

**UNIVERSIDAD POLITÉCNICA DE ATLACOMULCO**

Ingeniería en Sistemas Computacionales

**PROYECTO DE ESTANCIA I o II**

**“NOMBRE DEL PROYECTO”**

Presentado por:

ORLANDO RAMOS PÉREZ

Profesor Asesor:

JUAN GARCIA VILLAFUERTE

|  |  |
| --- | --- |
| Calificación Asesor Empresarial: |  |
| Calificación y firma del Profesor Asesor: |  |
| Calificación Final: |  |

Atlacomulco, México. Diciembre 2024

Resumen

Resumen del proyecto. Una cara. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres. En un máximo de una cuartilla o 300 caracteres

**Palabras clave**: un par de líneas

**Usar salto de página y sección, NO usar enter.**

Índice general

1. [Introducción y objetivos 1](#_Toc182210859)

[1.1 Introducción 1](#_Toc182210860)

[1.2 Justificación 1](#_Toc182210861)

[1.3 Objetivos 2](#_Toc182210862)

[1.4 Problema a resolver 3](#_Toc182210863)

[1.5 Alcances y limitaciones 3](#_Toc182210864)

1. [Marco teórico 4](#_Toc182210865)

[2.1 Definición de Sistema WEB. 4](#_Toc182210866)

[2.1.1 Elementos de un sistema en WEB. 4](#_Toc182210867)

[2.1.2 Funcionamiento de un Sistema WEB. 5](#_Toc182210868)

[2.1.3 Tipos de sistema WEB 5](#_Toc182210869)

[2.2 Auditoria por Capas 5](#_Toc182210870)

[2.2.1 Concepto de Auditoria por Capas. 6](#_Toc182210871)

[2.2.2 Ejemplo de Auditoria por Capas. 6](#_Toc182210872)

[2.2.3 Beneficios de las auditorias por capas. 6](#_Toc182210873)

[2.3 Mockups 7](#_Toc182210874)

[2.3.1 Herramientas utilizadas 7](#_Toc182210875)

[2.4 Base de datos 8](#_Toc182210876)

[2.4.1 Tipos de Bases de datos 8](#_Toc182210877)

[2.4.2 Tipos de Gestores de Base de Datos. 9](#_Toc182210878)

[2.5 LARAGON 9](#_Toc182210879)

[2.6 Que es un FRAMEWORK 10](#_Toc182210880)

[2.6.1 Ventajas de usar un FRAMEWORK. 10](#_Toc182210881)

[2.7 CODEIGNITER 4 11](#_Toc182210882)

[2.8 PHP 12](#_Toc182210883)

[2.8.1 ¿Por qué PHP? 12](#_Toc182210884)

[2.9 MVC 13](#_Toc182210885)

[2.9.1 ¿Qué significa MVC? 13](#_Toc182210886)

[2.9.2 Porque MVC 13](#_Toc182210887)

[2.10 JAVA SCRIPT 14](#_Toc182210888)

[2.10.1 Características principales de Java Script 14](#_Toc182210889)

[2.10.2 Java Script en el Frontend 15](#_Toc182210890)

[2.11 HTML 15](#_Toc182210891)

[2.12 CSS 16](#_Toc182210892)

[2.12.1 Que es CSS 16](#_Toc182210893)

[2.12.2 Conceptos claves sobre CSS 16](#_Toc182210894)

[2.13 VISUAL STUDIO CODE 17](#_Toc182210895)

[2.13.1 ¿Por qué Visual Studio Code? 17](#_Toc182210896)

[2.14 HEIDI SQL 18](#_Toc182210897)

[2.14.1 Características de HeidiSQL. 18](#_Toc182210898)

[2.15 NAVICAT 19](#_Toc182210899)

[2.16 GIT 19](#_Toc182210900)

[2.16.1 ¿Qué es Git y cómo funciona? 19](#_Toc182210901)

[2.16.2 Características principales de GIT 20](#_Toc182210902)

[2.16.3 Ventajas de GIT 20](#_Toc182210903)

[2.17 GITHUB 21](#_Toc182210904)

[2.18 FIREFOX DEVELOPMENT 21](#_Toc182210905)

[2.19 BOOTSTRAP 21](#_Toc182210906)

[2.20 W3-SCHOOL 21](#_Toc182210907)

[2.21 STAR UML 21](#_Toc182210908)

1. [Metodología 22](#_Toc182210909)

[3.1 Exploración 22](#_Toc182210910)

[3.1.1 Historias de usuario del administrador. 22](#_Toc182210911)

[3.1.2 Historias de Usuario del Auditor 25](#_Toc182210912)

1. [Desarrollo 27](#_Toc182210913)

[4.1 Desarrollo de Mockups 27](#_Toc182210914)

[4.1.1 Desarrollo de login y “Olvide mi contraseña” 27](#_Toc182210915)

[4.1.2 Desarrollo de mockups del administrador. 29](#_Toc182210916)

1. [Productos del proyecto 32](#_Toc182210917)

[7.1 Contenido 32](#_Toc182210918)

[7.2 Más información 32](#_Toc182210919)

[7.3 Más información 32](#_Toc182210920)

1. [Conclusiones y recomendaciones 34](#_Toc182210921)

[8.1 Contenido 34](#_Toc182210922)

[8.2 Más información 34](#_Toc182210923)

[8.3 Más información 34](#_Toc182210924)

1. [Referencias 36](#_Toc182210925)

Índice de figuras

[Figura 1. Evolución y previsión del número de usuarios en España (inferior) [AUI] 2](#_Toc27217078)

[Figura 2. Porcentaje de españoles que compran por Internet [AIMC] 4](#_Toc27217079)

Índice de tablas

[Tabla 1. Relación entre ancho de banda de las tecnologías y aplicaciones 2](#_Toc27217080)

Capítulo 1

Introducción y objetivos

* 1. Introducción

Describe brevemente de la institución, organismo o empresa y el departamento donde desarrolla la Estancia o Estadía; el panorama general del tema relacionado con el proyecto; los antecedentes sobre el tema y el impacto del trabajo en el contexto de la Ingeniería en Sistemas Computacionales y la actividad que desarrolla la institución, organismo o empresa

* 1. Justificación

La ejecución de un sistema automatizado de **auditorías por capas** es crucial para mejorar la eficiencia y efectividad en el control de los procesos organizacionales, específicamente en las áreas de seguridad, calidad y productividad. Actualmente, la captura manual de datos durante las auditorías consume tiempo significativo dentro de la empresa y está propensa a errores humanos, lo que retrasa la identificación de desviaciones y limita la capacidad de los líderes para tomar decisiones informadas en tiempo real.

Automatizar las auditorías por capas, no solo permitirá reducir el tiempo empleado en la recolección de información, sino que también facilitará el seguimiento de las acciones correctivas y preventivas que se deriven de los hallazgos durante las mismas. Esto resultará en una mayor agilidad para implementar mejoras en los procesos y así aumentar la efectividad, asegurando el cumplimiento de los estándares establecidos.

Además, la automatización fortalecerá la retroalimentación entre los distintos niveles de la organización, involucrando de manera activa a gerentes y supervisores en el monitoreo de los procedimientos. Esto fomentará una cultura de mejora continua, ya que permitirá la detección temprana de problemas y la rápida corrección de estos, impactando positivamente en la productividad, seguridad y calidad operativa.

En resumen, el sistema automatizado es una solución indispensable para optimizar la gestión de auditorías, mejorar el rendimiento organizacional y garantizar que la empresa mantenga los más altos estándares en todos los niveles.

* 1. Objetivos

Crear un sistema automatizado de auditorías por capas que ayuden a capturar, controlar y monitorear información relacionada con el cumplimiento de los procedimientos de seguridad, calidad y productividad en los diversos niveles de la organización, tanto para gerentes como supervisores que detecten y corrijan desviaciones en los procesos.

* **Implementar un sistema de auditorías visual automatizado** que permita a los usuarios realizar auditorías de manera eficiente y reducir el tiempo de captura de información.
* **Facilitar el seguimiento de los hallazgos** detectados durante las auditorías y proporcione un mecanismo que permita a los líderes seguir el curso de acción necesario para corregir errores.
* **Optimizar la retroalimentación** en todos los niveles de la organización para que los gerentes y supervisores puedan tomar decisiones rápidamente.
* **Garantizar el cumplimiento de los procedimientos existentes** en áreas clave como seguridad, calidad y productividad, permitiendo la identificación temprana de errores y la implementación de mejoras continuas.
* **Reducir el tiempo y esfuerzo y análisis de datos** generados por las auditorias requerido para administrar y analizar los datos producidos por las auditorías, facilitando la generación de reportes y decisiones más informadas.
  1. Problema por resolver

Explicar con detalles cual es la situación actual o las problemáticas que esto enfrenta

La realización manual de auditorías por capas en la organización consume un tiempo considerable en la captura de información, lo que dificulta el seguimiento de los resultados de manera efectiva y precisa en diferentes momentos del proceso de seguridad, calidad y producción. Además, la falta de automatización limita la capacidad de los gerentes y supervisores para monitorear de manera eficiente el cumplimiento de los procedimientos de los procedimientos, lo que resulta en retrasos para la implementación de correcciones correctivas y la mejora continua para corregir y mejorar el proceso continuo.

* 1. Alcances y limitaciones

Tiempo

Versión de PHP

Capítulo 2

Marco teórico

* 1. Definición de Sistema WEB.

