

**UNIVERSIDAD POLITÉCNICA DE ATLACOMULCO**

Ingeniería en Sistemas Computacionales

**PROYECTO DE ESTANCIA I o II**

**“NOMBRE DEL PROYECTO”**

Presentado por:

Nombre del Alumno

Profesor Asesor:

Nombre del Profesor

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

Atlacomulco, México. Diciembre 2017

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](#_Toc181886075)

[1.1 Introducción 1](#_Toc181886076)

[1.2 Justificación 1](#_Toc181886077)

[1.3 Objetivos 2](#_Toc181886078)

[1.4 Problema a resolver 3](#_Toc181886079)

[1.5 Alcances y limitaciones 3](#_Toc181886080)

1. [Marco teórico 4](#_Toc181886081)

[2.1 Definición de Sistema WEB. 4](#_Toc181886082)

[2.1.1 Elementos de un sistema en WEB. 4](#_Toc181886083)

[2.1.2 Funcionamiento de un Sistema WEB. 5](#_Toc181886084)

[2.1.3 Tipos de sistema WEB 5](#_Toc181886085)

[2.2 Auditoria por Capas 5](#_Toc181886086)

[2.2.1 Concepto de Auditoria por Capas. 6](#_Toc181886087)

[2.2.2 Ejemplo de Auditoria por Capas. 6](#_Toc181886088)

[2.2.3 Beneficios de las auditorias por capas. 6](#_Toc181886089)

[2.3 Mockups 7](#_Toc181886090)

[2.3.1 Herramientas utilizadas 7](#_Toc181886091)

[2.4 Base de datos 8](#_Toc181886092)

[2.4.1 Tipos de Bases de datos 8](#_Toc181886093)

[2.4.2 Tipos de Gestores de Base de Datos. 9](#_Toc181886094)

[2.5 LARAGON 9](#_Toc181886095)

[2.6 Que es un FRAMEWORK 10](#_Toc181886096)

[2.6.1 Ventajas de usar un FRAMEWORK. 10](#_Toc181886097)

[2.7 CODEIGNITER 4 11](#_Toc181886098)

[2.8 PHP 12](#_Toc181886099)

[2.8.1 ¿Por qué PHP? 12](#_Toc181886100)

[2.9 MVC 13](#_Toc181886101)

[2.9.1 ¿Qué significa MVC? 13](#_Toc181886102)

[2.9.2 Porque MVC 13](#_Toc181886103)

[2.10 JAVA SCRIPT 14](#_Toc181886104)

[2.10.1 Características principales de Java Script 14](#_Toc181886105)

[2.10.2 Java Script en el Frontend 15](#_Toc181886106)

[2.11 HTML 16](#_Toc181886107)

[2.12 CSS 16](#_Toc181886108)

[2.13 VISUAL STUDIO CODE 16](#_Toc181886109)

[2.14 HEIDI SQL 16](#_Toc181886110)

[2.15 NAVICAT 16](#_Toc181886111)

[2.16 CHART DBS 16](#_Toc181886112)

[2.17 GIT 16](#_Toc181886113)

[2.18 GITHUB 16](#_Toc181886114)

[2.19 FIREFOX DEVELOPMENT 16](#_Toc181886115)

[2.20 BOOTSTRAP 16](#_Toc181886116)

[2.21 W3-SCHOOL 16](#_Toc181886117)

1. [Metodología 17](#_Toc181886118)

[3.1 Exploración 17](#_Toc181886119)

[3.1.1 Historias de usuario del administrador. 17](#_Toc181886120)

[3.1.2 Historias de Usuario del Auditor 20](#_Toc181886121)

1. [Desarrollo 22](#_Toc181886122)

[4.1 Desarrollo de Mockups 22](#_Toc181886123)

[4.1.1 Desarrollo de login y “Olvide mi contraseña” 22](#_Toc181886124)

[4.1.2 Desarrollo de mockups del administrador. 24](#_Toc181886125)

1. [Productos del proyecto 27](#_Toc181886126)

[7.1 Contenido 27](#_Toc181886127)

[7.2 Más información 27](#_Toc181886128)

[7.3 Más información 27](#_Toc181886129)

1. [Conclusiones y recomendaciones 29](#_Toc181886130)

[8.1 Contenido 29](#_Toc181886131)

[8.2 Más información 29](#_Toc181886132)

[8.3 Más información 29](#_Toc181886133)

1. [Referencias 31](#_Toc181886134)

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 a 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 pagina de manera eficiente y clara.

HTML contiene diferentes tipos de etiquetas y elementos que permiten definir estructuras y formato a la pagina WEB. Por ejemplo colocar textos en colores diferentes títulos de diferentes tamaños, enlaces que nos puedan dirigir a una pagina en concreto, colocar imágenes, insertar cuadros de dialogo, etc. HTML va de la mano con la Vista

## CSS

## VISUAL STUDIO CODE

## HEIDI SQL

## NAVICAT

## CHART DBS

## GIT

## GITHUB

## FIREFOX DEVELOPMENT

## BOOTSTRAP

## W3-SCHOOL

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. 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 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 están desactivados. | 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 y ver mis auditorias pendientes |  |  |
| HU002 | Como auditor deseo poder responder las preguntas correspondientes a la auditoria |  |  |
| HU003 | Como auditor deseo poder elegir una foto o subir una imagen cada que una pregunta no se cumpla |  |  |
| HU004 | Como auditor deseo poder agregar un comentario cuando una pregunta no se cumpla |  |  |
| HU005 | Como auditor deseo que las preguntas que no se cumplan tengan una selección de retroalimentación o acciones tomadas |  |  |
| HU006 | **Como auditor puedo seleccionar un supervisor para dar a conocer los problemas que se encuentran durante la auditoria** |  |  |
| HU007 | Como auditor deseo que cada que me agreguen una auditoria me llegue una notificación por correo |  |  |
| HU008 | Como auditor me gutaria que cuando termine la auditoria la mande y no tenga nada pendiente |  |  |
| HU008 | Como auditor deseo |  |  |

Capítulo 4

Desarrollo

## 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. |