## 



S-Gadgets

Proyecto de Tienda en Línea.

Proyecto de Sistemas Web.

**─**

Luis Aaron Granillo Hernández

Joyce Enid García López

Omar Alfredo Hidalgo Jacinto

E.E Sistemas Web, NRC:80688  
Lic. Tecnologías Computacionales  
Universidad Veracruzana – diciembre 2022

# Introducción

Este documento, registra la información que se necesita para comprender la realización del proyecto integrador de la Experiencia Educativa de Sistemas Web.

Este es un Sistema Web sobre una página de Tienda en línea llamada: “S-Gadgets”, en la cual, tiene varias funciones las cuales permiten accesibilidad en muchos de los aspectos básicos de un sistema web.

Este proyecto fue desarrollado en HTML, Java en uso de framework Spark.

Se implementó en el sistema de control de versiones GitHub donde los miembros del equipo tenían acceso a las versiones más recientes de las partes correspondientes del proyecto, esto para poder trabajar de manera síncrona desde diferentes puntos y manejar las versiones de trabajo.

# Propósito

El propósito general y el de la Experiencia Educativa es realizar software funcional y usable, mismo que pueda solucionar una necesidad.

El propósito es lograr la funcionalidad de un Sistema Web haciendo uso de herramientas de desarrollo web.

Los lenguajes utilizados fueron Java, HTML, CSS y JavaScript, mismos que se utilizaron para el desarrollo de lo siguiente

* Java : para el desarrollo de BackEnd, y la programación usando el framework de Spark
* HTML, CSS, JavaScript para el desarrollo de interfaces de usuario y FrontEnd

# Propuesta de Negocio

Esta es una tienda en línea de aparatos electrónicos y gadgets, en esta existen 2 tipos de usuario, los clientes y un administrador.

Por un lado los clientes son aquellos que visitan la página, para consultar los productos de la tienda, su uso principal es de un catálogo, El administrador es aquel que pueden iniciar sesión con el propósito de poder tanto eliminar, agregar elementos a la tienda y editarlos que están en existencia, esta acción la hace con solo 4 elementos necesarios, el nombre del producto, el precio del producto, el link/URL de una imagen de referencia del producto a agregar y el ide en caso de actualización.

Todos los administradores son registrados previamente, tienen un usuario y contraseña específicos. Los elementos de la tienda pueden ser visualizados y en el pie de la imagen se da más información sobre cada uno de ellos.

# Desarrollo

Como primera parte, creamos un proyecto con la herramienta de software que ofrece Maven, y también utilizando el framework de Spark, ambas herramientas de uso gratuito.

Para el despliegue de BackEnd su uso la plataforma de despliegue de Railway una plataforma que permite habilitar funciones de manera local y desplegarlas en la nube para poder acceder de manera remota a “la parte trasera”, del proyecto.

De igual manera la plataforma de GitHub nos permite desplegar en la nube el FrontEnd del sistema, lo cual nos permite, tener un sitio funcional en la nube.

Como parte de este desarrollo se crearon varios archivos, los cuales te vamos a describir los más importantes:

**BackEnd.**

* App.java
  + Archivo ejecutable java del sistema, aquí se encuentran las ejecuciones de los procesos que necesita el sistema, para que sea funcional, como el “/creaProductos”, el cual permite al administrador entrar en la ventana donde añade nuevos productos.
  + También está el “/borrarProducto” el cual ejecuta los procesos para que el administrador elimine el producto que desee.
  + Está el “/productos” donde muestra la página principal del sistema.
  + Esta “/actualizarProducto” donde lanza la interfaz que sucede cuando el administrador actualiza los datos de un producto, ya sea porque, el producto cambio de precio, de variación de modelo o simplemente hubo un error de sintaxis.

cabe mencionar que se tiene una estructura predeterminada para la implementación del mapeo de los objetos Json con los nombres de la estructura de nuestra clase Producto

{

id:

nombre:

precio:

link:

}

Con estos nombres se trabajó bidireccionalmente para hacer un mapeo correcto.

* Conexión.java
  + En este archivo se encuentran los datos que permiten la conexión a nuestra base de datos en la nube, datos como, el driver de conexión, el url de la base de datos, el usuario y contraseña para poder acceder a los datos.
* DAO.java
  + En este archivo se encuentran todas las sentencias de SQL que son necesarias para el funcionamiento del sistema, además de ser las órdenes básicas en el manejo de una base de datos.
* Pom.xml

Archivo con extensión xml (Extensible Markup Language). donde se encuentran todas las dependencias, propiedades, y plugins que son necesarios para el despliegue

* Usuario.java
  + Clase de java donde el usuario generado como clase, se crean para la utilización de los atributos como lo son el nombre, el correo y la contraseña, mismas que son utilizadas y permiten que a lo largo de el backend puedan ser utilizados a lo largo del backend.
* Producto.java
  + Clase de java que crea la clase Producto la cual tiene como atributos, ID, nombre, precio y link (este usado como la referencia al link/URL de la imagen del producto).

**FrontEnd.**

* Index2.html
  + En este archivo de HTML (Lenguaje de marcas de HiperTexto), se encuentra la pantalla principal, es decir la pantalla que ve el usuario inmediato, puede ver los productos de la página, y los realizadores de esta.
* Index.html
  + Este archivo HTML tiene la diferencia que esta es la vista de sesión que visualiza el administrador que previamente iniciar sesión. este archivo tiene la diferencia que ahora hay campos para agregar, editar, eliminar o actualizar un producto, para agregar son necesarios el Nombre, precio y link de la imagen, para editar es necesario el ID del producto, el precio, y el link de la imagen, para eliminar el producto, son necesarios el nombre el precio y el ID.
* login.html
  + En este archivo HTML está la pantalla que visualiza el administrador, por el cual, con su usuario y contraseña previamente registrados pueden acceder a la pantalla principal esta vez con los cambios descritos en la parte index.html.
* Main.css
  + Archivo de CSS (Hojas de Estilo en Cascada), donde se encuentran las modificaciones a los elementos de los archivos html, estos se pueden encontrar clasificados por medio del nombre de la clase a la que pertenecen, o a la parte del documento en la que fueron ordenados, también en ciertas partes del archivo, se pueden encontrar breves descripciones donde indique en qué parte del documento HTML se encuentra con breves anotaciones.
* ValidarUsuario.js
  + En este archivo JS, este analiza las entradas en la parte de login.html, el cual analiza si el usuario y la contraseña, correspondan con el registro previo, si no es correcto o tuvo una equivocación, lanza una alerta indicando que quedan cierta cantidad de intentos de inicio.

# Anexos

Link a los repositorios de GitHub.

Link de la base de datos Railway: [mysql://root:FkTkEsfKGhaGUsXdtHKm@containers-us-west-91.railway.app:7884/railway](https://railway.app/project/02da3d56-7253-4992-8e9a-cb6998a0a71c/plugin/9dd8f760-c0df-41d8-9154-431e509dfd82/Connect)

**BackEnd**: <https://github.com/aaronMan12/BackEnd>

**FrontEnd**: <https://github.com/JoyceEnidGL/FrontEnd>

Link a los sitios en la nube de las respectivas partes del proyecto.

**BackEnd**: <https://backend-production-1432.up.railway.app/>

**FrontEnd**: <https://joyceenidgl.github.io/FrontEnd/>