Un sistema web se puede definir como una plataforma tecnológica que opera en un ambiente de internet o en algunos casos localmente y posee la habilidad de proporcionar servicios o características a los usuarios mediante navegadores web. Estos sistemas pueden incluir desde sencillos sitios remotos hasta aplicaciones sofisticadas que necesitan procesamiento tanto en el servidor como en el cliente. Los sistemas web desempeñan un rol vital en la vida cotidiana, simplificando desde el acceso a datos hasta la ejecución de actividades complicadas como las compras en línea, la administración de empresas o las plataformas de educación.

### Elementos de un sistema en WEB.

Un sistema web común consta de tres capas fundamentales:

• Cliente (Frontend): Esta capa es la plataforma con la que el usuario interactúa. Se crean empleando tecnologías como HTML, CSS y JavaScript, las cuales se utilizan para su desarrollo. La cual permite crear vistas atractivas e interactivas.

• Servidor (Backend): La capa del servidor es el lugar donde se administran las operaciones lógicas y el manejo de información. Emplea tecnologías como PHP, JAVA, PYTHON, RUBY, etc. Las cuales funcionan para manejar las peticiones del cliente y proporcionar las respuestas adecuadas.

• Base de datos: En este lugar se guarda toda la información del sistema, que incluye datos de usuarios, operaciones o contenido multimedia. Los sistemas web suelen utilizar bases de datos como MySQL, PostgreSQL, MongoDB, etc.

### Funcionamiento de un Sistema WEB.

La operación de un sistema web se fundamenta en la interacción entre el usuario y el servidor. Cuando un usuario hace una petición mediante su navegador (por ejemplo, introduciendo una URL o enviando un formulario), esta petición es transmitida al servidor por medio del método GET, POST, PUT, etc. El cual la procesa y proporciona una respuesta. Generalmente, esta interacción se administra a través del protocolo HTTP o HTTPS.

### Tipos de sistema WEB

* Sistemas informáticos: Estos están elaborados para suministrar datos tanto estáticos como dinámicos a los usuarios. Un caso típico son los blogs o las páginas de noticias.
* Sistemas de operaciones: Facilitan a los usuarios la realización de operaciones más complejas, tales como adquisiciones en línea por ejemplo mercado libre, Amazon, etc. operaciones de banca.
* Sistemas de cooperación: Promueven la interacción entre diversos usuarios, tales como plataformas de redes sociales o de administración de proyecto.
* Sistemas de Negocios o gestión empresarial: Son programas web destinados a administrar procesos internos de una compañía, tales como CRM (Administración de Relaciones con Clientes), ERP (Planificación de Recursos Empresariales), y otros sistemas de administración de recursos y relaciones.
* Sistemas WEB: Son sistemas interactivos complejos que funcionan en el navegador. Se utilizan para funciones avanzadas como sistemas de gestión, comercio electrónico, redes sociales, etc. Se desarrollan con frameworks.

Existen más sistemas los cuales se ocupan dependiendo del área especifica, en este caso y para este proyecto nos enfocaremos en el último punto puesto que el sistema que se está desarrollando para que se maneje por medio de una URL y servirá para gestionar las peticiones que realicen por parte del frontend, a la lógica del backend y se consumirán datos o guardaran en la base de datos.

## Auditoria por Capas

La auditoría por capas o LPA (Layered Process Audit) es un proceso que se utiliza mucho en los sectores industriales, especialmente en la manufactura o producción, para garantizar que los procesos se sigan correctamente en cada nivel de la organización. Su objetivo es detectar fallas de manera temprana, asegurando que se adhieran a los estándares de calidad, productividad, seguridad, etc. Llevan el nombre de auditorías por los distintos niveles jerárquicos o en este caso llamados “capas” dentro de la industria o empresa. Haga clic en Insertar y elija los elementos que desee de las distintas galerías. Los temas y estilos también ayudan a mantener su documento coordinado. Cuando haga clic en Diseño y seleccione un tema nuevo, cambiarán las imágenes, gráficos y gráficos SmartArt para que coincidan con el nuevo tema. Al aplicar los estilos, los títulos cambian para coincidir con el nuevo tema. Ahorre tiempo en Word con nuevos botones que se muestran donde se necesiten.

### Concepto de Auditoria por Capas.

La auditoría de niveles requiere que los empleados de diversos niveles, desde trabajadores hasta directivos o gerentes, auditen el mismo proceso o área en periodos de tiempo determinados. Esto posibilita que cada nivel posea una visión única, garantizando que las comprobaciones sean rigurosas y se identifiquen con rapidez las diferencias que puedan perjudicar la calidad o la seguridad del procedimiento y de esa manera poder tomar decisiones eficazmente.

### Ejemplo de Auditoria por Capas.

* Primer nivel: Auditores de nivel de producción, como por ejemplo los encargados de línea, que llevan a cabo revisiones regulares diarias o semanales sobre los procesos bajo su supervisión y de esta manera encontrar posibles errores para su mejora.
* Segundo panel: directores o encargados de área, capaces de llevar a cabo auditorías semanales o mensuales para garantizar que los supervisores cumplen con los estándares.
* Tercer nivel: Supervisores o auditores externos, que examinan de manera regular el proceso global para identificar problemas sistémicos o fallos en el control.

### Beneficios de las auditorias por capas.

* Identificación errores a temprana edad: Gracias a las auditorías realizadas en distintos niveles, es posible identificar y solucionar rápidamente los problemas.
* Obligación en Todos los Escalones: Al involucrar a trabajadores de todas las áreas, se promueve la responsabilidad colectiva y el compromiso con la excelencia operativa de todas las áreas.
* Implementación de Normas: Mejora la conformidad con los criterios de calidad y seguridad y otras áreas a través de una revisión continua.
* Avance Constante: Las revisiones continuas generan una cultura de perfeccionamiento constante, en la que los trabajadores prestan mayor atención a los pormenores de su labor y así evitar fallas.

## Mockups

Un **mockup** es una herramienta de diseño que permite maquetar y plasmar visualmente un concepto, facilitando la comunicación de la idea del producto final. Su propósito principal es asegurar que el diseño esperado sea el entregado, evitando malentendidos en la fase de desarrollo. Los mockups son representaciones estáticas de las pantallas clave de un sistema y constituyen un paso fundamental para validar la experiencia de usuario (UX) antes de iniciar el desarrollo técnico.

### Herramientas utilizadas

En la actualidad, existen diversas herramientas, tanto gratuitas como de pago, que facilitan el desarrollo de interfaces gráficas, ya sea para sistemas web o áreas de marketing. Estas herramientas permiten optimizar el proceso de creación de vistas con el objetivo de hacer el trabajo más eficiente.

Inicialmente, los diseños de los mockups se realizaron en papel, lo que permitió a los interesados visualizar el flujo de las pantallas y aportar sus opiniones y recomendaciones para mejorar la usabilidad.

#### Excalidraw.

Es una herramienta que permite crear diagramas de manera sencilla, intuitiva y colaborativa. Se utiliza frecuentemente para crear wireframes, esquemas, y diagramas de flujo. Excalidraw se destaca por su capacidad para realizar bocetos visuales de forma rápida, facilitando la creación de prototipos preliminares. En este caso, se utilizó para digitalizar los diseños realizados en papel, permitiendo también la utilización de una paleta de colores para ofrecer una vista más realista de la interfaz final.

## Base de datos

Una base de datos es donde se organiza la información o datos a los cuales después se pueden ingresar para administrar o actualizar. Las bases de datos son una rama importante en las aplicaciones de informática como por ejemplo sistemas web o móviles puesto que gracias a que las bases de datos se pueden acumular y recuperar grandes cantidades de información de manera eficiente y así poder utilizarlos cuando el usuario lo necesite.

Existen administradores de bases de datos (DBMS) como MySQL, PostgreSQL y SQLite, las cuales son bases de datos estructuradas, por otro lado, existen las bases de datos NoSQL las cuales como por ejemplo MongoDB, Casandra, etc. Los cuales facilitan las operaciones de administrar, insertar, actualizar y eliminar información. Además, existen herramientas y entornos de desarrollo como XAMPP, WAMP, Laragon y MAMP, que facilitan a los programadores el uso de bases de datos en servidores locales.

