MYSQL Workbench 8.32.2

Es una herramienta visual de diseño de base de datos integra desarrollo de software, administración, diseña, creación y mantenimiento del almacenamiento para todo el sistema. Es uno de los primeros productos de MYSQL el cual ofrece una versión de código abierto y una de edición comercial.

HTML 5

El lenguaje marcado de hipertexto es el código que se utiliza para estructurar y desplegar una página web y sus contenidos visuales. Consiste en una serie de elementos que se utilizan para encerrar diferentes pastes del contenido para visualizar su comportamiento de una manera determinado.

JavaScript

Es un lenguaje de programación robusto de secuencias de comandos que permite crear contenidos que se puede aplicar en un documento HTML para crear interactividad con actualización dinámica compleja en el sitio web con un mínimo de esfuerzo.

Bootstrap versión 4.4

Es un framework Font-end utilizado para desarrollar aplicaciones web y sitios de teléfonos inteligentes, es de código abierto y combina Javascrip y CSS para estandarizar los elementos.

CSS

Es un lenguaje que determina el estilo de documentos HTLM, abarca opciones relativas a fuentes, colores, márgenes, línea, altura, anchura, imágenes de fondo entre otros. Ofrece varias opciones y es muy preciso, además es compatible con todos los navegadores actuales.

Visual studio 2019

Es un entorno de desarrollo integrado (IDE por sus siglas en ingles) para Windows y macOS. Es compatible con múltiples lenguajes de programación como C++, C#, .NET entre otros al igual que entornos de desarrollo web como ASP.NET.

API

Las API son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. Las API simplifican en gran medida el trabajo de un creador de programas, ya que no tiene que «escribir» códigos desde cero. Estas permiten al informático usar funciones predefinidas para interactuar con el sistema operativo o con otro programa.

CRUD

En programación el acrónimo CRUD (Create, Read, Update and Delete) se utiliza para hacer referencia a los algoritmos a implementar para permitir crear, leer, modificar y borrar datos de una tabla de una base de datos. CRUD resume las funciones requeridas por un usuario para crear y gestionar datos. Varios procesos de gestión de datos están basados en CRUD, en los que dichas operaciones están específicamente adaptadas a los requisitos del sistema y de usuario, ya sea para la gestión de bases de datos o para el uso de aplicaciones.

Lenguaje C#

C# es un lenguaje de programación desarrollado por Microsoft, orientado a objetos, que ha sido diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. Se trata de un lenguaje simple, eficaz y con seguridad de tipos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C.

La sintaxis viene derivada de C y C+ + y utiliza el modelo de objetos de la plataforma .NET, muy parecido al de Java, aunque incluye mejoras propias de otros lenguajes .

# Arquitectura y descripción de la arquitectura implementada

**¿Qué es la arquitectura de Software?**

Se entiende por arquitectura de software a una serie de reglas, estándares y guías generales, con base a las cuales se puedan resolver los problemas. Proporciona un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores y analistas del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

*Tipos de arquitectura*

Entre las arquitecturas más comunes se encuentran:

1. Arquitectura de Capas
2. **Cliente-Servidor**
3. Peer to Peer
4. **M-V-C**

En este proyecto se hará mención y ocupación de la arquitectura cliente-servidor y la arquitectura MVC.

**¿Qué es la Arquitectura 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 y Controlador.

Modelo: este componente se encarga de manipular, gestionar y actualizar los datos. Si se utiliza una base de datos aquí es donde se realizan las consultas, búsquedas, filtros y actualizaciones.

Vista: este componente se encarga de mostrarle al usuario final las pantallas, ventanas, páginas y formularios como resultado de una solicitud enviada a través del controlador. Desde la perspectiva del programador este componente es el que se encarga del frontend; la programación de la interfaz de usuario si se trata de una aplicación de escritorio, o bien, la visualización de las páginas web (CSS, HTML, HTML5 y JavaScript).

Controlador: este el componente principal de la aplicación, donde se especifican los métodos y funcionalidades que una aplicación (o módulo de una aplicación) tienen que realizar. Se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. A través del controlador se realizan las consultas al modelo (una búsqueda, por ejemplo), y una vez se hayan obtenido dichos datos, se envía a la vista las instrucciones necesarias para poder mostrarlos de una forma legible para el usuario.

*Características de cada componente de esta Arquitectura*

Modelo:

1. Contiene una representación de los datos que maneja el sistema
2. Define las reglas de negocio (que es lo que va a hacer el sistema)
3. Es el encargado de acceder a la capa de almacenamiento de datos

Vista:

1. Es la encargada de la presentación e interacción del sistema con el usuario

Controlador:

1. Actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno
2. Recibe los eventos de la entrada (un clic, un cambio en un campo de texto, etc)
3. Define la lógica de presentación, mediante manejadores de eventos

**Interacción de los componentes**

1. El flujo que habitualmente se sigue en este estilo de arquitectura es el siguiente:
2. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
3. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
4. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
5. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo.
6. El modelo no debe tener conocimiento directo sobre la vista.
7. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista.
8. El controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice.
9. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

**Arquitectura Cliente-Servidor**

Una arquitectura o modelo cliente-servidor es una red de aplicaciones que separa las tareas entre los clientes y los servidores que viven en el mismo sistema o tienen que comunicarse a través de una red informática. El servidor-cliente se basa principalmente en enviar la solicitud a otro programa para acceder al servicio que el servidor hace accesible. Este servidor ejecuta algunos programas que comparten los recursos y distribuyen el trabajo entre los clientes.

Una relación cliente-servidor corresponde al patrón de solicitud-respuesta y debe adherirse al procedimiento de comunicaciones común que define el lenguaje, las reglas o los patrones de diálogo utilizados. La comunicación cliente-servidor se adhiere al conjunto de protocolos TCP o IP.

*Definiciones:*

Cliente: La palabra Cliente significa una organización o un individuo que utiliza un servicio. Incluso en el mundo digital, el Cliente es un Host (computadora) que puede recibir información o utilizar el servicio de los Servidores.

Servidor: Servidor significa una persona que sirve algo. El servidor, en el mundo digital, es la computadora remota que ofrece información o acceso a servicios.

