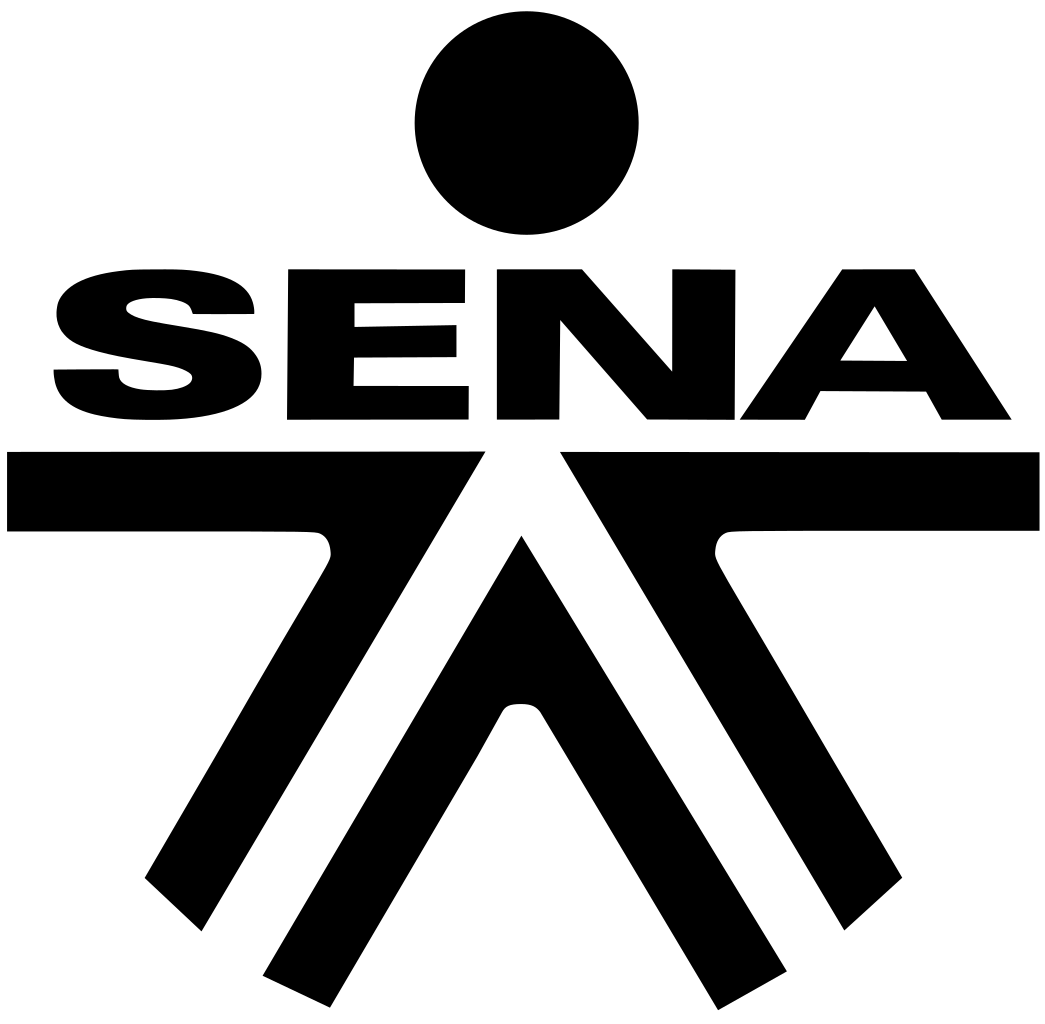
Manual de Programador

Buscador de Multímetros



Brahian Sleyder Sánchez Larrarte

Tecnólogo en Análisis y Desarrollo de Sistemas de Información

Centro de Electricidad Electrónica y Telecomunicaciones

Servicio Nacional de Aprendizaje Sena

Bogotá

2018

Tabla de Contenido

[Introducción 6](#_Toc528414037)

[Lenguaje de Programación 7](#_Toc528414038)

[PHP 7](#_Toc528414039)

[Arquitectura de software 8](#_Toc528414040)