### Tipos de Bases de datos

Existen diversos tipos de bases de datos lo cuales se utilizan ampliamente según el tipo de dato y las necesidades del usuario. A continuación, se mostrará una tabla sobre las bases de datos relacionales y no relacionales.

|  |  |  |
| --- | --- | --- |
| Característica | Bases de Datos Relacionales (RDBMS) | Bases de Datos No Relacionales (NoSQL) |
| Estructura de Datos | Tablas con filas y columnas | Diversos modelos (clave-valor, documentos, grafos, columnas) |
| Lenguaje de Consulta | SQL (Structured Query Language) | Varía según el modelo (por ejemplo, JSON para MongoDB, Gremlin para grafos) |
| Esquema | Esquema fijo y estructurado | Flexible y dinámico |
| Escalabilidad | Escalabilidad vertical (agregar más potencia al servidor) | Escalabilidad horizontal (agregar más nodos o servidores) |
| Transacciones | Soporta transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) | Soporta transacciones eventualmente consistentes (no todas soportan ACID) |
| Casos de Uso | Aplicaciones financieras, ERP, sistemas donde la integridad de datos es crítica | Big Data, aplicaciones en tiempo real, redes sociales, IoT |
| Ejemplos | MySQL, PostgreSQL, Oracle Database | MongoDB, Cassandra, Neo4j, Redis |
| Almacenamiento de Datos | Datos estructurados | Datos no estructurados y semi-estructurados |

### Tipos de Gestores de Base de Datos.

Como se mencionó anteriormente existen diversos tipos de bases de datos, pero también existen diversos tipos de herramientas para administrarlas. A continuación, se muestran algunos de ellos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Características | XAMMP | LARAGON | WAMP | MAMP | EASYPHP |
| Compatibilidad de Sistemas | Windows, macOS, Linux | Windows | Windows | macOS, Windows | Windows |
| Componentes Incluidos | Apache, MySQL, PHP, Perl | Apache, MySQL/MariaDB, PHP, Node.js | Apache, MySQL, PHP | Apache, MySQL, PHP | Apache, MySQL, PHP |
| Instalación | Sencilla | Muy sencilla, portátil | Sencilla, pero está limitada a Windows | Muy sencilla | Sencilla |
| Facilidad de Uso | Interfaz simple | Intuitiva y amigable | Interfaz de usuario | Intuitiva | Interfaz sencilla |
| Portabilidad | No es portable | Portable | No es portable | No es portable | Portable |
| Gestión de Versiones | Limitada | Soporta múltiples versiones | Limitada | Limitada | Limitada |
| Virtual Hosts Automáticos | No | Si | Si | Limitado | No |
| Soporte SSL | Limitada | Si, muy sencillo | Alta | Popular en MacOS | Limitada |
| Popularidad | Alta | Mediana, en crecimiento | Sí | No | Media |

## LARAGON

Laragon es un recurso potente para equipos técnicos, diseñado para facilitar la creación de diversos entornos de desarrollo, lo que simplifica enormemente la labor con aplicaciones web y locales. Esta herramienta es ideal para desarrolladores que buscan configurar rápidamente entornos listos para trabajar, ya que incluye un servidor ligero y rápido, optimizado para el desarrollo además como se mencionara a continuación su importancia.

Con Laragon, tendremos la posibilidad de trabajar con lenguajes de programación de servidor como NodeJS, Ruby, Python y, naturalmente, PHP esto por mencionar alguno de los frameworks más utilizados e importantes. Además de estos lenguajes, Laragon soporta múltiples bases de datos como MySQL, MariaDB, PostgreSQL y SQLite, MongoDB, etc. Lo que brinda flexibilidad en el desarrollo de aplicaciones de diversa naturaleza y para diferentes ámbitos dependiendo del campo en el que se trabaje.

Una de las principales ventajas de Laragon es su facilidad de uso. Además de que es portable y de escritorio, como se mencionó en la tabla anterior de los diferentes gestores de bases de datos y para desarrollo web, pero de todos los antes mencionados Laragon cumplía con los estándares gracias a que es compatible con diferentes versiones de PHP, soporta diferentes tecnologías, genera los “virtual- host” automáticamente entre otras cosas que facilitan su uso.

## Que es un FRAMEWORK

Un **framework** es una estructura o conjunto de herramientas que proporciona una base estandarizada para el desarrollo de software. Los frameworks contienen código predefinido que los desarrolladores pueden usar y ampliar para crear aplicaciones, reduciendo el tiempo y el esfuerzo necesario para programar desde cero. Además, un framework ofrece convenios o estándares de cómo organizar el código creando buenas prácticas y así facilitar el mantenimiento y seguridad de aplicaciones a largo plazo.

Los frameworks se utilizan en diferentes áreas de programación incluyendo el desarrollo WEB, móvil o de escritorio estos varían su complejidad y intención.

### Ventajas de usar un FRAMEWORK.

* **Eficiencia:** Al proporcionar componentes y archivos ya creados y configuradospermiten acelerar en grandes cantidades el tiempo de desarrollo. Por ejemplo, no es necesario desarrollar el código desde cero para gestionar las solicitudes HTTP o conexiones a bases de datos pues el framework se encargará de eso.
* **Mantenimiento:** Un framework establecerá reglas sobre cómo desarrollar el código para que de esta manera se obtenga un código mas completo y la colaboración entre diferentes desarrolladores.
* **Reutilización de Código:** Al utilizar componentes y módulos ya definidos, se evita la duplicación de código en algunos segmentos del código, las personas que desarrollen sobre el código pueden utilizar o personalizar algunos segmentos del código para soluciones más completas.
* **Seguridad:** La mayoría de los frameworks contiene características de seguridad integradas, todo esto con el fin de brindar protección contra inyecciones SQL, o cualquier otro tipo de ataque y gestionar la autenticación.

## CODEIGNITER 4

**CodeIgniter 4** es un framework de PHP diseñado para ofrecer a los desarrolladores una base sólida, ágil y liviana sobre la cual edificar aplicaciones web. Es conocido por su simplicidad, bajo consumo de recursos y rendimiento rápido, lo que lo convierte en una opción popular entre los desarrolladores que buscan una solución eficaz para proyectos de cualquier tamaño.

Una de las razones por las que **CodeIgniter 4** es tan utilizado es porque sigue siendo fiel a su filosofía original de proporcionar un marco mínimo y modular, que permite a los desarrolladores añadir solo las herramientas y bibliotecas que necesiten. Esto ayuda a mantener el código limpio y fácil de mantener, lo que es crucial en proyectos a largo plazo .

Además, **CodeIgniter 4** se destaca por su curva de aprendizaje baja. Para aquellos que ya están familiarizados con PHP, la transición a CodeIgniter es sencilla, lo que hace que sea accesible tanto para desarrolladores novatos como para expertos. A diferencia de otros frameworks más complejos, CodeIgniter te permite empezar rápidamente, sin configuraciones complicadas ni la necesidad de aprender patrones de diseño avanzados antes de ser productivo.

Otra razón por la cual se utiliza es su excelente rendimiento. CodeIgniter está optimizado para ser rápido, incluso en servidores de bajo costo, lo que lo convierte en una opción ideal para aplicaciones web que requieren respuestas rápidas sin consumir demasiados recursos. Su núcleo liviano permite que las aplicaciones funcionen de manera eficiente sin sobrecargar los servidores .

**CodeIgniter 4** también es utilizado por su soporte para una arquitectura MVC (Modelo-Vista-Controlador), lo que fomenta la separación del código y hace que las aplicaciones sean más fáciles de escalar y mantener. Esta arquitectura permite organizar el código de una manera estructurada, lo que simplifica el trabajo en equipo y facilita el crecimiento del proyecto con el tiempo.

Además, **CodeIgniter 4** incorpora mejoras significativas en comparación con sus versiones anteriores, incluyendo soporte para PHP 7.3 o superior, una mejor gestión de rutas, y herramientas más avanzadas para la depuración y el manejo de errores, lo que lo convierte en una opción sólida para el desarrollo de aplicaciones modernas.

## PHP

**PHP (Hypertext Preprocessor)** es un lenguaje de programación de propósito general, ampliamente utilizado para el desarrollo web. Fue creado en 1994 por Rasmus Lerdorf y, a lo largo de los años, ha evolucionado hasta convertirse en uno de los lenguajes más populares en el ámbito del desarrollo web y para mi uno de los mejores lenguajes, especialmente para el desarrollo del lado del servidor. Su popularidad proviene en gran parte de su simplicidad, maleabilidad y capacidad para integrarse fácilmente con HTML, bases de datos y otros lenguajes.

### ¿Por qué PHP?

1. **Facilidad de uso**: PHP es relativamente sencillo de aprender y escribir. Su sintaxis es clara y accesible, lo que lo convierte en una opción popular tanto para desarrolladores novatos como experimentados.
2. **Desarrollo rápido**: PHP permite a los desarrolladores crear aplicaciones web dinámicas de manera rápida. Gracias a su capacidad de integrarse fácilmente con HTML y otras tecnologías web como CSS y JavaScript, los desarrolladores pueden crear aplicaciones completas en menos tiempo. Además de su adaptabilidad de hacia diferentes gestores de bases de datos.
3. **Código embebido en HTML**: Uno de los puntos fuertes de PHP es que puede insertarse directamente en el código HTML, lo que permite a los desarrolladores alternar entre HTML y PHP en una misma página sin complicaciones. Esta capacidad hace que PHP sea extremadamente flexible y fácil de usar para crear contenido web dinámico, puntos de venta, desarrollos hacia empresas, etc.
4. **Compatibilidad multiplataforma**: PHP es compatible con la mayoría de los sistemas operativos, incluidos Windows, Linux y macOS. Esto le permite ser ejecutado en una variedad de servidores y entornos, lo que le otorga flexibilidad para desarrolladores y empresas que trabajan con diferentes plataformas. Además, puede integrarse con varios servidores web como Apache y Nginx sin mayores inconvenientes.
5. **Rendimiento**: PHP ha mejorado significativamente su rendimiento a lo largo de los años, especialmente con las versiones más recientes como PHP 7.x y PHP 8.x, que ofrecen mejoras en la velocidad de ejecución y optimizaciones en el consumo de memoria. Esto lo hace ideal para aplicaciones que manejan grandes volúmenes de tráfico.
6. **Código abierto**: PHP es de código abierto, lo que significa que cualquier persona puede usarlo, modificarlo y distribuirlo de forma gratuita. Esto lo ha convertido en una opción preferida para startups, pequeñas empresas y desarrolladores independientes que buscan una solución económica para el desarrollo web.

