UNIVERSIDAD LATINA DE COSTA RICA

FACULTAD DE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIONES

INGENIERÍA DE SISTEMAS COMPUTACIONALES

Creación de página interna para colaboradores y departamento de Recursos Humanos de la empresa CCM Soluciones en Servicios

Daniel Cordero Calderón

Heredia

2024

# Hoja del tribunal

# Cumplimiento de objetivos

# Tutor

# Filólogo

# Crai

# Dedicatoria

# Agradecimientos

# Índice

[Hoja del tribunal ii](#_Toc533694152)

[Cumplimiento de objetivos iii](#_Toc533694153)

[Tutor iv](#_Toc533694154)

[Filólogo v](#_Toc533694155)

[Crai vi](#_Toc533694156)

[Dedicatoria vii](#_Toc533694157)

[Agradecimientos viii](#_Toc533694158)

[Índice ix](#_Toc533694159)

[Índice Figuras xi](#_Toc533694160)

[Índice de tablas xii](#_Toc533694161)

[Primera parte: Problema y Propósito 13](#_Toc533694162)

[Estado actual del caso 14](#_Toc533694163)

[Formulación del problema 14](#_Toc533694164)

[Problema general 14](#_Toc533694165)

[Problemas específicos 14](#_Toc533694166)

[Justificación de las herramientas usadas en el proyecto 14](#_Toc533694167)

[Hardware a utilizar 14](#_Toc533694168)

[Software a Utilizar 14](#_Toc533694169)

[Objetivo del proyecto 14](#_Toc533694170)

[Objetivo general 14](#_Toc533694171)

[Objetivos específicos 14](#_Toc533694172)

[Alcances y limitaciones 14](#_Toc533694173)

[Alcances 14](#_Toc533694174)

[Limitaciones 14](#_Toc533694175)

[Segunda Parte: Marco situacional y Marco conceptual 16](#_Toc533694176)

[Marco situacional 17](#_Toc533694177)

[Institución 17](#_Toc533694178)

[Misión 17](#_Toc533694179)

[Visión 17](#_Toc533694180)

[Valores 17](#_Toc533694181)

[Estructura organizacional 17](#_Toc533694182)

[Marco teórico 17](#_Toc533694183)

[Tercera Parte: Metodología 18](#_Toc533694184)

[Sujetos y Fuentes de Información 19](#_Toc533694185)

[Sujetos 19](#_Toc533694186)

[Fuentes de Información 19](#_Toc533694187)

[Tipo de investigación 19](#_Toc533694188)

[Diseño de investigación 19](#_Toc533694189)

[Instrumentos y Técnicas utilizadas en la recopilación de los datos. 19](#_Toc533694190)

[Método de investigación 19](#_Toc533694191)

[Cuarta Parte: Conclusiones y Recomendaciones 20](#_Toc533694192)

[Conclusiones 21](#_Toc533694193)

[Recomendaciones 21](#_Toc533694194)

[Quinta Parte: Propuesta 22](#_Toc533694195)

[Análisis de requerimientos 23](#_Toc533694196)

[Fase de preparación 23](#_Toc533694197)

[Fase de instalación 23](#_Toc533694198)

[Sub faces necesarias 23](#_Toc533694199)

[Costos Estimados 23](#_Toc533694200)

[Cronograma 23](#_Toc533694201)

[Bibliografía 24](#_Toc533694202)

[Glosario 25](#_Toc533694203)

[Anexos 26](#_Toc533694204)

# Índice Figuras

# Índice de tablas

# Primera parte: Problema y Propósito

## Estado actual del caso

La empresa CCM Soluciones y Servicios actualmente necesita un sistema para recursos humanos donde se pueda centralizar y automatizar procesos que se realizan manualmente, por lo que se desea realizar un sistema web en donde cada empleado tenga la información a mano como boletas de pago, solicitud de vacaciones y permisos, consulta de vacaciones disponibles, entre otros temas, todo desde un mismo lugar sin tener que pasar por un tercero para obtener la información. De esta forma se garantiza la autogestión de la información que cada colaborador requiere centralizando la acción, asegurando mejores tiempos de respuesta y abaratando los costos que implica la gestión administrativa en el área de recursos humanos propiamente.

## Formulación del problema

### Problema general

¿Cómo puede la empresa CCM Soluciones y Servicios centralizar y automatizar sus procesos de recursos humanos, permitiendo a los empleados acceder fácilmente a información, eliminando la necesidad de intermediarios para obtener dicha información?

