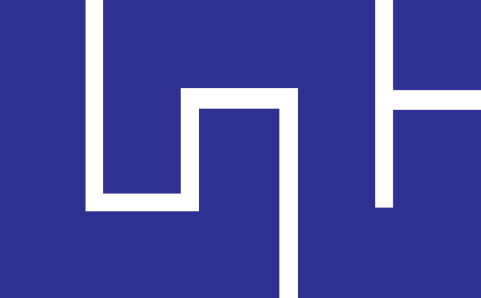
**UNIVERSIDAD NACIONAL DE INGENIERÍA**



**Facultad de Electrotecnia y Computación**

**(UNI-RUSB)**

**PROTOCOLO 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.

Managua, agosto de 2020

**Índice de Contenido**

[**I.** **INTRODUCCIÓN** 1](#_Toc49008954)

[**II.** **ANTECEDENTES** 2](#_Toc49008955)

[**III.** **JUSTIFICACIÓN** 3](#_Toc49008956)

[**IV.** **OBJETIVOS** 4](#_Toc49008957)

[**IV.1 Objetivo general** 4](#_Toc49008958)

[**IV.2 Objetivos específicos** 4](#_Toc49008959)

[**V.** **SUBESTACIONES ELÉCTRICAS** 5](#_Toc49008960)

[**V.1 Definición** 5](#_Toc49008961)

[**V.2 Localización de subestaciones eléctricas** 5](#_Toc49008962)

[**V.3 Sistema Informático** 15](#_Toc49008963)

[**V.3.1 Definición** 15](#_Toc49008964)

[**V.3.2 Base de datos** 15](#_Toc49008965)

[**V.3.3 Modelo Relacional** 16](#_Toc49008966)

[**V.3.4 Aplicación Web** 17](#_Toc49008967)

[**V.3.5 Modelo Cliente Servidor** 17](#_Toc49008968)

[**V.4 Metodología del desarrollo de Software** 19](#_Toc49008969)

[**V.4.1 Ciclo de vida** 19](#_Toc49008970)

[**V.4.2 UML** 21](#_Toc49008971)

[**V.4.3 Metodología UWE** 21](#_Toc49008972)

[**V.4.4 Arquitectura MVC** 22](#_Toc49008973)

[**V.5 Herramientas de desarrollo** 22](#_Toc49008974)

[**V.5.1 HTML5** 22](#_Toc49008975)

[**V.5.2 CSS** 23](#_Toc49008976)

[**V.5.3 JavaScript** 23](#_Toc49008977)

[**V.5.4 Jquery** 24](#_Toc49008978)

[**V.5.5 Bootstrap** 24](#_Toc49008979)

[**V.5.6 Fabric JS** 25](#_Toc49008980)

[**VI.** **METODOLOGÍA** 25](#_Toc49008981)

[**VII.** **CRONOGRAMA** 27](#_Toc49008982)

[**VIII.** **REFERENCIA BIBLIOGRÁFICAS** 28](#_Toc49008983)

[**IX.** **GLOSARIO DE TÉRMINOS** 31](#_Toc49008984)

**Índice de ilustraciones**

[Ilustración 1: Superficie de distribución normal 8](#_Toc43583756)

[Ilustración 2: Proyección del seccionamiento transversal de la superficie curva de distribución normal. 8](#_Toc43583757)

[Ilustración 3: Esquema del modelo relacional. 16](file:///C:\Users\river\OneDrive\Desktop\Protocolo%20-%20GLOCSE%20-%20V7-%20Yader%20-%20Oscar.docx#_Toc43583758)

[Ilustración 4: Puesta en escena del modelo cliente/servidor. 18](file:///C:\Users\river\OneDrive\Desktop\Protocolo%20-%20GLOCSE%20-%20V7-%20Yader%20-%20Oscar.docx#_Toc43583759)

[Ilustración 5: Ciclo de vida Lineal. 20](file:///C:\Users\river\OneDrive\Desktop\Protocolo%20-%20GLOCSE%20-%20V7-%20Yader%20-%20Oscar.docx#_Toc43583760)