[Modelo vista controlador (MVC) 8](#_Toc528414041)

[Frameworks 9](#_Toc528414042)

[Bootstrap 10](#_Toc528414043)

[Control de versiones 10](#_Toc528414044)

[GitHub 11](#_Toc528414045)

[Entornos de desarrollo integrados (IDE) 12](#_Toc528414046)

[PHP Strom 12](#_Toc528414047)

[Editores de código 13](#_Toc528414048)

[Visual studio Code 13](#_Toc528414049)

[Sistemas de gestión de base de datos 14](#_Toc528414050)

[MySQL 15](#_Toc528414051)

[Diseño de una base de datos 15](#_Toc528414052)

[MySQL Workbench 15](#_Toc528414053)

[Servidor Web 16](#_Toc528414054)

[XAMPP 17](#_Toc528414055)

[Núcleo de aplicativo 17](#_Toc528414056)

[Vistas 19](#_Toc528414057)

[Carpeta all 21](#_Toc528414058)

[Carpeta index 22](#_Toc528414059)

[Carpeta insertions 23](#_Toc528414060)

[Controladores 24](#_Toc528414061)

[IndexController 26](#_Toc528414062)

[LoginController 27](#_Toc528414063)

[Lógica de negocio 27](#_Toc528414064)

[Database 28](#_Toc528414065)

[Index 29](#_Toc528414066)

[Login 30](#_Toc528414067)

[Prepared SQL 31](#_Toc528414068)

[Prepared Login SQL 32](#_Toc528414069)

[Core 33](#_Toc528414070)

[APP 34](#_Toc528414071)

[APP (CSS) 35](#_Toc528414072)

[APP (JavaScript) 35](#_Toc528414073)

[Files 36](#_Toc528414074)

[Bibliografía 38](#_Toc528414075)

Tabla de ilustraciones

[Ilustración 1 Modelo vista controlador 9](#_Toc528414014)

[Ilustración 2 Diseño de DB 16](#_Toc528414015)

[Ilustración 3 Ubicación Index.php 18](#_Toc528414016)

[Ilustración 4 Index.php 19](#_Toc528414017)

[Ilustración 5 Ubicación de las Vistas 20](#_Toc528414018)

[Ilustración 6 Carpetas de las Vistas 21](#_Toc528414019)

[Ilustración 7 Carpeta all 22](#_Toc528414020)

[Ilustración 8 Carpeta index 23](#_Toc528414021)

[Ilustración 9 Carpeta insertions 24](#_Toc528414022)

[Ilustración 10 Carpeta controller 25](#_Toc528414023)

[Ilustración 11 Index Controller 26](#_Toc528414024)

[Ilustración 12 Login Controller 27](#_Toc528414025)

[Ilustración 13 Carpeta model 28](#_Toc528414026)

[Ilustración 14 Conexión con la DB 29](#_Toc528414027)

[Ilustración 15 Lógica de Negocio (Index) 30](#_Toc528414028)

[Ilustración 16 Lógica de negocio (Login) 31](#_Toc528414029)

[Ilustración 17 Prepared SQL 32](#_Toc528414030)

[Ilustración 18 Prepared login 33](#_Toc528414031)

[Ilustración 19 Core 33](#_Toc528414032)

[Ilustración 20 APP 34](#_Toc528414033)

[Ilustración 21 APP (CSS) 35](#_Toc528414034)

[Ilustración 22 APP (JS) 36](#_Toc528414035)

[Ilustración 23 Files 37](#_Toc528414036)

## Introducción

El propósito de este documento es exponer las técnicas de desarrollo de software que fueron usadas para realizar el aplicativo web Buscador de Multímetros. Esto con el fin de que la persona o personas puedan contribuir con el desarrollo de dicho proyecto, en el documento se dará a conocer los patrones y arquitecturas de desarrollo de software que fueron usadas, además se expondrá el lenguaje de programación, Clases y demás características necesarias para el correcto funcionamiento del proyecto.