## MVC

Modelo-Vista-Controlador es un patrón o arquitectura de diseño de software que se utiliza comúnmente en aplicaciones WEB. Este patrón organiza el código en tres grandes componentes principales: Modelo, Vista y Controlador, permitiendo una clara separación de responsabilidades dentro de la aplicación. Este enfoque mejora la organización, fascilita el manejo del código y permite la escalabilidad del proyecto con el paso del tiempo.

### ¿Qué significa MVC?

* **Modelo:** El modelo representa la parte de la aplicación donde maneja la lógica relacionada con los datos (consultas SQL). Es responsable de interactuar con la base de datos y como deben de ser manipulados. El modelo se define como se estructuran los datos y como deberían ser manipulados. Por ejemplo, si estamos construyendo un punto de venta, el modelo se encarga de acceder a la base de datos para gestionar, obtener, actualizar entre otras funciones para los usuarios dependiendo de lo que necesite.
* **Vista:** La vista es parte de la aplicación que gestiona la presentación de la información. En otras palabras, es la interfaz de usuario (UI) que los usuarios ven e interactúan. La vista toma datos proporcionados por el modelo y los muestra en un formato que los usuarios pueden entender, Siguiendo con el ejemplo del punto de control, la vista es donde se muestran precios e inventarios de los productos.
* **Controlador:** El controlador actúa como intermedio entre el modelo y la vista. Su tarea es recibir solicitudes del usuario a través de una URL o una acción en especifica, interactúan con el modelo para procesar los datos que se traen y luego pasarlos a la vista dinámicamente, nuevamente como ejemplo en el punto de venta el controlador se basa de gestionar las solicitudes de los usuarios, buscar, seleccionar algún producto en coordinación con la vista y el modelo en consecuencia se obtienen buenos resultados.

### Porque MVC

MVC es una arquitectura para desarrollar software mas utilizado en el area de programación WEB puesto que los tres módulos (Modelo-Vista y controlador) son independientes y esto conlleva a grandes ventajas como.

* **Se separan las responsabilidades:** Al dividir el codigo en tres bloques es fácil deliberar responsabilidades a los desarrolladores para que cada quien pueda trabajar independientemente por ejemplo alguien podría trabajar en el backend mientras que otros en el frontend.
* **Mantenimiento:** Al tener separado cada módulo, y sobre cada modulo sus funciones cuando se le necesite dar alguna actualización esta se podrá desarrollar efectivamente puesto que el codigo estará organizado y separado por secciones y a largo plazo quien desee realizar cambios dentro del sistema lo realizara sin ningún problema. Por ejemplo, si se desea cambiar algo e la vista esto se podrá hacer sin modificar algo dentro del modelo o el controlador.
* **Escalabilidad:** La arquitectura MVC permite que las aplicaciones crezcan sin perder su claridad en la estructura del codigo, además si se utiliza un framework como lo que es CodeIgniter, Laragon, NodeJS, etc. Se podrá mejorar aún más sin perder claridad y si se busca que el sistema contenga nuevas funcionalidades o mejoras se podrá realizar sin mayor problema.
* **Desarrollo en equipo:** El patrón MVC se puede desarrollar además de facilitar el trabajo en equipo puesto que gracias a que se desarrolla en módulos se peuden centrar en diferentes áreas del proyecto, algunos en el backend otros en la lógica de las consultas, etc.

## JAVA SCRIPT

Java Script es un lenguaje de programación dinámico que se utiliza principalmente en el desarrollo para páginas que son dinámicas o interactivas y aplicaciones web modernas. Originalmente este lenguaje fue desarrollado para trabajar del lado del cliente (Servidor WEB), pero con el paso del tiempo también se ha pasado del lado del servidor. JavaScript es muy conocido por su flexibilidad, su capacidad para interactuar con el Document Obejct Model o por sus siglas (DOM) y su amplia adaptación a aplicaciones WEB o móviles.

### Características principales de Java Script

* **Lenguaje Interpretado:** Se ejecuta directamente en el navegador sin la necesidad de un proceso de compilación anterior. Esto lo convierte en un lenguaje rápido para desarrollar y probar código en tiempo real asi facilitando las pruebas.
* **Interactividad WEB**: Es el lenguaje principal utilizado para hacer que las páginas web sean interactivas. Permite crear menús desplegables, formularios dinámicos, Tablas dinámicas, validación de entradas en el cliente, galerías de imágenes y mucho más, sin tener que recargar la página se podría decir que ejecuta código por secciones.
* **Compatibilidad con diversos navegadores**: JavaScript está soportado por todos los navegadores modernos, lo que significa que el código puede ser ejecutado en cualquier dispositivo sin importar el sistema operativo. Es compatible con Google Chrome, Firefox, Safari, Edge, entre otros.
* **Librerías:** Java Script tiene un ecosistema enorme de librerías y que constantemente se mejoran con el paso del tiempo. Existen librerías que van de la mano con frameworks para el desarrollo de interfaz de usuario las cuales han ampliado enormemente este lenguaje.
* **JSON (JavaScript Object Notation):** JavaScript también es conocido por ser la base del formato JSON, que es ampliamente utilizado para intercambiar datos entre el cliente y el servidor a través del controlador de manera eficiente. JSON es fácil de entender, ligero. Este a ganado mucha popularidad a lo largo del tiempo por su forma tan eficiente de utilizar.

### Java Script en el Frontend

JavaScript es la columna vertebral de la interactividad en las páginas web. Se utiliza para manejar eventos del usuario, manipular el DOM, y crear aplicaciones web dinámicas que pueden actualizarse sin recargar toda la página. Por ejemplo, formularios interactivos, menús dinámicos, y animaciones son comúnmente creados con JavaScript.



## HTML

HTML (HyperText Markup Language) es fundamental cuando se habla de sistemas WEB pues este sirve como base sobre la que se construye las interfaces de usuario y se estructura el contenido visual en el navegador. Se trata de un lenguaje de marcado que permite organizar textos, imágenes, botones, enlaces, entre otros elementos multimedia que permiten al usuario interactuar con la información de la página de manera eficiente y clara.

HTML contiene diferentes tipos de etiquetas y elementos que permiten definir estructuras y formato a la página WEB. Por ejemplo, colocar textos en colores diferentes títulos de diferentes tamaños, enlaces que nos puedan dirigir a una página en concreto, colocar imágenes, insertar cuadros de dialogo, etc. HTML va de la mano con la vista lo cual pueden ser enlaces estáticos pero como se menciono anteriormente con el modelo MVC podemos obtener un sistema dinámico el cual existe la oportunidad de que dependiendo de las características que el usuario necesite sobre su proyecto se pueden realizar cambios, retomando el ejemplo del punto de venta puede que tengamos una vista con la misma estructura HTML pero que presenta diferentes productos no es necesario tener ponerlos manualmente estos ya que se incrustan automáticamente sin la necesidad de modificar nada.

## CSS

**CSS (Cascading Style Sheets)** es un lenguaje utilizado para para describir la presentación, formato y diseño visual de las paginas WEB. La función principal de CSS es proporcionar como su nombre lo indica estilos y formato a los elementos de HTML por lo que permite que los desarrolladores y diseñadores creen paginas visualmente atractivas y bien organizadas. CSS muestra aspectos como colores, diferentes fuentes, tamaños de imágenes, vistas responsivas, márgenes y en generar diferentes diseños en la vista del usuario. Con esto se busca tener un codigo mas llamativo, organizado y fácil de mantener con el paso del tiempo.

### Que es CSS

CSS es una hoja de estilo en forma de cascara que va de la mano con HTML para dar un formato mas estructurado a la pagina WEB. Mientras que HTML define la estructura y el contenido, CSS se encarga de como se vera ese contenido en la pantalla de los dispositivos. Su importancia aplica en integrar múltiples elementos de una pagina esto con el fin de que sea llamativa, lo que facilita cambios rápidos en el diseño sin la necesidad de modificar la estructura HTML.

### Conceptos claves sobre CSS

A continuación, se muestran algunos conceptos clave sobre CSS y por qué.

* **Selectores de etiquetas:** Aplican estilos a todas las etiquetas especificas (por ejemplo, **p** para cada párrafo, **h1, h2, h3,** etc. Para los títulos).
* **Selectores de clases:** Aplica estilos a elementos con una clase en especifica, estos se utilizan utilizando el **“.”** antes de escribir el nombre de la clase, por ejemplo **.class** dentro del HTML y dentro de la hoja de estilos puedes agregarle el formato que desees.
* **Selector de ID:** Aplica estilos a un elemento con un ID único, usando el símbolo **“#”** antes de ID a utilizar por ejemplo **#Iduser.**
* **Selectores combinados:** Estos funcionan para definir estilos avanzados y dinámicos como: hover(cuando el ratón pasa sobre un elemento) entre otros elementos para darle más diseño al sistema.

En conclusión, las hojas de estilo son de suma importancia en el desarrollo WEB ya que gracias a esto podemos crear sistemas responsivos a lo que el usuario necesita además de que los estilos son adaptativos a cualquier dispositivo esto con el fin de tener una mejor eficiencia además de que es versátil y además de que se continua desarrollando.