### Problemas específicos

¿Cómo puede la empresa CCM Soluciones y Servicios superar la fragmentación y la falta de eficiencia en los procesos del área de recursos humanos mediante la implementación de un sistema web que permita la centralización y automatización de tareas, facilitando el acceso directo de los empleados a esta información de manera rápida y sin necesidad de intermediarios?

## Justificación de las herramientas usadas en el proyecto

* HTML
* CCS
* Boostrap
* PHP: El lenguaje de programación PHP se utilizará para el desarrollo de los módulos y submódulos del sistema que se desea implementar.
* Javascript
* MySQL: Se utilizará por la capacidad que tiene para almacenar información y por la seguridad que esta misma brinda.
* GitHub: Se utilizará ya que es un sistema de gestión de proyectos y control de versiones de código, que contribuye a que el desarrollo sea más ágil, además ayuda a corregir errores cuando el sistema está en producción sin tener que eliminar el proyecto directamente del servidor.
* Visual Studio Code: Esta herramienta es el editor de código, en donde se desarrolla todo el código, está adaptado para diferentes lenguajes de programación y tiene la ventaja de traer incorporados diferentes extensiones que facilitaran la elaboración del código.

### Hardware por utilizar

El hardware por utilizar en el desarrollo del proyecto es una computadora, la cual tiene instalado lo necesario para el buen desarrollo de la página interna, en este caso es una HP ENVY que cuenta con 32GB, procesador INTEL Core i7-1260P 12th Generación con sistema operativo 64bits.

### Software Por Utilizar

## Objetivo del proyecto

### Objetivo general

Implementar un sistema web integral de gestión de recursos humanos en la empresa CCM Soluciones y Servicios para el segundo cuatrimestre del 2024, con el propósito de centralizar, automatizar y facilitar el acceso a procesos de gestión de boletas de pago y solicitudes de vacaciones, promoviendo así la eficiencia operativa, la transparencia y la satisfacción del personal.

### Objetivos específicos

* Analizar las necesidades del departamento de Recursos Humanos de la empresa CCM Soluciones en Servicios.
* Diseñar la interfaz gráfica y experiencia de usuario que va a requerir el sistema web.
* Desarrollar el backend y frondend del sistema web.
* Implementar el sistema web como parte del día a día de cada empleado.

## Alcances y limitaciones

### Alcances

1. Análisis de requerimientos: Reuniones con los diferentes departamentos de la empresa para identificar las necesidades específicas de cada área en cuanto a la gestión de recursos humanos.
2. Diseño del sistema: Elaboración de prototipos y diagramas de flujo para visualizar la arquitectura y funcionalidades del sistema web.
3. Desarrollo del sistema: Programación de las diferentes funcionalidades del sistema web, incluyendo la gestión de boletas de pago, solicitudes de vacaciones, permisos, entre otros.
4. Implementación y pruebas: Instalación del sistema en los servidores de la empresa y realización de pruebas exhaustivas para asegurar su correcto funcionamiento y detectar posibles fallos o errores.
5. Capacitación del personal: Elaboración de materiales de capacitación y realización de sesiones informativas para que los empleados aprendan a utilizar el nuevo sistema de recursos humanos de manera efectiva.
6. Puesta en marcha: Lanzamiento oficial del sistema web y acompañamiento durante los primeros días de su uso para resolver cualquier duda o inconveniente que pudiera surgir.

### Limitaciones

#### Limitaciones de tiempo

1. Restricción de tiempo: La cantidad limitada de horas disponibles podría afectar la profundidad del análisis de requerimientos, el diseño del sistema, la implementación, las pruebas y la capacitación del personal.
2. Complejidad del proyecto: Dada la brevedad del tiempo disponible, puede ser necesario simplificar el alcance del proyecto y priorizar las funcionalidades clave del sistema de recursos humanos para garantizar su entrega dentro del plazo establecido.
3. Capacidad de personal: La cantidad limitada de horas también podría afectar la disponibilidad del personal capacitado para llevar a cabo todas las actividades necesarias en el proyecto.
4. Pruebas y ajustes: El tiempo limitado puede dificultar la realización de pruebas exhaustivas del sistema y la implementación de ajustes o mejoras basadas en los resultados de las pruebas.
5. Documentación y seguimiento: Dado el corto plazo, puede ser difícil dedicar tiempo suficiente a la documentación adecuada del proyecto y al seguimiento continuo de su progreso.

#### Limitaciones de costo: Con más recursos de tiempo y dinero, es posible ampliar el alcance del sistema generando un mejor retorno a la empresa.