## Lenguaje de Programación

# PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Fue unos de los primeros lenguajes de programación del lado del servidor que podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor con un modulo de procesador de PHP que genera el HTML resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. ​ Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3\_01, es una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, esta licencia no tiene restricciones de copyleft" asociadas con GPL.

## Arquitectura de software

En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal debido a la dificultad que entrañaba para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas y guías generales, con base a las cuales se puedan resolver los problemas. A estas, se les ha denominado **arquitectura de software**, porque, a semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software. En el libro "An introduction to Software Architecture", David Garlan y Mary Shaw definen que la arquitectura es un nivel de diseño que hace foco en aspectos "más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema".

# Modelo vista controlador (MVC)

Para el desarrollo de este aplicativo web se uso unas de las arquitecturas mas usadas en el mercado de desarrollo web, el Modelo vista controlador (MVC), ya que este nos ofrece la separación de la capa lógica y nuestra capa de diseño de nuestro aplicativo. Además permite que nuestra aplicación se encuentre en todo momento organizada y que sea de fácil acceso a todos los controladores de nuestra aplicación, esto gracias a su núcleo principal.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

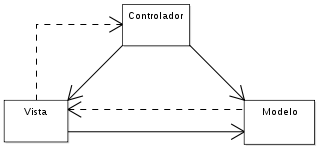


Ilustración 1 Modelo vista controlador

## Frameworks

Un ***framework****,* **entorno de trabajo** o **marco de trabajo** es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un entorno de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de *software*, que puede servir de base para la organización y desarrollo de *software*.

Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de *software* que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.

# Bootstrap

El FrontEnd de este aplicativo web fue desarrollado en su totalidad con el FrameWork de Bootstrap, ya que este FrameWork nos ofrece una amplia documentación de todos sus componentes. Además, Bootstrap es unos de los FrameWorks mas usados en el mercado de Desarrollo Web, también posee una Grilla o el manejo y distribución de columnas y alineamiento que en su última versión es de fácil comprensión y aplicación.

## Control de versiones

Se llama **control de versiones** a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados **sistemas de control de versiones** o **VCS** (del inglés *Version Control System*). Estos sistemas facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas (por ejemplo, para algún cliente específico). Ejemplos de este tipo de herramientas son entre otros: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, SCCS, Mercurial, Perforce, Fossil SCM, Team Foundation Server.

# GitHub

En este caso fue usado GitHub como sistema de Control de Versiones ya que nos permite que nuestros repositorios sean de acceso libre esto logra que este aplicativo web sea de código abierto, lo que permite que las personas que deseen aportan a este aplicativo puedan Clonar el repositorio, modificar el código y realizar un PullRequest del cambio que se realizo en el aplicativo y finalmente siendo aprobado por el administrador del repositorio. Esto garantizara que las personas con el conocimiento necesario puedan aportar al aplicativo.

A continuación proporcionamos el link del repositorio donde se encuentra alojado dicho aplicativo web, <https://github.com/Brahiasn0701/MultimeterSena>.

## Entornos de desarrollo integrados (IDE)

Un **entorno de desarrollo integrado**​ o **entorno de desarrollo interactivo**, en inglés *Integrated Development Environment* (**IDE**), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de los IDE tienen auto-completado inteligente de código (*IntelliSense*). Algunos IDE contienen un compilador, un intérprete, o ambos, tales como NetBeans y Eclipse; otros no, tales como SharpDevelop y Lazarus.

# PHP Strom

Para el desarrollo de este aplicativo fue usado el (IDE) de desarrollo PHP Strom, debido a que este IDE posee diferentes características que pueden ser de mucha a utilidad en el momento de codificar nuestro aplicativo. Estas características no las poseen los IDE de desarrollo del mercado, por estas razones se opto por el uso de PHP Strom para el desarrollo del Backend de nuestro aplicativo.

Sin embargo, la persona o personas que usen este manual como guía para desarrollar proyectos similares o modificar este aplicativo pueden hacer uso de cualquier otro IDE de desarrollo que sea de su gusto.

## Editores de código

Un **editor de código fuente** es un editor de texto diseñado específicamente para editar el código fuente de programas informáticos. Puede ser una aplicación individual o estar incluido en un entorno de desarrollo integrado.