## VISUAL STUDIO CODE

Visual Studio Code es un software de editor de código gratuito, ligero y multiplataforma (funciona para Windows, MacOS y Linux) este software fue desarrollado por Microsoft y lanzado en 2015, se ha convertido rápidamente en una herramienta de suma importancia y populares ya que soporta diferentes tipos de lenguajes de códigos además de su flexibilidad, rendimiento, y su gran cantidad de extensiones disponibles que permiten que el desarrollo sea agradable y entendible. Este software no solo funciona para implementar la programación también funciona como depurador, control de versiones, terminal integrada, entre mas características que hace que el trabajo sea fácil, ágil durante el proceso de desarrollo.

### ¿Por qué Visual Studio Code?

Visual Studio Code se ha convertido en una herramienta imprescindible para desarrolladores de cualquier nivel y en prácticamente cualquier lenguaje de programación. Una de sus mayores ventajas es que permite organizar y gestionar proyectos completos desde una única interfaz, ofreciendo la capacidad de explorar todas las carpetas y archivos del sistema con facilidad. Esto ayuda a mantener una visión clara del proyecto, independientemente de su tamaño.

Además, Visual Studio Code permite visualizar los cambios en tiempo real, lo que es especialmente útil al trabajar en proyectos web o aplicaciones donde es necesario ver cómo se ven y funcionan los cambios al instante. Desde el mismo editor, también se pueden crear y organizar nuevos directorios y archivos, facilitando la estructura del proyecto sin necesidad de recurrir a herramientas externas.

Una de las características que hace destacar a VS Code es su amplio ecosistema de extensiones. Estas extensiones permiten añadir funcionalidades que ayudan a mejorar la calidad del código, como detectar errores de sintaxis, sugerencias para optimizar el rendimiento y recomendaciones de formato. Estas herramientas ayudan a los desarrolladores a corregir problemas rápidamente y a seguir buenas prácticas de desarrollo, lo que mejora la eficiencia y calidad del proyecto

## HEIDI SQL

HeidiSQL es una herramienta gratuita y de código abierto para la gestión y administración de bases de datos. Es especialmente popular entre desarrolladores y administradores de bases de datos que necesitan una solución intuitiva y completa para trabajar con sistemas de gestión de bases de datos como **MySQL**, **MariaDB**, **Microsoft SQL Server**, **PostgreSQL**, y **SQLite**. Desarrollada inicialmente como un cliente para MySQL, HeidiSQL se ha expandido a lo largo de los años para incluir soporte a otros motores, lo que la convierte en una opción versátil y ligera para la administración de bases de datos.

### Características de HeidiSQL.

* **Plataforma sencilla e intuitiva:** HeidiSQL sobresale por su interfaz simple y de fácil manejo. Los programadores y gestores tienen la habilidad de desplazarse con rapidez entre tablas, bases de datos, vistas y otros elementos, sin requerir conocer lo suficiente del software. Su diseño gráfico facilita la comunicación con las bases de datos, incluso para quienes empiezan a utilizar este software y sus herramientas que incluye.
* **Soporte para múltiples motores de bases de datos**: Aunque fue diseñado originalmente para MySQL, HeidiSQL ahora acepta MariaDB, SQL Server, PostgreSQL y SQLite, lo que lo convierte en una herramienta de administración variable para aquellos que trabajan en diferentes ambientes.
* **Edición y visualización de datos en tablas**: HeidiSQL permite a los usuarios ver y editar datos de manera rápida y eficiente. Puedes editar el contenido de una tabla, añadir o eliminar registros y realizar consultas SQL directamente desde la interfaz, lo cual facilita las tareas de mantenimiento y ajuste sin la necesidad de ejecutar el navegador para visualizar la base de datos.
* **Emisión e importación de información:** La herramienta posibilita la exportación de bases de datos completas o tablas concretas a diversos formatos, tales como SQL, CSV, XML y HTML, lo que simplifica la transferencia de datos o la generación de copias de seguridad. Además, facilita la importación de datos en varios formatos, un recurso esencial para la incorporación de bases de datos en proyectos de gran envergadura.
* **Ejecución de múltiples consultas SQL**: HeidiSQL permite ejecutar múltiples consultas SQL en una misma sesión. Los usuarios pueden ver los resultados de cada consulta en pestañas separadas, lo que facilita el análisis de datos y la creación de reportes sin necesidad de abrir múltiples ventanas o cambiar de herramienta.

## NAVICAT

Navicat, junto con HeidiSQL, es un potente gestor de bases de datos fácil de utilizar, que permite desarrollar y administrar bases de datos de manera eficaz desde una única interfaz. Es compatible con una amplia variedad de sistemas de gestión de bases de datos, incluyendo MySQL, MariaDB, MongoDB, PostgreSQL, SQLite, Oracle y Microsoft SQL Server. Además, Navicat ofrece un gran soporte para la visualización de diagramas entidad-relación, lo que facilita la comprensión y diseño de la estructura de la base de datos. Durante el desarrollo del sistema, Navicat resultó ser una herramienta fundamental, ya que facilitó tanto el diseño como la administración de las bases de datos.

## GIT

GIT es un sistema de control de versiones distribuido, esta herramienta fue diseñada para rastrear y gestionar cambios en el código de proyectos de software. Git se ha convertido en una herramienta fundamental para desarrolladores y equipos de software de todo el mundo. Su fama se debe a su capacidad para gestionar proyectos de manera eficients y colaborativa, permitiendo a varios desarrolladores trabajar en el mismo proyecto din interferir o chocar con el trabajo de los demás desarrolladores.

### ¿Qué es Git y cómo funciona?

Git permite a los desarrolladores realizar un seguimiento de los cambios en su código, volver a versiones anteriores, y trabajar en diferentes funcionalidades o "ramas" como se llama normalmente a los cambios del proyecto de forma independiente. Esto resulta especialmente útil en proyectos grandes donde múltiples colaboradores están modificando el código al mismo tiempo. Git no solo facilita la colaboración, sino que también permite mantener un historial completo de todos los cambios realizados, lo que es invaluable para depuración, auditoría y recuperación de código en caso de errores.

### Características principales de GIT

* **Distribuido**: Git es un sistema de control de versiones distribuido, lo que significa que cada desarrollador tiene una copia completa del repositorio de código, incluido todo el historial de cambios, en su propia máquina. Esto permite trabajar sin conexión y asegura que los datos no dependan de un servidor central.
* **Control de versiones**: Git guarda un historial de todos los cambios realizados en el proyecto, lo que permite a los desarrolladores regresar a una versión anterior en cualquier momento. Esto es útil para depurar errores o deshacer modificaciones no deseadas.
* **Ramas y fusiones (branches y merges)**: Git permite crear ramas o "branches" para trabajar en funcionalidades, correcciones de errores o experimentos sin afectar el código principal. Las ramas pueden fusionarse con el proyecto principal una vez que están listas, lo que facilita el trabajo en equipo y el desarrollo de nuevas características sin interrupciones.
* **Colaboración y trabajo en equipo**: Git permite a múltiples desarrolladores trabajar en el mismo proyecto simultáneamente. Las funcionalidades de branching que permite que los colaboradores trabajen en sus propias ramas, lo que minimiza los conflictos y facilita la integración del código.
* **Comandos de Git**: Git ofrece una gran variedad de comandos para gestionar el código, como **commit** (guardar cambios), **push** (subir cambios a un repositorio remoto), **pull** (traer los cambios del repositorio remoto), **clone** (clonar un repositorio), entre otros. Estos comandos son esenciales para manipular y gestionar el código en Git.
* **Repositorios remotos**: Git permite trabajar con repositorios remotos, alojados en plataformas como **GitHub**, **GitLab** y **Bitbucket**, lo que facilita el acceso y la colaboración de equipos distribuidos. Estas plataformas ofrecen además herramientas adicionales para gestión de proyectos, como la creación de issues, wikis y revisiones de código.

### Ventajas de GIT

* **Historial detallado**: Git guarda un historial completo de todos los cambios en el código, lo que facilita la revisión, el deshacer errores y la restauración de versiones anteriores.
* **Eficiencia para trabajo en equipo**: Git facilita la colaboración en proyectos de gran escala, permitiendo a los desarrolladores trabajar en diferentes ramas y fusionar cambios sin conflictos importantes.
* **Desempeño rápido**: Al ser distribuido, Git permite trabajar en el proyecto y realizar commits sin conexión a internet, haciendo que el trabajo local sea rápido y eficiente.

## GITHUB

**GitHub** es una plataforma de desarrollo y colaboración basada en la web que utiliza **Git**, el sistema de control de versiones distribuido, para facilitar la gestión de proyectos y el trabajo en equipo. GitHub permite a los desarrolladores almacenar, compartir y colaborar en el código de manera agrupada, lo que es fundamental para proyectos de software, especialmente cuando involucran a múltiples colaboradores. Además de sus capacidades de alojamiento de repositorios Git, GitHub ofrece una amplia gama de herramientas de gestión de proyectos, automatización y colaboración, lo que la convierte en una de las plataformas preferidas para desarrolladores y empresas de software en todo el mundo.

### GitHub VS GIT

Es importante mencionar que estas tecnologías no son lo mismo a continuación se menciona por qué.

1. **GitHub** es una plataforma basada en la web que permite alojar repositorios Git y colaborar en línea. GitHub facilita compartir el código y colaborar con otros, además de ofrecer herramientas de gestión de proyectos como issues, pull requests, y revisiones de código. Otras plataformas similares a GitHub incluyen **GitLab** y **Bitbucket**.
2. **Git** es la tecnología de control de versiones que se ejecuta en tu máquina local, permitiendo gestionar el historial de cambios de tu proyecto.

