

MANUAL TÉCNICO

PUNTO DE VENTA: GAMER PRO XELA PROYECTO 1 DEL CURSO: MANEJO E IMPLEMENTACIÓN DE ARCHIVOS

BREVE DESCRIPCIÓN

El presente proyecto, es una implementación de un sistema que permite llevar un control del inventario de los productos, empleados, sucursales y clientes que la empresa Gamer Pro Xela posee.

HERRAMIENTAS UTILIZADAS

● LENGUAJES DE PROGRAMACIÓN: TYPESCRIPT y JAVASCRIPT

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se publicó en julio de 2015.

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. Anders Hejlsberg, diseñador de C# y creador de Delphi y Turbo Pascal, ha trabajado en el desarrollo de TypeScript.

TypeScript es usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor, o extensiones para programas (Node.js y Deno).

TypeScript extiende la sintaxis de JavaScript, por tanto cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen al código JavaScript original.

● VUE JS 3 - INTERFAZ DE USUARIO

Vue.js (comúnmente conocido como Vue; se pronuncia /vjuː/, como "view") es un framework de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. Fue creado por Evan You, y es mantenido por él y por el resto de los miembros activos del equipo central que provienen de diversas empresas como Netlify y Netguru.

Descripción

Vue.js cuenta con una arquitectura de adaptación gradual que se centra en la representación declarativa y la composición de componentes. La biblioteca central se centra sólo en la capa de vista. Las características avanzadas necesarias para aplicaciones complejas como el enrutamiento, la gestión de estados y las herramientas de construcción se ofrecen a través de librerías y paquetes de apoyo mantenidos oficialmente, con Next.js como una de las soluciones más populares.

Vue.js permite extender el HTML con atributos HTML llamados directivas. Las directivas ofrecen funcionalidad a las aplicaciones HTML, y vienen como directivas incorporadas o definidas por el usuario.

También se utilizaron librerías/bibliotecas externas de componentes:

- **Vuetify 3**

● EDITOR DE TEXTO: VISUAL STUDIO CODE

Visual Studio Code (también llamado VS Code) es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es software gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Monaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

● GITHUB

Es un sistema de control de versiones de código y gestión de proyectos, a su vez también funciona como una plataforma de estilo red social diseñada para desarrolladores para poder compartir código entre más personas y colaborar en el mismo. Se utilizó GitFlow como metodología

● MACOS v14 - SISTEMA OPERATIVO

macOS (previamente Mac OS X, luego OS X) es una serie de sistemas operativos gráficos desarrollados y comercializados por Apple desde 2001. Es el sistema operativo principal para la familia de computadoras Mac de Apple. Dentro del mercado de computadoras de escritorio, portátiles, hogareñas y mediante el uso de la web.

macOS se basa en tecnologías desarrolladas entre 1985 y 1997 en NeXT. Se logró la certificación UNIX 03 para la versión Intel de Mac OS X 10.5 Leopard y todos los lanzamientos de Mac OS X 10.6 Snow Leopard hasta la versión actual también tienen la certificación UNIX 03. macOS comparte su núcleo basado en Unix, llamado Darwin, y muchos de sus frameworks con iOS 16, tvOS y watchOS.

● POSTGRESQL - RDBMS

PostgreSQL, también llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL no tiene un gestor de errores (bugs), haciendo muy difícil conocer el estado de corrección de los mismos

ARQUITECTURA DE LA APLICACIÓN

Este proyecto se enfoca principalmente en el uso del motor de base de datos Postgresql, por lo que se prioriza el uso de funcionalidades que proporciona este como los *triggers* y los *procedures* o *funciones*.

En el directorio *data* del proyecto se puede encontrar los scripts necesarios para configurar la base de datos junto con todas sus automatizaciones y también para poblar la misma y tener la base de datos lista con algunos datos ya predeterminados.

En el caso del backend de este proyecto, se utilizó el framework Nest.js que permite una arquitectura parecida a modelo-vista-controlador, y también sigue el patrón de inyección de dependencias para evitar la duplicación de instancias a lo largo de los diferentes módulos que posee la aplicación a nivel de backend.

Se divide por módulos y estos a su vez contienen, los servicios quienes albergan toda la lógica de manejo de datos y luego están los controladores quienes manejan las rutas y las peticiones que llegan a este.

Para el caso del frontend, se utilizó un framework de javascript de frontend que permite hacer y maquetar interfaces de usuario de forma declarativa por lo que la experiencia de desarrollo es agradable y es rápido crear interfaces con funcionalidad de esta manera.

SCRIPTS DE LA BASE DE DATOS

1. Archivo database.sql:

En este archivo se encuentran todas las instrucciones para la inicialización de la base de datos, tanto su creación como la creación de esquemas y tablas en la misma.

2. Archivos de datos

En los diferentes archivos con nombres de algunas entidades se encuentran por separado algunas inserciones de datos que permiten poblar la base de datos con información básica para el funcionamiento de la aplicación

3. Archivo seed.sql

Una combinación de los archivos anteriores para fácil ejecución en el motor de base de datos

4. Archivo functions.sql

Este archivo contiene las funciones que se utilizan a nivel de base de datos para asegurar la integridad de datos y también provee algunas verificaciones al momento de hacer inserciones en algunas tablas de la base de datos.

INSTRUCCIONES PARA EJECUCIÓN

A continuación, se presentan los pasos necesarios para poner en funcionamiento el proyecto.

Se requiere tener instalado el entorno de ejecución Node.js versión 20

- Descargar el código fuente desde el repositorio
- Ejecutar los scripts de creación y población de la base de datos
- Instalar dependencias con npm tanto como en el proyecto de frontend como de backend
- Ejecutar los scripts de ejecución de ambos proyectos
- Abrir el navegador en la dirección localhost en el puerto 3000