Los editores de código fuente tienen características diseñadas exclusivamente para simplificar y acelerar la escritura de código fuente, como resaltado de sintaxis, autocompletar y pareo de llaves. Estos editores también proveen un modo conveniente de ejecutar un compilador, un intérprete, un depurador, o cualquier otro programa que sea relevante en el proceso de desarrollo de software. Por lo que, si bien muchos editores de texto pueden ser usados para editar código fuente sin problemas, si no mejoran, automatizan y facilitan la edición del código, no ameritan ser llamados "editores de código fuente", y son únicamente editores de texto que pueden ser usados para editar código fuente.

# Visual studio Code

En este aplicativo se uso Visual Studio Code específicamente para el desarrollo de FrontEnd del aplicativo.

Ya que Visual Studio Code ofrece menos características de gran utilidad que un IDE de desarrollo, en nuestro caso PHP Strom. Se decidió hacer uso de Visual Studio Code por su fácil agregación de extensiones que son de gran ayuda al momento de codificar nuestro FrontEnd, como por ejemplo el autocompletar nuestras etiquetas de HTML, la carga inmediata de todas las clases de la librería usada en el aplicativo, etc.

Sin embargo, la persona o personas que usen este manual como apoyo o busquen apoyar con el desarrollo del aplicativo pueden hacer uso del editor de código que ellos prefieran.

## Sistemas de gestión de base de datos

Un **sistema gestor de base de datos** (**SGBD**) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos .Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos.

La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas

# MySQL

Para la administración del acceso a nuestra base de datos se uso el sistema de gestión de bases de datos MySQL ya que este sistema de gestión de bases de datos es libre acceso, esto significa que no debemos pagar para hacer uso de los servicios de este SGBD ya que es totalmente gratuito.

## Diseño de una base de datos

Una base de datos correctamente diseñada le permite obtener acceso a información actualizada y precisa. Como es esencial tener un diseño correcto para lograr sus objetivos de trabajar con una base de datos, tiene sentido invertir el tiempo necesario para obtener información sobre los principios de un buen diseño. Al final, es mucho más probable que acabe con una base de datos que se ajusta a sus necesidades y que puede adaptarse fácilmente al cambio.

# MySQL Workbench

Para el diseño de la base de datos se realizo uso de la herramienta visual MySQL Workbench debido a que esta herramienta es de uso gratuito y posee de diferentes caracteristicas que no son de utilidad al momento de realizar el diseño de nuestra base de datos.

Esto con el fin de que nuestra base de datos se encuentre bien diseñada y que nos permita navegar de forma correcta y rápida por las tablas correspondientes, mediante la aplicación correcta de las relaciones entre tablas.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 2 Diseño de DB

## Servidor Web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente.

El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador.

# XAMPP

El servidor web que se uso para el alojamiento a nivel local de este aplicativo fue XAMPP debido a que es de uso libre y ofrece un panel por el cual podemos administrar los servicios que nos ofrece, podremos encender, apagar y reiniciar un servicio a través de este panel.

## Núcleo de aplicativo

El Núcleo principal de este aplicativo esta ubicado en **Index.php** este archivo es cargado de forma automática por nuestro servidor, lo encontramos fuera de todas las carpetas de nuestro aplicativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 3 Ubicación Index.php

Dentro de nuestro Núcleo encontraremos la codificación que nos permite la navegación dentro de todos nuestros controladores, todo esto por medio de la url. En Index.php es el lugar donde nuestro proyecto toma vida.

Imagen que contiene captura de pantalla, monitor

Descripción generada con confianza alta

Ilustración 4 Index.php

## Vistas

Las vistas de este aplicativo se encuentran ubicadas en una carpeta en específico, en dicha carpeta se encuentra de manera organizada cada una de las vistas de nuestro aplicativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 5 Ubicación de las Vistas

Como se menciono anteriormente dentro de la carpeta de las Vistas se encuentran organizadas todas las vistas de nuestro aplicativo, subdivididas por carpetas.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 6 Carpetas de las Vistas

# Carpeta all

Dentro de esta carpeta encontraremos los archivos fundamentales para el funcionamiento de nuestras vistas, como lo es nuestro Head, Navbar y Footer. Estos archivos son llamados posteriormente.