### Características principales de GITHUB

* **Repositorios**: En GitHub, el código de un proyecto se almacena en un **repositorio** (o repo), que actúa como un contenedor para todos los archivos y el historial de cambios del proyecto. Los repositorios pueden ser **públicos** (accesibles para cualquier persona) o **privados** (accesibles solo para usuarios autorizados).
* **Branches (Ramas)**: Al igual que Git, GitHub permite trabajar en **ramas** o "branches" del proyecto. Las ramas son útiles para desarrollar nuevas características, corregir errores o realizar experimentos sin afectar el código principal. Cada rama se puede fusionar con el código principal mediante pull requests una vez que esté lista.
* **Pull Requests (PR)**: Los pull requests son una de las características más poderosas de GitHub. Permiten a las personas que desarrollan algún sistema proponer cambios en el código desde una rama, y que otros miembros del equipo pueden revisar, comentar y aprobar los cambios antes de que se fusionen con la rama principal del proyecto. Los PR son esenciales para mantener el código limpio y de alta calidad.

### Ventajas de usar GITHUB

* **Facilita la colaboración en equipo**: GitHub es ideal para trabajar en equipo gracias a herramientas como pull requests, issues y revisiones de código, que organizan y simplifican la colaboración.
* **Visibilidad y control**: Los desarrolladores pueden llevar un registro detallado de los cambios, ver el historial completo del proyecto y revisar el estado de las contribuciones, lo que mejora el control y la transparencia del proyecto.
* **Seguridad y control de acceso**: GitHub permite configurar repositorios públicos y privados y establecer permisos para proteger el código de accesos no autorizados.

### GITLENS

**GitLens** es una extensión para Visual Studio Code que mejora las capacidades de manejo de Git dentro del editor. Proporciona una serie de características avanzadas que facilitan la visualización y comprensión del historial de versiones de un proyecto.

#### ¿Qué hace GitLens?

**GitLens** extiende las funcionalidades de Git en el edito de código Visual Studio Code al ofrecer información detallada sobre el historial de cambios, autores, commits y ramas. A diferencia de las herramientas Git estándar, GitLens permite ver detalles de cada línea o bloque de código, así como que cambios se le realizan, lo cual resulta útil para comprender el contexto histórico y el flujo de trabajo del equipo. A continuación, se presentan algunas de las características principales de **GitLens.**

* **Anotaciones en línea (blame)**: GitLens permite ver quién fue el último en modificar cada línea de código directamente en el editor. Esta función, conocida como "blame", muestra el autor, la fecha y el mensaje del commit de cada línea, lo que facilita el rastreo de cambios y la identificación de posibles errores.
* **Explorador de historial y commits**: Con GitLens, puedes explorar el historial completo del repositorio, ver todos los commits y analizar los cambios realizados en cada uno. Esta vista de historial incluye detalles del commit, los archivos afectados, y las modificaciones específicas en cada archivo, lo que permite entender cómo ha evolucionado el código.
* **Comparación de ramas y commits**: GitLens permite comparar fácilmente entre diferentes ramas y commits, mostrando las diferencias entre versiones del código. Esto es útil para ver cómo una rama se diferencia de la principal (main/master) antes de realizar una fusión o para revisar los cambios entre versiones específicas.
* **Vista de cambios recientes**: La extensión muestra una lista de los archivos y cambios más recientes, permitiendo a los desarrolladores ver rápidamente las últimas modificaciones y revisar el progreso del proyecto.
* **Explorador de ramas y repositorios**: GitLens ofrece una vista completa de todas las ramas y etiquetas del repositorio, lo que facilita la navegación y permite a los usuarios cambiar de rama o realizar fusiones de forma rápida y sencilla.
* **Visualización de líneas de tiempo**: Con la línea de tiempo de GitLens, puedes ver una representación cronológica de los cambios en un archivo específico. Esto incluye una lista de commits y cambios en el archivo, mostrando la evolución del archivo desde su creación hasta el presente.
* **Anotaciones y comentarios en los commits**: GitLens permite visualizar los comentarios de cada commit y asociarlos con el código correspondiente, lo que facilita la revisión y comprensión de cada cambio. Además, puedes ver el impacto de un commit específico en diferentes archivos y líneas de código.

## FIREFOX DEVELOPMENT

**Firefox Developer Edicion** es una versión del navegador Firefox de Mozilla diseñada específicamente para desarrolladores web. Esta edición incluye herramientas avanzadas de desarrollo, características experimentales y soporte para estándares web emergentes, lo que permite a los desarrolladores crear, probar y depurar aplicaciones web de manera eficiente. Firefox Developer Edicion es gratuito y está utilizable en sistemas como Windows, macOS y Linux, ofreciendo un entorno optimizado para quienes trabajan en el desarrollo y depuración de aplicaciones web.

### Ventajas de Usar el navegador Firefox Developer Edicion

* **Ambiente de pruebas y desarrollo especializado**: Firefox Developer Edition proporciona un entorno controlado donde los desarrolladores pueden probar sus aplicaciones web con características experimentales. Esto permite explorar nuevas APIs y tecnologías que aún no están disponibles en los navegadores convencionales.
* **Optimización para el flujo de trabajo de desarrollo**: La interfaz de Firefox Developer Edition está diseñada para facilitar el acceso a herramientas de desarrollo. La disposición de los menús y la estructura del navegador ayudan a que las herramientas sean fáciles de encontrar y utilizar, lo cual optimiza el flujo de trabajo diario de los desarrolladores.
* **Enfoque en el rendimiento**: La versión Developer Edition de Firefox está optimizada para pruebas de rendimiento. Herramientas como Firefox Profiler ayudan a analizar la eficiencia de la aplicación, facilitando la identificación y resolución de problemas de rendimiento.

## BOOTSTRAP

**Bootstrap** es un framework de código abierto utilizado para el desarrollo de interfaces WEB responsivas y modernas. Creado originalmente en Twiter por Mark Otto y Jacob Thortnton, permite a los desarrolladores crear aplicaciones y sitios web adaptables a diferentes tamaños de pantalla, desde dispositivos móviles hasta pantallas de escritorios, con una mínima configuración y sin necesidad de diseñar desde cero. Esta herramienta se ha vuelto popular y de gran ayuda desde su lanzamiento, es ampliamente utilizada por desarrolladores, empresas y igniciones de todo el mundo.

### ¿Qué es y cómo se utiliza?

Bootstrap es una colección de herramientas y componentes CSS, JavaScript y HTML que facilitan el diseño de interfaces WEB de manera coherente y eficiente. Al ofrecer componentes predefinidos como botones, formularios, templetes, modales, etc. Bootstrap permite a los usuarios acelerar el proceso de diseño y asegurar que los elementos visuales se mantengan consistentes en diferentes secciones del código.

### Características de Bootstrap.

* **Componentes predefinidos:** Bootstrap incluye una amplia gama de componentes listos para usar, como botones, formularios, barras de navegación, tarjetas y alertas, entre otros. Estos componentes pueden personalizarse y combinan estilos y funcionalidades que facilitan la creación de interfaces coherentes y atractivas.
* **JavaScript integrado**: Además de sus componentes CSS, Bootstrap incorpora varios **plugins de JavaScript** que añaden funcionalidad interactiva a las interfaces. Estos plugins incluyen elementos como modales, carruseles de imágenes, desplegables, etc. que permiten a los desarrolladores añadir interactividad sin necesidad de codificar desde cero.
* **Compatibilidad multiplataforma y multicapa**: Bootstrap es compatible con todos los navegadores modernos y se basa en una filosofía de diseño multiplataforma. Esto significa que el diseño resultante es uniforme y consistente en diversos entornos de navegador y sistemas operativos, facilitando la accesibilidad y optimización del proyecto.

## W3-SCHOOL

## STAR UML

Capítulo 3

Metodología

La metodología XP (Extreme Programming) es una metodología ágil de desarrollo de software desarrollada por Kent Beck autor del primer libro Extreme Programming Explained: Embrace Change (1999). Esta metodología se centrada en mejorar la calidad del software y la capacidad de respuesta ante los cambios. Se basa en valores clave como la comunicación, la simplicidad, el feedback, el coraje y el respeto. Las fases principales del ciclo de vida de un proyecto XP son las siguientes.

* 1. Planificación del proyecto

### Exploración

* En esta fase, el equipo se familiariza con los requisitos del cliente. Se generan historias de usuario que refieren las funcionalidades del sistema desde el punto de vista del beneficiario.
* El equipo explora diferentes opciones técnicas y prepara un plan preliminar.

### Historias de usuario

Las historias de usuario con el cliente tienen la finalidad de los casos de uso pero con las diferencias de que solo constan de 3 o 4 líneas escritas sobre el objetivo que se desea obtener sin tener que incluir código ni dar mucho hincapié a lo que se va a realizar Estas son adecuadas para dar a conocer para tener un tiempo estimado de desarrollo y verificar que el programa se cumple. Cuando se desea desarrolla una historia de usuario los desarrolladores y el usuario se reúnen para concretar o detallar el tiempo y lo que tiene que hacer dicha historia.

#### Historias de usuario del Administrador

|  |  |  |  |
| --- | --- | --- | --- |
| ID | HISTORIA DE USUARIO | CRITERIOS DE ACEPTACION | PRIORIDAD |
| HU001 | Como **auditor**, deseo poder **crear auditorías** para registrar información. | * El auditor puede ingresar todos los datos necesarios. * La auditoría se guarda automáticamente en la base de datos. * El auditor recibe una notificación cuando se guarda correctamente. | Alta |
| HU002 | Como **administrador**, deseo poder **agregar usuarios** al sistema. | * El administrador puede ingresar todos los datos requeridos. * El sistema permite seleccionar información de la base de datos para facilitar el proceso. * Los datos se guardan automáticamente. | Alta |
| HU003 | Como **administrador**, deseo poder **activar y desactivar usuarios** para controlar el acceso al sistema. | * El administrador puede activar o desactivar usuarios. * Los cambios se reflejan en la base de datos y los usuarios no podrán ingresar si no están activos. | Alta |
| HU005 | Como **administrador**, deseo poder **ver las auditorías respondidas** por los auditores. | * El administrador puede ver una lista de auditorías que ya han sido respondidas. | Alta |
| HU006 | Como **administrador**, me gustaría que el **usuario recibiera una notificación por correo** cuando tenga una auditoría pendiente. | * El usuario recibe una notificación automática por correo cuando tiene una nueva auditoría o una auditoría pendiente. | Alta |
| HU007 | Como **administrador**, deseo poder **ver datos y gráficas** de las auditorías en general. | * El administrador puede visualizar gráficos con datos clave sobre las auditorías, preguntas, usuarios, etc. | Alta |
| HU008 | Como **administrador**, deseo que las **auditorías se generen automáticamente** cada semana. | * Las auditorías se crean automáticamente y se asignan a los auditores.  - El administrador no necesita crear manualmente nuevas auditorías cada semana. | Alta |
| HU009 | Como **administrador**, me gustaría poder **editar una auditoría** para agregar o quitar preguntas. | * El administrador puede editar una auditoría existente. * Los cambios se guardan automáticamente en la base de datos. | Alta |
| HU010 | Como **administrador**, deseo poder **actualizar datos de los usuarios**. | * El administrador puede seleccionar un usuario de la lista para editar sus datos. * Los datos actualizados se guardan automáticamente en la base de datos. | Media |
| HU011 | Como **administrador**, me gustaría poder **ver qué usuarios están activos e inactivos** en el sistema. | * El administrador puede visualizar los usuarios activos e inactivos en una tabla. | Media |
| HU012 | Como **administrador**, deseo **recibir una notificación por correo** cuando el usuario conteste la auditoría por capas de la semana. | * El administrador recibe una notificación cuando se completa una auditoría. * Los datos se guardan automáticamente en la base de datos. | Media |
| HU013 | Como **administrador**, me gustaría poder **ver todas las preguntas activas** en la base de datos. | * El administrador puede ver una lista de todas las preguntas activas en el sistema. | Baja |
| HU014 | Como **administrador**, me gustaría poder **activar o desactivar preguntas** en las auditorías. | * El administrador puede activar o desactivar preguntas, y los cambios se actualizan automáticamente en la base de datos. | Baja |

#### Historias de Usuario del Auditor

|  |  |  |  |
| --- | --- | --- | --- |
| ID | HISTORIA DE USUARIO | CRITERIOS DE ACEPTACION | PRIORIDAD |
| HU001 | Como auditor deseo poder ingresar al sistema de auditorías. | * El usuario ingresara sus credenciales para poder ingresar al sistema. | Alta |
| HU002 | Como auditor deseo poder ingresar al sistema y ver mis auditorias pendientes | * Cuando el usuario inicie sesión podrá ver un menú con las auditorias pendientes | Alta |
| HU003 | Como auditor deseo poder responder las preguntas correspondientes a la auditoria | * El usuario podrá obtener las preguntas correspondientes a su auditoria y responderlas | Alta |
| HU004 | Como auditor deseo poder tomar una foto cada que una pregunta no se cumpla | * Cada que una pregunta no se cumpla podrá abrir su cámara y tomar una fotografía para evidenciar el problema | Alta |
| HU005 | Como auditor deseo que las preguntas que no se cumplan tengan una selección de retroalimentación o acciones tomadas | * Cuando una pregunta no se cumpla podrá verla en la sección de “Acciones tomadas” * Podrá responder con la acción que se implementará. | -Alta |
| HU006 | Como auditor puedo seleccionar un supervisor para dar a conocer los problemas que se encuentran durante la auditoria. | * El auditor que realice la auditoria podrá pasarle la auditoria a su supervisor para que verificar y seguir las acciones tomadas | Alta |
| HU007 | Como auditor deseo poder agregar un comentario cuando una pregunta no se cumpla | * El auditor agregara comentarios sobre las preguntas que no se cumplan | Media |
| HU008 | Como auditor deseo que cada que me agreguen una auditoria me llegue una notificación por correo | * El auditor cada que le agreguen una auditoria podrá recibir una notificación para ver esa auditoria y responderla. | Media |
| HU009 | Como auditor me gustaría que cuando termine la auditoria la mande y no tenga nada pendiente | * Cuando el auditor termine su auditoria podrá enviarla y no tener nada pendiente o que se mantenga pendiente si aun no se responde. | Media |
| HU0010 | Como auditor deseo ver un dashboard. | * El auditor podrá visualizar un dashboard relacionado a sus auditorias | Baja |
| HU0011 | Como auditor deseo poder agregar preguntas | * El usuario o auditor podrá agregar preguntas que le gustaría que se integraran en su auditoria * El administrador podrá ver su pregunta en su panel | Baja |

Capítulo 4

Desarrollo

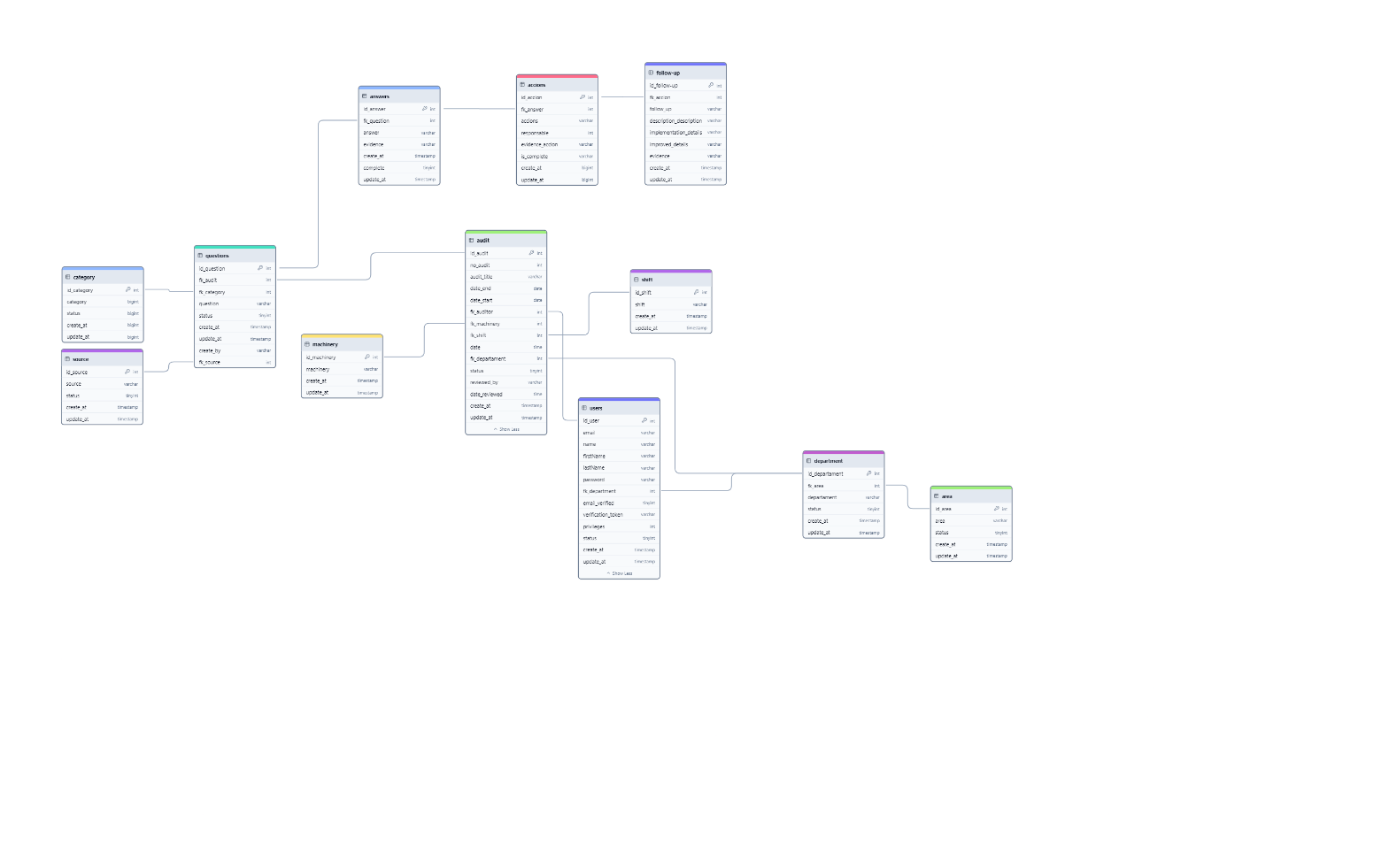
Durante el desarrollo se dará a conocer cómo se fue desarrollando el Sistema auditoria por capas, desde el diagrama entidad relación hasta como se desarrolló el sistema y su lógica a fin de comprender un poco más y detalladamente el sistema, inconvenientes y resultados.

