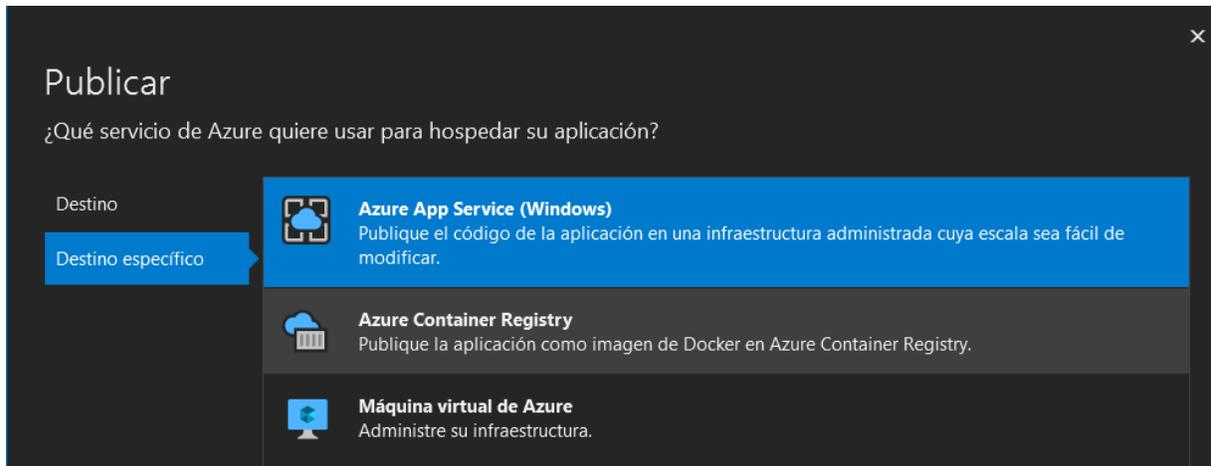


Ilustración 70 - Sub destinos de una aplicación en Azure

En el destino seleccionamos **Azure** y luego **Azure App Service**.

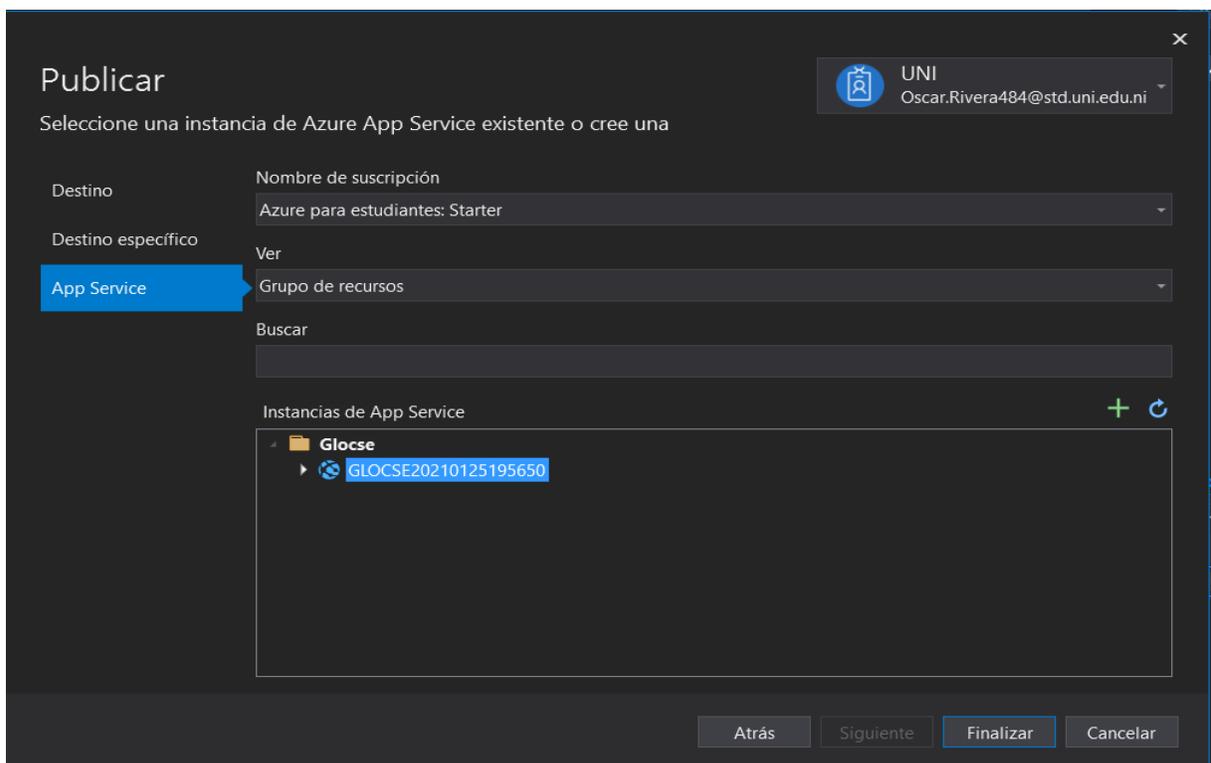


Ilustración 71 - Configuración de instancia



Ingresamos con nuestra cuenta, seleccionamos nuestra suscripción y damos clic en **Finalizar**.

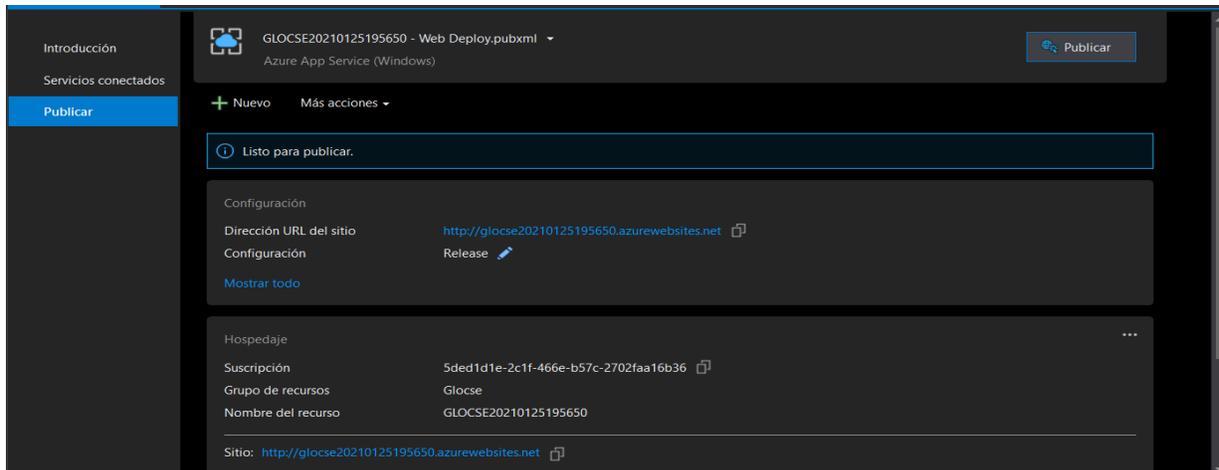


Ilustración 72 - Resumen de implementación

Damos clic en publicar y con eso habrá concluido el proceso.

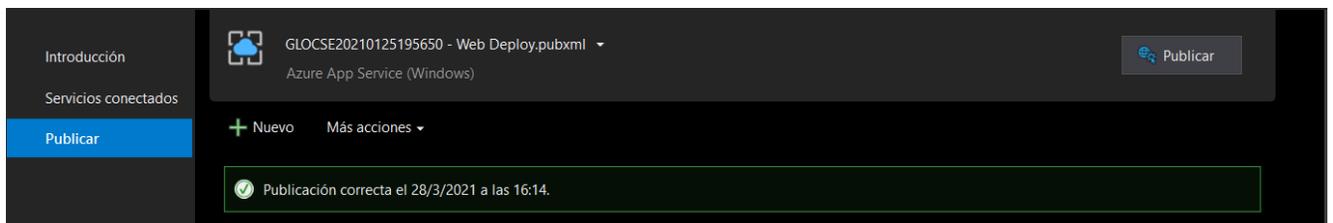


Ilustración 73 - Mensaje de implementación exitosa

Sistema web para determinar y visualizar la geolocalización de subestaciones eléctricas en Nicaragua.

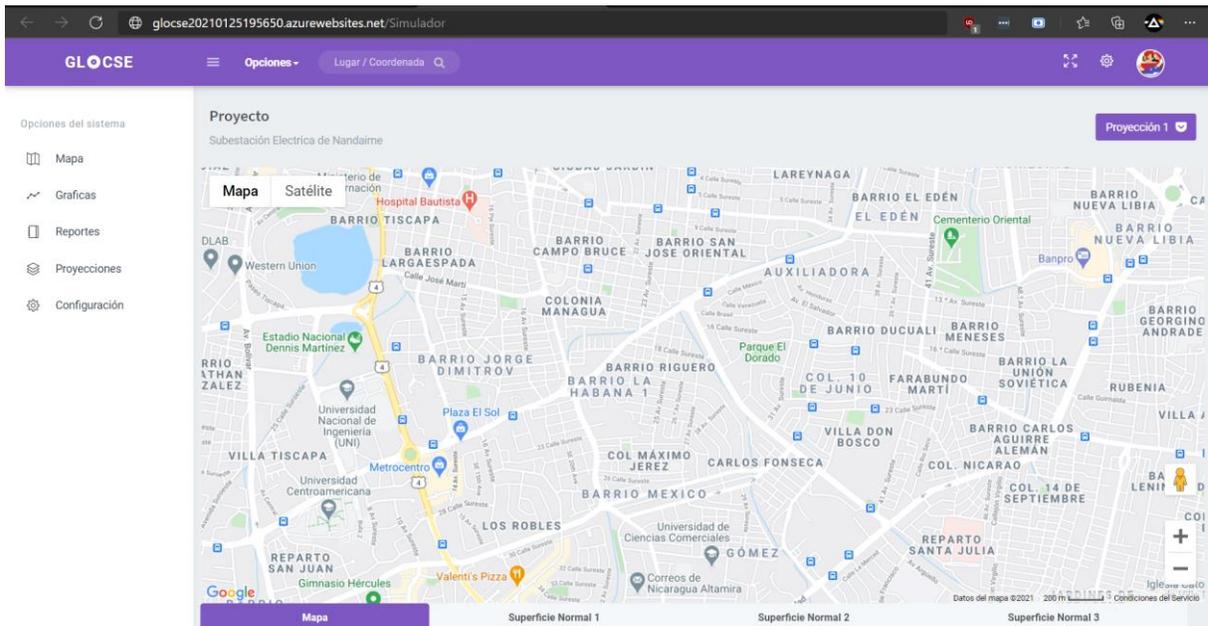


Ilustración 74 - Sistema desplegado en Azure

Al terminar automáticamente Visual Studio nos redireccionara al URL¹⁰ del sistema.

¹⁰ URL: Uniform Resource Identifier o, identificador de recurso único.



VIII. CONCLUSIONES

La realización del presente trabajo monográfico tuvo como objetivo desarrollar un prototipo de sistema web que permita determinar la ubicación geográfica óptima de subestaciones eléctricas aplicando un modelo matemático soportado por funciones de Liapunov.

Como solución a la necesidad expuesta en este documento en forma precisa y metodológica, se planteó el objetivo general y los objetivos específicos, los cuales se cumplieron en su totalidad, permitiendo llegar a las siguientes conclusiones:

- Mediante las técnicas de recolección de datos (entrevistas no estructuradas y observación directa) fue posible analizar cada una de las fases del modelo matemático basado en funciones de Liapunov, el cual provee las pautas y criterios para la graficación de las elipses en una ubicación geográfica establecida por el usuario.
- La comprensión de dicho modelo matemático permitió la creación de un algoritmo computacional mediante librerías de funciones matemáticas que permitieron la codificación del modelo matemático en este prototipo de sistema.
- La implementación de la metodología UWE facilitó en gran manera el desarrollo de este sistema mediante el diagramado en UML, permitiendo seccionar a detalle cada uno de los módulos y de igual manera la recopilación de requerimientos funcionales y no funcionales.
- La codificación del sistema se realizó mediante ASP.Net Core MVC, JavaScript, JQuery, Bootstrap, Google Maps Y Fabric.JS. Dichas tecnologías permitieron crear un ambiente amigable para el usuario, siendo este rápido y sencillo de navegar.

En conclusión, mediante las pruebas realizadas se observó que el tiempo es un factor que se ve impactado positivamente a gran escala. Dicho lo anterior, se demuestra el cumplimiento de los objetivos propuestos, habiendo resuelto las diversas situaciones que se presentaron a lo largo del proyecto; se han consolidado y adquirido conocimientos, técnicas y herramientas que aportan a la formación de los autores como ingenieros.



IX. RECOMENDACIONES

Al momento de usar este prototipo de sistema web, se recomienda leer el manual de usuario para una mejor comprensión de las funciones en caso de que existan dudas en base a la secuencia del funcionamiento del sistema.

Para futuros proyectos de desarrollo de sistemas web recomendamos el uso de la metodología UWE en base a la experiencia de este proyecto. Aunque puede ser un poco demandante en la cantidad de diagramas que deben construirse como parte del análisis y diseño, esto contribuye a la comprensión de la funcionalidad del sistema y su posterior mantenimiento.

Así también, al ser una metodología propiamente diseñada para el desarrollo de sistemas web, ésta establece una serie de elementos relevantes para este tipo de sistemas que son de utilidad para guiar el proceso de desarrollo de software.

En el proceso del análisis del sistema web se utilizó la herramienta de modelación de diagramas Visual Paradigm, la cual resultó muy útil en la creación, muy intuitiva para usar y posee mucha variedad de diagramas, y documentación que sustenta su uso.

Para comprensión de modelos matemáticos complejos, se recomienda dividir en partes los algoritmos, para una comprensión minimalista y posteriormente su inclusión en el lenguaje de programación deseado.

Este sistema web desarrollado, tiene un gran potencial de escalabilidad a futuro, por lo cual se propone la creación de una aplicación móvil para Android y iOS que pueda realizar la misma funcionalidad que la aplicación web actualmente realiza.

Adicionalmente se sugiere la creación de un módulo alternativo para cargar mapas estructurales de empresas, y calcular el punto óptimo de ubicación de generadores de energía para las mismas, proveyendo así una distribución energética que reduzca costos de construcción y mantenimiento a dichos generadores.



X. REFERENCIA BIBLIOGRÁFICAS

- [1] ENATREL, «Nicaragua alcanza el 97.69% de cobertura eléctrica,» Junio 2020. [En línea]. Available: <http://www.enatrel.gob.ni/nicaragua-alcanza-el-97-69-de-cobertura-electrica/#:~:text=Gracias%20a%20la%20labor%20del,al%20concluir%20este%20a%C3%B1o%202020>.
- [2] Ministerio de Energía y Minas, «Plan de Inversión – Nicaragua (PINIC) del Programa SREP Nicaragua,» Gobierno de Reconciliación y Unidad Nacional, Managua, 2015.
- [3] F. Núñez, Centrales de generación y subestaciones electricas, Republica Dominicana: Universidad de APEC, 2015.
- [4] A. R. E. Feodorov, Suministro Eléctrico de Empresas Industriales, Ciudad de La Habana: Pueblo y Educación, 1980.
- [5] R. M. R. M. S. y. Y. K. Walpole, Probabilidad y estadística para ingeniería y ciencias, México: PEARSON EDUCACIÓN, 2012, p. 816.
- [6] J. Cánovas, «TEORIA CUALITATIVA DE LAS ECUACIONES DIFERENCIALES,» Octubre 2014. [En línea]. Available: <http://www.dmae.upct.es/~jose/mastering/ecuadif.pdf>.
- [7] J. Alfaro, «Ley de Gauss,» 18 Marzo 2010. [En línea]. Available: <http://www.fis.puc.cl/~jalfaro/FIZ0321/clases/gauss.pdf>.
- [8] D. M. d. C. Gómez Fuentes, «Notas del curso de base de datos.,» Universidad Autonoma Metropolitana, Mexico D.F, 2013.
- [9] M. Peñafiel, «Definición de aplicación WEB,» Escuela Superior Politecnica de Chimborazo, Chimborazo, Ecuador., 2008.
- [10] Addison-Wesley, Computacion Distribuida Fundamentos y Aplicaciones., Madrid(España): PERSON EDUCACION, S.A., 2004.
- [11] A. Marquez, «Cliente Servidor,» Universidad de las Americas Puebla, Puebla, Mexico, 2011.
- [12] D. Cantone, Implementacion Y Debugging, España: Creative Andina Corp., 2006.
- [13] F. Berzal, «El lenguaje de modelo unificado,» Madrid, España, 2004.
- [14] L. Maximilians, «UWE - UML - based Web Engineering,» 10 Agosto 2016. [En línea]. Available: <http://uwe.pst.ifi.lmu.de/aboutUwe.html>.



- [15] N. Koch, A. Kraus y R. Hennicker, *The Authoring Process of the UML-based Web Engineering Approach*, München, Germany, 2001.
- [16] B. Camacho, *El pequeño libro de ASP.NET Core*, 2019.
- [17] J. D. Gauchat, *El gran libro de HTML5, CSS3 y Javascript*, Barcelona: MARCOMBO, S.A., 2012.
- [18] D. Write Less, «jQuery the future,» de *JQuery the future*, California, Angel City Press, 2016, pp. 10-11.
- [19] L. V. Iancker, *JQuery: El framework de JavaScript de la web 2.0*, 2015.
- [20] Efron, Bradley; Tibshirani, Robert J. ;, «An Introduction to the Bootstrap,» de *An Introduction to the Bootstrap*, New York, CHAPMAN y HALL/CRC, 2014, pp. 12-15.
- [21] S. Laurent, «Google Maps,» de *Google Maps*, First Edition, 2015, p. 33.
- [22] S. K. A. B. Juriy Zaytsev, «fabricjs,» [En línea]. Available: <http://fabricjs.com/fabric-intro-part-1>.
- [23] Universidad Nacional de Ingeniería, «NORMATIVA PARA LOS TRABAJOS MONOGRAFICOS,» Managua, 2001.
- [24] P. Kimmel, *Manual de UML*, Mexico: McGraw-Hill Interamericana Editores S.A. de C.V., 2006.
- [25] J. Schmuller, *Aprendiendo UML en 24 horas*, Pearson Educación, 2001.



XI. GLOSARIO DE TÉRMINOS

SE: Subestaciones eléctricas.

CENTRO DE CARGA: Energía consumida por consumidor en un tiempo delimitado de una hora.

CSS: Hojas de Estilo en Cascada, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado

CONSUMIDORES: Hace referencia a proveedores de energía (transformadores) que se encuentren en hospitales, parques, residenciales, entre otros.

GLOCSE: Geolocalización de Subestaciones Eléctricas.

MVC: Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: Modelo, Vista, Controlador.

PC: Personal Computer, mejor conocido como computadora personal.

PUNTO DE LOCALIZACIÓN: Ubicación en donde será construida la subestación eléctrica.

SIN: Sistema interconectado Nacional.

SNT: Sistema nacional de transmisión.

UML: Unified Modeling Language (Lenguaje unificado de modelado)

UWE: UML – BASED WEB ENGINEERING (Lenguaje unificado de modelado – Basado en ingeniería Web)



XII. ANEXOS

Se presenta constancia proveída por Ing. Adilson Gonzalez que dio seguimiento con rol de tutor al proyecto GLOCSE en su versión preliminar en evento COMPDES en el año 2019, en el país de El Salvador. Complementariamente se adjuntan los certificados de participación de ambos bachilleres obtenidos en dicho evento.

CONSTANCIA

Por este medio, hago constar que los estudiantes:

- BR. Oscar Danilo Rivera Moreira con código de estudiante 2015-0578U, y
- BR. Yader David Castillo Hurtado con código de estudiante 2015-0012U,

Fueron guiados por mi persona en el proyecto denominado *“Sistema web para determinar y visualizar la localización de subestaciones eléctricas”*, bajo la asesoría MSc. Víctor Hernández, el cual luego de su aprobación presentaron dicho trabajo en el *XII Congreso Iberoamericano de Computación para el Desarrollo COMPDES-2019* realizada en la Universidad de El Salvador del 22 al 26 de Julio de 2019, un espacio para exponer temas de Investigación Científica en la rama de la Ingeniería y Tecnología a nivel Universitario en Iberoamérica.

Doy fe de lo anterior, a solicitud de la parte interesada, a los 24 días del mes de mayo de 2021.

MSc. Adilson Isaac González
Docente
Departamento de Lenguaje y Simulación
Ingeniería en Computación FEC-UNI
Contacto: adilson.gonzalez@fec.uni.edu.ni

Ilustración 76 - Constancia de participación en evento COMPDES



Ilustración 77 - Certificado de participación - Oscar Rivera



Ilustración 78 - Certificado de participación - Yader Castillo