Imagen que contiene captura de pantalla, monitor, electrónica

Descripción generada con confianza alta

Ilustración 7 Carpeta all

# Carpeta index

Dentro de esta carpeta se encuentran los archivos que los usuarios anónimos pueden observar, es decir los usuarios que no se encuentre logueados en el aplicativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 8 Carpeta index

# Carpeta insertions

Esta carpeta contiene las vistas correspondientes a los usuarios que ya se encuentran logueados en el aplicativo, más específicamente dentro de esta carpeta se encuentra la vista de los administradores.

Imagen que contiene captura de pantalla, monitor

Descripción generada con confianza muy alta

Ilustración 9 Carpeta insertions

## Controladores

Los controladores en este aplicativo forman parte fundamental de el con respecto al funcionamiento del aplicativo, es aquí donde comenzamos a darle funcionamiento a nuestro aplicativo, todos los controladores se encuentran en una carpeta llamada Controller.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 10 Carpeta controller

Dentro de la carpeta donde se encuentran los controladores encontraremos los controladores que dan vida a este aplicativo, cada uno de ellos desempeña una función diferente dentro del aplicativo.

# IndexController

Este controlador nos ofrece básicamente toda la funcionalidad de nuestro aplicativo, en este archivo realizamos algunas técnicas de la programación orientada a objetos (POO) y luego retornamos resultados en las vistas, además es aquí donde cargamos y llamamos nuestras vistas.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 11 Index Controller

# LoginController

En esta clase se define todas las funciones que nos ayudaran a realizar el Login de nuestro aplicativo, esto se separó del controlador IndexController por motivos de seguridad. Igualmente, dentro de esta clase se aplican algunas técnicas de la programación orientada a objetos (POO).

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 12 Login Controller

## Lógica de negocio

La lógica de negocio donde el aplicativo tiene interacción directa con nuestra base de datos se encuentra en la carpeta model, es allí donde se realiza todo la matriz crud del aplicativo.

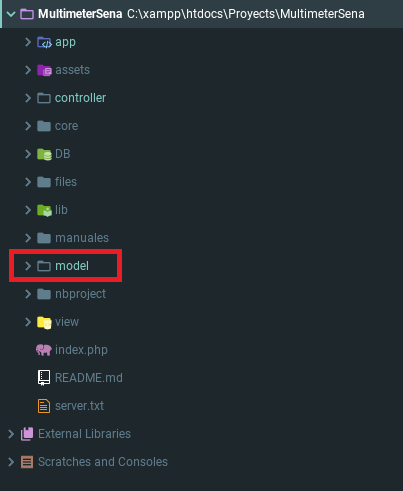


Ilustración 13 Carpeta model

# Database

El archivo Database que se encuentra dentro de la carpeta model es la encargada de realizar la conexión con nuestra base de datos y de manejar la excepción en caso de que la conexión sea incorrecta.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 14 Conexión con la DB

# Index

Dentro del archivo index es donde realizamos todas nuestras inserciones, consultas, actualización y eliminaciones que son necesarias en nuestro aplicativo. Para realizar esto heredamos la conexión, para así conectar con la base de datos.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 15 Lógica de Negocio (Index)

# Login

En este archivo usamos la conexión con la base de datos y posteriormente realizamos la interacción necesaria para que un respectivo usuario pueda iniciar sesión correctamente.

Imagen que contiene captura de pantalla, pantalla, sentado

Descripción generada con confianza alta

Ilustración 16 Lógica de negocio (Login)

# Prepared SQL

Dentro de la lógica de negocio encontraremos varios archivos llamados PreparedSQL estos archivos son los encargados de definir las constantes SQL que usamos en nuestro aplicativo y que respectivamente llamamos desde los demás archivos de la lógica de negocio.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 17 Prepared SQL

# Prepared Login SQL

En este prepared realizamos todas las constantes que son necesarias para el inicio de sesión o login de nuestro aplicativo y posteriormente usamos desde los demás archivos de la lógica de negocio.