## Desarrollo de la base de datos.

A continuación, se presentan un boceto de cómo se desarrollaba una auditoria con capas con el fin de entender consecuentemente el diagrama ER.



El primer enfoque se basa en en la construcción de la base de datos. Se comenzó por definir el diagrama entidad-relación esto con el fin de poder organizar, gestionar y analizar relevante de las auditorias, una herramienta fundamental para visualizar cómo se relacionarían las diferentes tablas y qué atributos serían necesarios con el fin de tener los necesarios para la correcta funcion. Esto permitió establecer una estructura sólida y coherente, siguiendo las tres formas normales para garantizar la integridad de los datos.



### Entidades principales

1. **audit**: Esta es la entidad central del sistema, que representa cada auditoría realizada. Contiene atributos clave como “audit\_id”, “date”, “status”, entre otros, que ayudan a identificar y gestionar cada auditoría individual. Esta entidad se relaciona con otras entidades que brindan información detallada sobre las áreas, categorías y preguntas evaluadas en cada auditoría.
2. **question**: Define las preguntas que se aplican en la auditoría. Cada pregunta tiene su propio “question\_id” y está asociada a una categoría a través del campo “category\_id”, lo que permite organizar las preguntas en grupos específicos dependiendo de la categoría.
3. **category**: Representa las categorías de auditoría, como calidad, seguridad o productividad. Esta entidad permite agrupar las preguntas relacionadas con cada tema, lo cual ayuda a estructurar el contenido de la auditoría en secciones lógicas.
4. **answer**: Almacena las respuestas proporcionadas a cada pregunta durante la auditoría. Esta entidad incluye el “answer\_id”, el “question\_id” correspondiente y otros atributos necesarios para guardar el resultado de cada evaluación.

### Entidades de soporte y gestión.

1. **area**: Define las áreas específicas de las que depende cada departamento que son auditadas. Esto permite un análisis más detallado y específico, ya que la auditoría puede realizarse en distintos sectores dentro de un área por ejemplo el área de papel tiene como departamento a calidad papel, producción papel, etc. Entonces cada departamento tendrá una auditoria.
2. **department**: Representa los distintos departamentos de la empresa en los cuales se pueden realizar auditorías. Esta entidad permite especificar el departamento auditado, facilitando la clasificación de auditorías por áreas de la organización.
3. **shift**: Representa los turnos en los cuales se realizan las auditorías. Esto permite una mejor gestión de los tiempos en los que se ejecuta cada evaluación, facilitando un seguimiento ordenado.
4. **user**: Almacena los datos de los usuarios que tienen acceso al sistema y ejecutan las auditorías. Esto incluye información como “username”, “password”, “status”, que ayudan a controlar el acceso y las funciones disponibles para cada usuario.

## Desarrollo de Mockups

Durante el desarrollo del sistema se tocaron varios puntos como los que son son los mockups, etc.

### Desarrollo de login y “Olvide mi contraseña”



*Figura 1. Mockuo del Login*

1. El primer mockup corresponde a el login o inicio de sesión donde un usuario podrá ingresar sus credenciales para entrar acceder al sistema como administrador o como auditor de un area especifica al ingresar los datos se hace la consulta ala base de datos para autenticar al usuario. Véase en la **Figura .**



*Figura 2. Vista del login*

1. En la **Figura**  se muestra la vista ya desarrollada de la vista de inicio se sesión. En la interfaz el sistema nos solicita el email y contraseña para poder iniciar sesión.



*Figura 3. "Recuperar contraseña"*

1. El segundo mockup corresponde a la funcionalidad de **"Recuperar mi contraseña"**. En esta pantalla, el usuario podrá ingresar el correo electrónico asociado a su cuenta. Una vez ingresado, el sistema enviará un correo con un enlace para que el usuario pueda restablecer su contraseña. Véase la **Figura .**



*Figura 4. Vista de "Recuperar contraseña"*

1. En la **figura**  se muestra la vista desarrollada de **“Recuperar contraseña”** donde el usuario debe ingresar su correo electrónico para que le mande un token y este pueda restablecer su contraseña mediante un enlace.

### Desarrollo de mockups del administrador.



*Figura 5. Home*

1. Cuando un usuario ingresa sus credenciales como administrador, será redirigido a la página **admin/home**, donde se mostrará una tabla con los usuarios activos e inactivos. El administrador tendrá la capacidad de buscar, editar y agregar usuarios. Véase la **Figura 4**



*Figura 6. Crear usuario*

1. Al seleccionar el botón **"Add User"** (véase la **Figura 3**, sección marcada en rojo), se desplegará una nueva ventana en la página **admin/create**, donde el administrador deberá completar los campos correspondientes para agregar un nuevo usuario. Al hacer clic en el botón **"Add"**, los datos ingresados serán guardados en la base de datos, y el nuevo usuario aparecerá en la tabla mostrada en la **Figura 3**.Desarrollo mockups del usuario.



*Figura 7. Crear una auditoria*

Al seleccionar la opción de “**crear auditoria**” del menú del sistema. Inicialmente se selecciona la maquinaria utilizada, el turno en el cual se realizará, el departamento involucrado y el nombre de auditor presente (todo eso traído desde la base de datos). Se necesitan plantear preguntas al departamento dirigido. Una vez completados todos los campos obligatoriamente. Se escribirán las preguntas las cuales están organizadas por categorías. Entonces cada que se escriba una pregunta el sistema nos mostrara la fuente y podremos seleccionar una y guardar en la base de datos, véase en la **figura 5**

**Filtros**

**Migraciones  
Base de Datos**

Capítulo 5

Productos del proyecto

* 1. Contenido

Manuales de usuario, códigos relevantes, infografías

* 1. Más información

El vídeo proporciona una manera eficaz para ayudarle a demostrar el punto. Cuando haga clic en Vídeo en línea, puede pegar el código para insertar del vídeo que desea agregar. También puede escribir una palabra clave para buscar en línea el vídeo que mejor se adapte a su documento. Para otorgar a su documento un aspecto profesional, Word proporciona encabezados, pies de página, páginas de portada y diseños de cuadro de texto que se complementan entre sí. Por ejemplo, puede agregar una portada coincidente, el encabezado y la barra lateral.

Haga clic en Insertar y elija los elementos que desee de las distintas galerías. Los temas y estilos también ayudan a mantener su documento coordinado. Cuando haga clic en Diseño y seleccione un tema nuevo, cambiarán las imágenes, gráficos y gráficos SmartArt para que coincidan con el nuevo tema. Al aplicar los estilos, los títulos cambian para coincidir con el nuevo tema. Ahorre tiempo en Word con nuevos botones que se muestran donde se necesiten.

* 1. Más información

El vídeo proporciona una manera eficaz para ayudarle a demostrar el punto. Cuando haga clic en Vídeo en línea, puede pegar el código para insertar del vídeo que desea agregar. También puede escribir una palabra clave para buscar en línea el vídeo que mejor se adapte a su documento. Para otorgar a su documento un aspecto profesional, Word proporciona encabezados, pies de página, páginas de portada y diseños de cuadro de texto que se complementan entre sí. Por ejemplo, puede agregar una portada coincidente, el encabezado y la barra lateral.

Haga clic en Insertar y elija los elementos que desee de las distintas galerías. Los temas y estilos también ayudan a mantener su documento coordinado. Cuando haga clic en Diseño y seleccione un tema nuevo, cambiarán las imágenes, gráficos y gráficos SmartArt para que coincidan con el nuevo tema. Al aplicar los estilos, los títulos cambian para coincidir con el nuevo tema. Ahorre tiempo en Word con nuevos botones que se muestran donde se necesiten.

Capítulo 6

Conclusiones y recomendaciones

* 1. Contenido

Resaltar lo esencial del proyecto realizado; comentar los logros, dimensionar los resultados en tiempo y espacio, mostrando medios para ir más lejos en desarrollos similares

* 1. Más información

El vídeo proporciona una manera eficaz para ayudarle a demostrar el punto. Cuando haga clic en Vídeo en línea, puede pegar el código para insertar del vídeo que desea agregar. También puede escribir una palabra clave para buscar en línea el vídeo que mejor se adapte a su documento. Para otorgar a su documento un aspecto profesional, Word proporciona encabezados, pies de página, páginas de portada y diseños de cuadro de texto que se complementan entre sí. Por ejemplo, puede agregar una portada coincidente, el encabezado y la barra lateral

* 1. Más información

El vídeo proporciona una manera eficaz para ayudarle a demostrar el punto. Cuando haga clic en Vídeo en línea, puede pegar el código para insertar del vídeo que desea agregar. También puede escribir una palabra clave para buscar en línea el vídeo que mejor se adapte a su documento. Para otorgar a su documento un aspecto profesional, Word proporciona encabezados, pies de página, páginas de portada y diseños de cuadro de texto que se complementan entre sí. Por ejemplo, puede agregar una portada coincidente, el encabezado y la barra lateral.

Referencias

|  |  |
| --- | --- |
| [1] | E. Place, PROGRAMACIÓN ORIENTADA A OBJETOS, www.surforce.com, 2009. |