[Ilustración 6: Ciclo de vida en V. 20](file:///C:\Users\river\OneDrive\Desktop\Protocolo%20-%20GLOCSE%20-%20V7-%20Yader%20-%20Oscar.docx#_Toc43583761)

[Ilustración 7: Etapas de la metodología UWE. 21](file:///C:\Users\river\OneDrive\Desktop\Protocolo%20-%20GLOCSE%20-%20V7-%20Yader%20-%20Oscar.docx#_Toc43583762)

[Ilustración 8: Definición grafica de la arquitectura MVC. 22](file:///C:\Users\river\OneDrive\Desktop\Protocolo%20-%20GLOCSE%20-%20V7-%20Yader%20-%20Oscar.docx#_Toc43583763)

# **INTRODUCCIÓN**

Este documento describe la propuesta del desarrollo de un sistema web basado en la implementación de un algoritmo matemático, para la localización de subestaciones eléctricas. Con el fin de facilitar este cálculo y los distintos aspectos que conlleva nace el proyecto GLOCSE[[1]](#footnote-1), una herramienta que permitirá calcular la localización de subestaciones eléctricas.

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 sub estación eléctrica. En esta misma se ubicarán a los consumidores[[2]](#footnote-2) y centros de cargas[[3]](#footnote-3), para posteriormente calcular los otros elementos importantes (centros eventuales de carga, el punto de localización[[4]](#footnote-4) y la zona de dispersión, entre otros).

El proyecto GLOCSE toma en cuenta el ciclo general de desarrollo de software con la metodología UWE, esto se logra notar en el apartado Metodología de Software.

Como solución proponemos desarrollar un sistema web, ya que asegura la compatibilidad con cualquier sistema operativo y teléfono móvil. Instamos al lector a atestiguar todo el modelo matemático existente detrás del algoritmo implementado, la facilidad y rapidez de cálculo que proveerá esta herramienta.

# **ANTECEDENTES**

Nicaragua es el país de la región centroamericana que posee la tasa generación de electricidad más baja del istmo, a su vez es el país que posee el porcentaje más bajo de acceso a la electricidad de toda América Central (Naciones Unidas, 2007), el sistema eléctrico de Nicaragua denominado SIN cubre el 90% del territorio nacional, siendo el restante 10% cubierto por sistemas de generación aislados (Mechanism & Mechanism Clean Development, 2014).

En lo que respecta a la distribución el 100% de la transmisión se encuentra gestionada por ENATREL, que también está a cargo del suministro del sistema, el SNT cuenta actualmente con 91 subestaciones eléctricas, 73 estatales y 18 privadas, estas a su vez se dividen en subestaciones de transmisión y de distribución (Enatrel, 2016).

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.

El principal factor a tomar en cuenta es la potencia eléctrica consumida, posterior 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 a la gran cantidad 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.

# **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. Sería 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 más óptimo en base a las necesidades de consumo de dicha zona y la empresa.

Por otra parte, no debemos de 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.

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 ocupará el Framework **ASP.Net Core MVC** ocupando ciertas librerías tales como **Google Maps API**, **Chart JS, Fabric.JS** y **SQL Server.**

# **OBJETIVOS**

## **IV.1 Objetivo general**

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

## **IV.2 Objetivos específicos**

1. Realizar un análisis de los modelos matemáticos necesarios para el proceso de ubicación de subestaciones eléctricas con el fin de diseñar un algoritmo computacional.
2. Diseñar el sistema mediante herramientas de modelado y diagramas de UML[[5]](#footnote-5), usando la metodología UWE para el desarrollo del sistema web.
3. Codificar la aplicación empleando una arquitectura MVC[[6]](#footnote-6), mediante la tecnología ASP.Net Core MVC y con apoyo de tecnologías de clientes tales como JavaScript, JQuery, Bootstrap, Google Maps Y Fabric.JS.
4. Implementar el sistema web y realizar pruebas que retroalimenten el funcionamiento del sistema propuesto.

# **SUBESTACIONES ELÉCTRICAS**

## **V.1 Definición**

“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.” (Núñez, 2015)

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 técnicos-económicos del sistema de suministro resultan óptimos y, consecuentemente garantizan el mínimo de gastos anuales.

## **V.2 Localización de subestaciones eléctricas**

1. **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.

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. (Feodorov, 1980)

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.

Las coordenadas (,) de los centros eventuales de carga constituyen los valores numéricos que adoptan las variables aleatorias , e . Éstas, a su vez forman lo que en teoría de las probabilidades se denomina una magnitud bidimensional o sistema de dos variables aleatorias (Walpole, 2012). Un correcto análisis demuestra que, con una probabilidad confiable, las coordenadas , ocurren dentro de una elipse, que tiene como coordenadas de su centro las dadas por la esperanza matemática de la magnitud bidimensional 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:

(1)

(2)

Donde:

= potencia de cada consumidor , en la hora ( = 1, 2, …, 24);

= coordenadas de cada consumidor;

= número total de consumidores.

Como lo demuestra la práctica, el cálculo de una tercera coordenada 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.

Teniendo en cuenta el teorema de Liapunov (Cánovas, 2014) y como resultado de las investigaciones realizadas, puede afirmarse que la magnitud bidimensional constituida por las coordenadas y de los centros eventuales de carga, está subordinada a la ley normal de distribución (ley de Gauss-Laplace). (Alfaro, 2010)

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

(3)

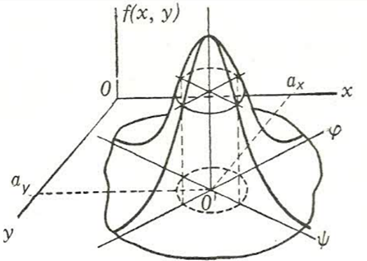
Donde:

= esperanzas matemáticas de las coordenadas eventuales;

= dispersiones de las coordenadas eventuales;

= 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 () (Fig. 1)



*Ilustración 1*: Superficie de distribución normal

A picture containing object, antenna, sky

Description generated with very high confidence

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 , se obtienen curvas semejantes a las curvas normales de distribución. El seccionamiento de la superficie de distribución, por planos paralelos al plano , proporcionará una familia de elipses, cuyo centro común se encuentra en el punto 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.” (Feodorov, 1980)

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

Esperanzas matemáticas:

; (4)

Donde:

= probabilidad empírica de ocurrencia del valor , .

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:

(5)

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

; (6)

O, lo que es lo mismo:

; (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 por , resultando las expresiones siguientes:

; (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:

(9)

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

(10)

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

(11)

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

(12)

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

Denominada coeficiente de correlación :

(13)

Una magnitud bidimensional será correlacionada siempre que ; 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 , () son independientes; si , () son dependientes.

* 1. ***Determinación de la zona de dispersión de los centros eventuales de cargas (Feodorov, 1980).***

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 , paralelo al .

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

(14)

(15)

(16)

Simbolizando , obtenemos la ecuación de una elipse, denominada elipse :

(17)

La ecuación de la elipse 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 ; que los ejes de simetría de la elipse forman ángulos con el eje , determinados por la expresión

(18)

(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 del sistema .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 y rotar los ejes coordenados un ángulo , determinando por la expresión (19).

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

(20)

Donde:

= desviaciones medio-cuadráticas de las coordenadas eventuales , en el sistema de coordenadas **.**

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

(21)

(22)

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

(23)

Donde:

(24)

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

(25)

La probabilidad de ocurrencia de los centros eventuales de carga, dentro de la elipse, está dada por:

(26)

Donde:

= área, limitada por la elipse **-.**

Resolviendo, se obtiene:

(27)

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

(28)

De donde:

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:

(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, 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. (Gómez Fuentes, 2013)

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**

“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." (Gómez Fuentes, 2013)

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.

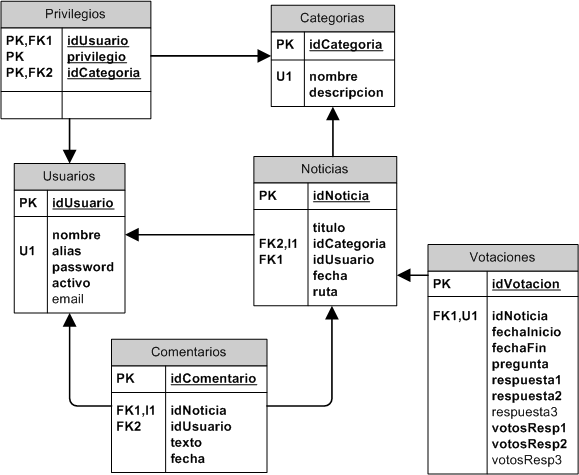


Ilustración 3: Esquema del modelo relacional.

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 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.” (Peñafiel, 2008)

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

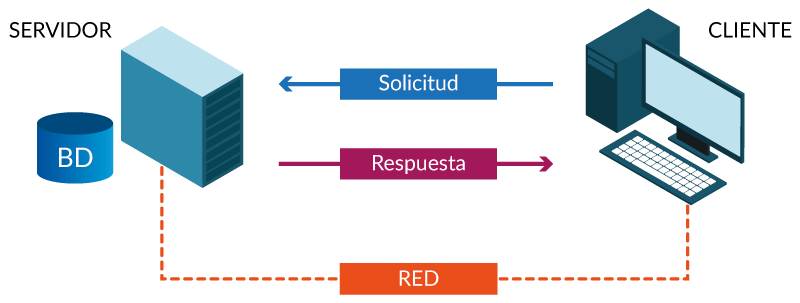
* El trabajo a distancia se realiza con mayor facilidad -Para trabajar en la aplicación web solo se necesita un computador con un buen navegador Web y conexión a internet.
* Las aplicaciones Web no necesitan conocimientos previos de informática. Con una aplicación Web tendrá total disponibilidad en cuanto 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 se refiere a un modelo de aplicaciones de red donde los procesos juegan uno de los dos diferentes papeles: Un proceso servidos, 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. (Addison-Wesley, 2004)

Desde el punto de vista funcional, se puede definir la computación Cliente/Servidor 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.” (Marquez A. , 2011)

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. Entonces, una metodología para el desarrollo de software son 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. “ (Cantone, 2006)

### **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. (Cantone, 2006)
  + **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). (Cantone, 2006)“

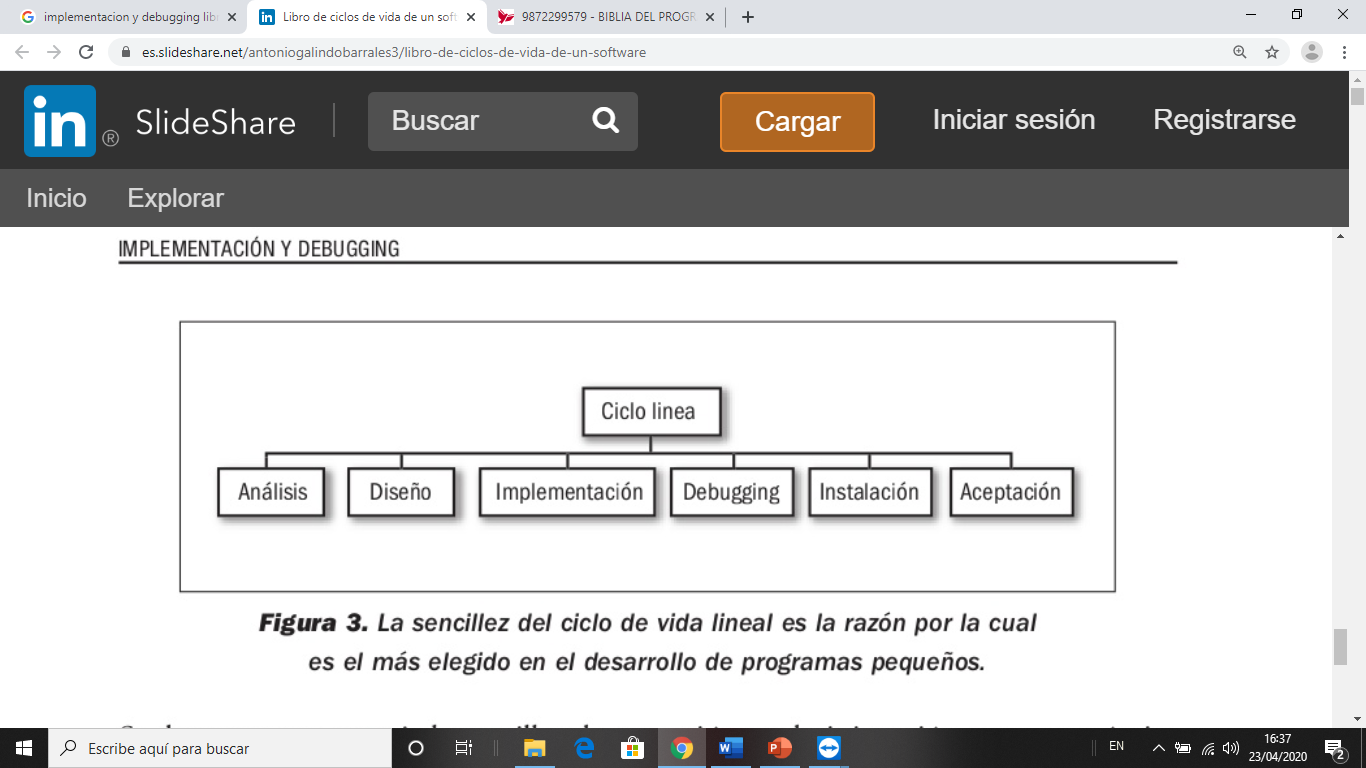


Ilustración 5: Ciclo de vida Lineal.

* + **Ciclo de vida en V**

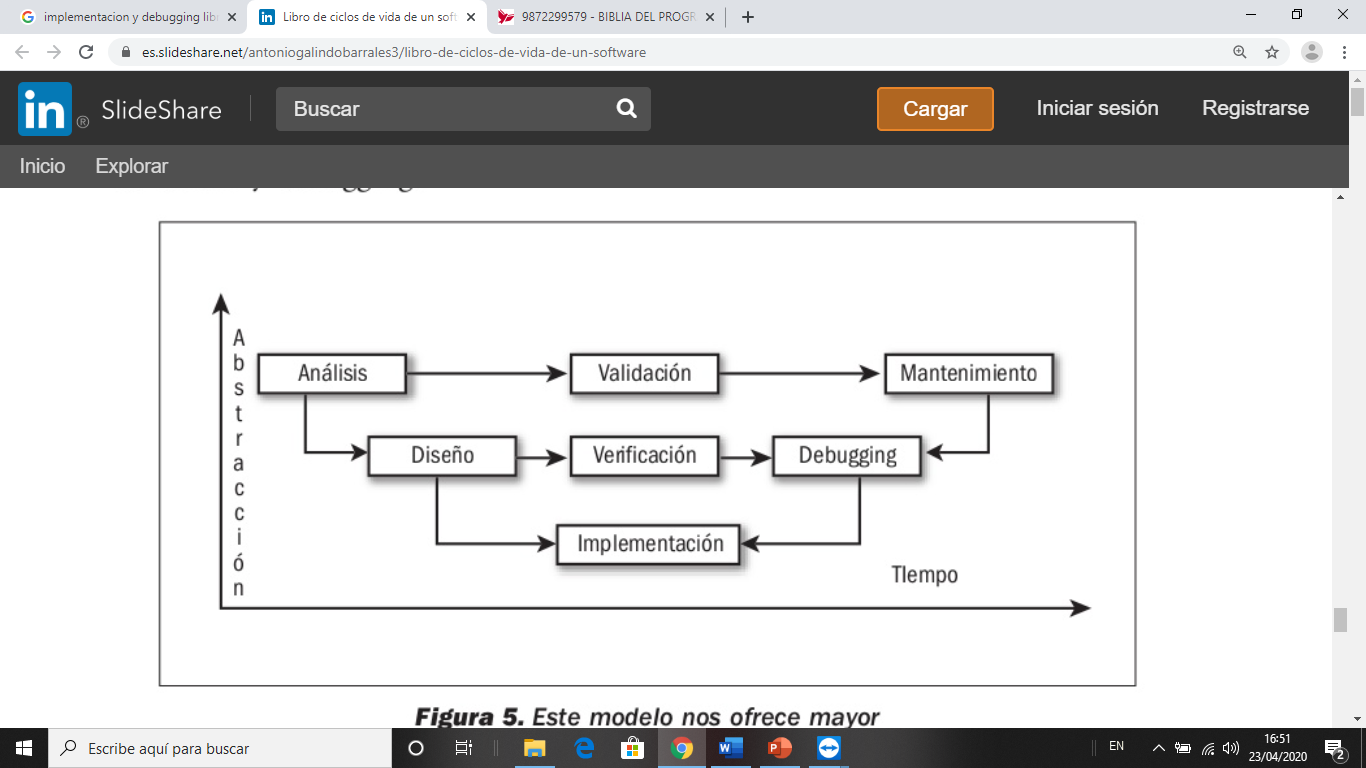
“Este ciclo fue diseñado por Alan Davis, y contiene las mismas etapas que el ciclo de vida en cascada puro. A diferencia de aquél, a éste se le agregaron dos subetapas de retroalimentación entre las etapas de análisis y mantenimiento, y entre las de diseño y depuración. “ (Cantone, 2006)

Ilustración 6: Ciclo de vida en V.

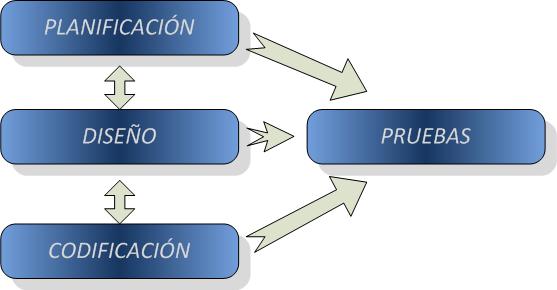
### **V.4.2 UML**

“El lenguaje UML es un estándar OMG[[7]](#footnote-7) diseñado para visualizar, especificar, construir y documentar software orientado a objetos.” (Berzal, 2004)

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. 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](http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado). ¡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”. (Maximilians, 2016)



**Ilustración 7: 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.

### **V.4.4 Arquitectura MVC**

“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.” (Camacho, 2019)

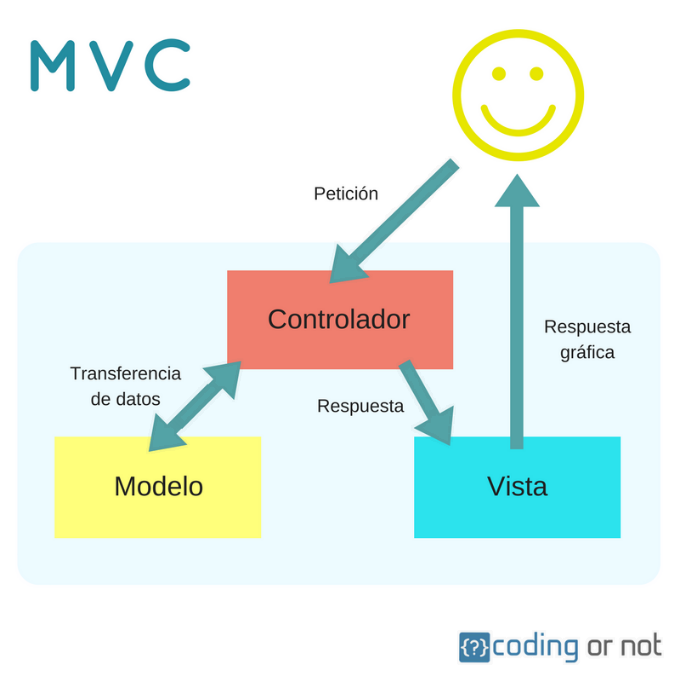


Ilustración 8: Definición grafica de la arquitectura MVC.

## **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. (Gauchat, 2012)

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. (Gauchat, 2012)

Para trabajar y crear sitios webs y aplicaciones con HTML5, necesitamos saber primero cómo esa estructura es construida. Crear fundamentos fuertes nos ayudará más adelante a aplicar el resto de los componentes para aprovechar completamente estas nuevas tecnologías.” (Gauchat, 2012)

### **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. Este nivel de cambio convierte CSS3 en una tecnología prácticamente inédita comparada con versiones anteriores. (Gauchat, 2012)

### **V.5.3 JavaScript**

**JavaScript** “es 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 “. (Gauchat, 2012)

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, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web.” (Gauchat, 2012)

### **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". (Write Less, 2016).

JQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace que cosas 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. (lancker, 2015)

### **V.5.5 Bootstrap**

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. (Efron, Bradley; Tibshirani, Robert J. ;, 2014)

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. (Laurent, 2015)

### **V.5.6 Fabric JS**

[Fabric.js](http://fabricjs.com/) , una potente biblioteca de Javascript que facilita el trabajo con HTML5 [Canvas](http://fabricjs.com/) . 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, con licencia del MIT, con muchas contribuciones a lo largo de los años. (Juriy Zaytsev, n.d.)

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](http://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. (Juriy Zaytsev, n.d.)

# **METODOLOGÍA**

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. Algo muy interesante es que la notación se basa en UML para aplicaciones web en general y para aplicaciones adaptativas en particular.

Por otra parte, el método consta de seis modelos, listados a continuación:

* 1. Modelo de casos de uso para capturar los requisitos del sistema.
  2. Modelo conceptual para el contenido (modelo del dominio).
  3. Modelo de usuario: modelo de navegación que incluye modelos estáticos y dinámicos.
  4. Modelo de estructura de presentación, modelo de flujo de presentación.
  5. Modelo abstracto de interfaz de usuario y modelo de ciclo de vida del objeto.
  6. Modelo de adaptación.

Cada uno de estos modelos 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.

# **CRONOGRAMA**

# **REFERENCIA BIBLIOGRÁFICAS**

Addison-Wesley. (2004). *Computacion Distribuida Findamentos y Aplicaciones.* Madris(España): PERSON EDUCACION, S.A.

Alfaro, J. (18 de Marzo de 2010). *Ley de Gauss.* Obtenido de Pontificia Universidad Católica de Chile: http://www.fis.puc.cl/~jalfaro/FIZ0321/clases/gauss.pdf

Alicante, U. d. (14 de Octubre de 2011). *Servicio de informatica*. Obtenido de Modelo vista controlador (MVC): https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html

Berzal, F. (2004). *El lenguaje de modelo unificado.* Madrid, España.

Bravent. (2017). *Bravent*. Obtenido de ¿Que es ASP.Net Core?: https://www.bravent.net/que-es-asp-net-core

Camacho, B. (2019). *El pequeño libro de ASP.NET Core.*

Cánovas, J. (Octubre de 2014). *TEORIA CUALITATIVA DE LAS ECUACIONES DIFERENCIALES.* Obtenido de Teoría cualitativa de ecuaciones diferenciales.: http://www.dmae.upct.es/~jose/mastering/ecuadif.pdf

Cantone, D. (2006). *Implementacion Y Debugging.* España: Creative Andina Corp.

Efron, Bradley; Tibshirani, Robert J. ;. (2014). An Introduction to the Bootstrap. En B. Efron, & R. J. Tibshirani, *An Introduction to the Bootstrap* (págs. 12-15). New York: CHAPMAN y HALL/CRC.

Enatrel. (N/A de N/A de 2016). *Subestaciones Electricas*. Obtenido de http://www.enatrel.gob.ni/subestaciones-electricas-2/

Feodorov, A. R. (1980). *Suministro Eléctrico de Empresas Industriales.* Ciudad de La Habana: Pueblo y Educación.

Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript .* Barcelona: MARCOMBO, S.A.

Gómez Fuentes, D. d. (2013). *Notas del curso de base de datos.* Mexico D.F: Universidad Autonoma Metropolitana.

Junta de Andalucia. (2015). *Los sistemas Informaticos : Conceptos Fundamentales.* Andalucia, España.

Juriy Zaytsev, S. K. (s.f.). *fabricjs*. Obtenido de fabricjs: http://fabricjs.com/fabric-intro-part-1

Lacker, L. V. (s.f.). *JQue.*

lancker, L. V. (2015). *JQuery: El framework de JavaScript de la web 2.0.*

Laurent, S. (2015). Google Maps. En S. Laurent, *Google Maps* (pág. 33). First Edition.

Manz. (2014). *Lenguaje CSS*. Obtenido de ¿Que es CSS?: https://lenguajecss.com/p/css/introduccion/que-es-css

Marquez, A. (2011). *Cliente Servidor.* Puebla, Mexico: Universidad de las Americas Puebla.

Marquez, A. (s.f.). *Capitulo 5. Cliente-Servidor.*

Maximilians, L. (10 de Agosto de 2016). *UWE - UML - based Web Engineering*. Obtenido de About UWE: http://uwe.pst.ifi.lmu.de/aboutUwe.html

Mechanism, C. D., & Mechanism Clean Development. (5 de Noviembre de 2014). *CDM UNFCCC*. Obtenido de https://cdm.unfccc.int/Projects/DB/TUEV-SUED1135170073.01

Naciones Unidas. (2007). *Reporte CEPAL.* Chile.

Núñez, F. (2015). *Centrales de generación y subestaciones electricas.* Republica Dominicana: Universidad de APEC.

Peñafiel, M. (2008). *Definición de aplicación WEB.* Chimborazo, Ecuador.: Escuela Superior Politecnica de Chimborazo.

Rocha, H. (2019). Learn Chart.js: Create interactive visualizations for the Web with Chart.js 2. En *Learn Chart.js: Create interactive visualizations for the Web with Chart.js 2* (pág. 18). New York: Packt Publishing.

Spurlock, J. (2014). Bootstrap: Responsive Web Development. En J. Spurlock, *Bootstrap: Responsive Web Development* (pág. 19). New York: Scripting News.

Walpole, R. M. (2012). *Probabilidad y estadística para ingeniería y ciencias.* México: PEARSON EDUCACIÓN.

Write Less, D. (2016). JQuery the future. En D. Write Less, *JQuery the future* (págs. 10-11). California: Angel City Press.

# **GLOSARIO DE TÉRMINOS**

**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](https://es.wikipedia.org/wiki/Dise%C3%B1o_gr%C3%A1fico) para definir y crear la presentación de un documento estructurado escrito en un [lenguaje de marcado](https://es.wikipedia.org/wiki/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 sub estación 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)

1. GLOCSE: Geolocalización de Subestaciones eléctricas. [↑](#footnote-ref-1)
2. Consumidores: Hace referencia a proveedores de energía (transformadores) que se encuentren en hospitales, parques, residenciales, entre otros. [↑](#footnote-ref-2)
3. Centro de carga: Energía consumida por consumidor en un tiempo delimitado de una hora. [↑](#footnote-ref-3)
4. Punto de Localización: Ubicación en donde será construida la sub estación eléctrica. [↑](#footnote-ref-4)
5. UML: Unified Modeling Language (Lenguaje unificado de modelado) [↑](#footnote-ref-5)
6. MVC: Modelo, Vista, Controlador. [↑](#footnote-ref-6)
7. OMG: Object Management Group. [↑](#footnote-ref-7)