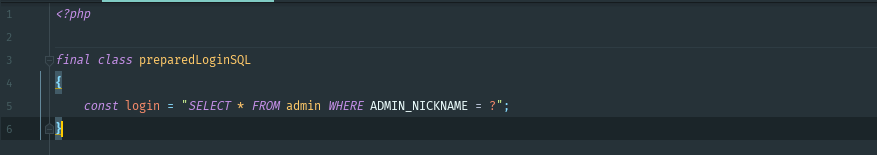


Ilustración 18 Prepared login

## Core

La carpeta core contiene uno de los archivos mas importantes para el funcionamiento de nuestro aplicativo, debido a que este archivo es el encargado de requerir por medio de un (Require) toda la lógica de negocio y demás archivos y luego incluirlos dentro del núcleo de nuestro aplicativo. Además es aquí en donde inicializamos las sesiones, esto es importante al momento de iniciar sesión.

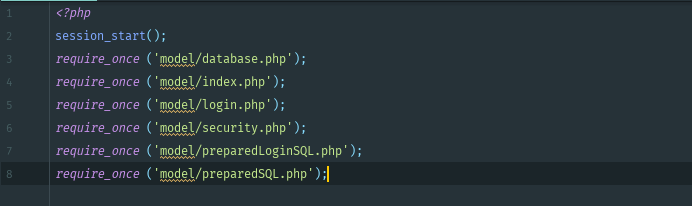


Ilustración 19 Core

## APP

Dentro de la carpeta app encontramos gran parte del funcionamiento de nuestro aplicativo, en esta carpeta encontraremos los archivos JavaScript y Css, estos archivos hacen parte fundamental tanto para el diseño y el funcionamiento de nuestro aplicativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 20 APP

# APP (CSS)

Dentro de la hoja de estilos que encontraremos dentro de la carpeta APP estarán algunos de los estilos que fueron necesarios para el desarrollo del diseño de esta aplicativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza alta

Ilustración 21 APP (CSS)

# APP (JavaScript)

Dentro de la carpeta APP también encontraremos un archivo JavaScript, debemos aclarar que dentro de este archivo no solo usamos JavaScript, casi en todo el archivo se usa JQuery. Es de vital importancia el uso de este archivo ya que casi todo el aplicativo funciona de forma asíncrona todas las funciones AJAX son ejecutadas de este archivo, estas funciones apuntan a diferentes lugares de nuestro aplicativo.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 22 APP (JS)

## Files

La carpeta Files nos sirve de ayuda para saber que imágenes se han subido a el servidor, estas imágenes solo las sube el administrador del sistema. Además, el nombre de las imágenes que se encuentran en la carpeta es encriptado lo que significa que el nombre será una cadena de texto bastante larga.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Ilustración 23 Files

## Bibliografía

* Servidor Web. En Wikipedia. Recuperado el 7 de agosto de 2018 de
* <https://es.wikipedia.org/wiki/Servidor_web>
* PHP. En Wikipedia. Recuperado el 13 de octubre de 2018 de
* <https://es.wikipedia.org/wiki/PHP>
* Arquitectura de Software. Wikipedia. Recuperado el 10 de agosto de 2018 de
* <https://es.wikipedia.org/wiki/Arquitectura_de_software>
* Framework. En Wikipedia. Recuperado el 5 de octubre de 2018 de
* <https://es.wikipedia.org/wiki/Framework>
* Control de versiones. En Wikipedia. Recuperado el 26 de septiembre de 2018 de <https://es.wikipedia.org/wiki/Control_de_versiones>
* Entorno de desarrollo integrado. En Wikipedia. Recuperado el 29 de septiembre de 2018 de <https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado>
* Editor de código fuente. En Wikipedia. Recuperado el 10 de enero de 2017 de <https://es.wikipedia.org/wiki/Editor_de_c%C3%B3digo_fuente>
* Sistema de gestión de base de datos. En Wikipedia. Recuperado el 23 de octubre de 2018 de <https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos>