



CENTRO DE ESTUDIOS CIENTIFICOS Y TECNOLOGICOS

“JUAN DE DIOS BATIZ”

Laboratorio de Desarrollo de Software III



Arquitectura de software

5IV8

PROFESOR: Román Alvarado Reyes

Equipo:

- Esquivel Zarate Leonardo
- Hernández Hernández Cesar Eduardo
 - López García Miguel Ángel
- Rueda Gutierrez Adrián Everardo
 - Camargo Huertero Jesús Saul
 - Flores Raya Cesar Damián

I. Componentes y módulos

- Bases de datos:
El software guarda mediante bases de datos la información proporcionada del usuario y de sus respectivos clientes para su debida gestión dentro del software.
- Interfaz de usuario:
La interfaz será de uso sencillo e intuitiva para ambas partes (cliente y mecánico), agradable a la vista que permita una comunicación sencilla para poder guiar a ambas partes a través, realizando sus tareas.
- Módulos de clientes:
Dependiendo el tipo de software que estemos hablando (el móvil, web o de escritorio), cada parte permite posibilidades distintas al usuario cliente. En la versión móvil, el cliente podrá visualizar el paso de su vehículo por el taller mecánico. La versión de escritorio, al ser principalmente hecha para el mecánico permitirá guardar los datos del cliente y sus respectivos vehículos.
- Módulos de mecánicos:
El módulo de mecánicos no cambiara tanto en las versiones, el mecánico tiene la tarea de llevar a cabo el debido tratamiento del vehículo, los gastos que van surgiendo para este y llevar un seguimiento en la bitácora.
- Módulos de compra y venta:
El taller mecánico contara con un debido catalogo que tenga ciertos productos para cada parte del vehículo, cada producto cuenta con un precio distinto y estos pueden ser comprados, antes de que el vehículo ingrese o durante su estancia.
- Modulo de bitácora:
La bitácora será gestionada por el mecánico, detallando los cambio y modificaciones hechas en el vehículo, la bitácora también podrá ser revisada por el cliente en la aplicación web o móvil para así ver lo que se le ha hecho al vehículo.

II. Estilos arquitectónicos

El estilo arquitectónico recomendado para el sitio web sería: Arquitectura en capas o cliente-servidor.

- **Arquitectura Cliente-Servidor**
En este estilo, el frontend (cliente) envía solicitudes HTTP (a través de formularios, botones, etc.) al backend (servidor), que responde con datos o recursos. El servidor está escrito en Java (por ejemplo, usando **Spring** o **Java EE**), mientras que el cliente está basado en HTML y JavaScript.
- **Arquitectura en Capas**
En la arquitectura en capas, la aplicación se divide en varias capas:
 1. **Capa de presentación** (HTML, CSS, JavaScript): La interfaz de usuario que interactúa con el usuario final.
 2. **Capa de lógica de negocio** (Java): Contiene la lógica empresarial y reglas de negocio.
 3. **Capa de acceso a datos** (Java): Gestiona la interacción con bases de datos y otros sistemas de almacenamiento.
 4. **Capa de almacenamiento de datos**: Donde se encuentran las bases de datos.

III. Decisiones arquitectónicas

- **Elección de estilo arquitectónico**
Anteriormente mencionado, se especifico que las arquitecturas más útiles para la elaboración del software serian la arquitectura cliente-servidor y la arquitectura en capas.
- **Tecnologías y herramientas**
Todos los integrantes de la empresa cuentan con una computadora especial para poder realizar lo que se pide, entre los principales softwares para programar, el mas importante será Visual Studio Code y NeatBeans, ambos apoyados con los lenguajes como Java, Javascript, HTML y CSS.
- **Persistencia de datos**
Los datos serán gestionados con bases de datos realizadas en MySQL.
- **Disponibilidad y tolerancia a fallos**

Todos los integrantes realizaremos varias pruebas para así poder dar una tolerancia a fallos y un mejor resultado final.

- **Mantenimiento y extensibilidad**
El mantenimiento que se le dará al sistema será puntual, desde las pequeñas quejas para así poder entregar un producto final eficiente.

IV. Atributos de calidad

- **Rendimiento**
La empresa esta compuesta por programadores eficientes que pueden dar calidad al software correspondiente, todas las tareas serán realizadas con un horario en tiempos calculados.
- **Escalabilidad**
El software tendrá la capacidad para manejar un aumento en la carga o en el tamaño del sistema sin degradar significativamente el rendimiento.
- **Disponibilidad**
El software contara con la capacidad del sistema para estar operativo y accesible cuando se necesita, minimizando el tiempo de inactividad.
- **Mantenibilidad**
El software contara con bastante facilidad al modificar, corregir, mejorar o adaptarse a las necesidades después de su implementación.
- **Portabilidad**
El software al principio solo funciona como un programa de escritorio, pero se piensa expandir más, llegando a ser un software web y una aplicación móvil.
- **Usabilidad**
El software será fácil de usar para cualquier usuario, dándole formas de aprender usar y operar el sistema con una gran satisfacción y sencillez.