#### Limitaciones de alcance. El factor tiempo mita el alcance del sistema según se detalló anteriormente.

#### Limitaciones de tecnología

#### Limitaciones de recurso humano De existir más profesionales del área de la ingeniería en sistemas, la complejidad y alcance de la propuesta podría ser mayor generando más beneficios asociados a la empresa.

#### Limitación legal

#### Limitaciones organizacionales Al no existir un departamento de TI con profesionales asignados para el desarrollo de proyectos como el que se encuentra en estudios, las posibilidades de impacto y crecimiento en el área son limitadas.

# Segunda Parte: Marco situacional y Marco conceptual

## Marco situacional

### Institución

Todo comenzó en el 2001 con una tienda llamada Bikestation, con el pasar de los años, en el 2008 se crea CCM Soluciones en Servicios como distribuidor mayorista de Specialized en Costa Rica. Hoy en día contamos con un e-commerce y 8 tiendas en todo el país, donde 5 de las mismas se encuentran en el GAM cubriendo más del 60% de la población, además se cuenta con una red de más de 50 tiendas en el resto del país para garantizar el posicionamiento y distribución de la marca.

### Misión

Brindamos la mejor experiencia de este mercado con conciencia individual e integración grupal, caracterizados por una gestión de liderazgo positivo y de crecimiento continuo, que evidencia pasión, respeto, perseverancia, responsabilidad y compromiso.

### Visión

Ser el equipo líder del mercado, excepcional, innovador, comprometido con la proactividad y productividad, en beneficio de accionistas, personas internas, externas y comunidad.

### Valores

* Pasión (Perseverancia-Compromiso)
  + Nuestro compromiso es ir más allá de nuestros deberes por el bien común Perseverar en la adversidad con alegría
* Respeto (Humildad)
  + Armonía y aceptación como formas de compañerismo.
  + Valorarnos aún en las diferencias.
  + Es recordar que denigrar es denigrarme a mí mismo.
  + Es respeto por el trabajo de los demás, independientemente de lo que sea.
  + Es Dignificar a la persona, no al puesto.
* Responsabilidad (Honestidad)
  + Las cosas a tiempo, bien hechas.
  + Es integridad en el manejo de los activos.
  + Es puntualidad para no afectar a los demás.
  + Es ser diligentes: soy responsable de lo que hago y de cómo ello afecta a los demás.
  + Hacer el trabajo de forma correcta.
  + Aceptar críticas constructivas.
  + Buen flujo de información.
* Gratitud (Lealtad-Solidaridad)
  + Es reconocimiento y valoración.
  + Es potencializar la motivación de los demás.
  + Es demostrarles a las personas, igualmente, como deseo ser tratado(a).
  + Es agradecimiento cuando alguien ejecuta una acción.
  + Lealtad: Reciprocidad en el servicio.
  + Solidaridad: Forma de agradecer a la comunidad.
* Trabajo en equipo (compañerismo)
  + Estar comprometidos en colaborar.
  + Solidaridad en momentos difíciles.
  + Dar más por una tarea de equipo. Milla extra. Sacrificio.
  + Compromiso.

### Estructura organizacional

A diagram of a company structure

Description automatically generated

## Marco teórico

**Gestión de Recurso Humano**

La gestión del talento humano (GTH) es la aplicación de los modelos y conceptos de la moderna gestión humana (GH) en las organizaciones. No es la gestión de las personas en sí, sino la gestión de los talentos y competencias que las personas aprenden, poseen y aplican. Se trata de una forma de desarrollar y empoderar y de orientar y apalancar a las personas, no de gestionarlas. Las organizaciones cambian sus conceptos y modifican sus prácticas administrativas para agregar, comprometer, movilizar y utilizar a plenitud las actividades de las personas. En lugar de invertir directamente en productos y servicios, ahora invierten en las personas que los conocen bien y que saben cómo crearlos, desarrollarlos, producirlos y mejorarlos. En lugar de simplemente invertir en los clientes, invierten en las personas que los atienden y les sirven, y que saben cómo satisfacerlos, en-cantarlos y hacer que sean fieles. Y ganan más con eso. Así las personas se convierten en el elemento básico del éxito de la empresa.

Chiavenato, I. (2020). *Gestión del talento humano: el nuevo papel de los recursos humanos en las organizaciones*. McGraw-Hill. <https://www.ebooks7-24.com:443/?il=9350>

**Diseño de experiencia de usuario**

El diseño de la experiencia de usuario es un conjunto de actividades de proceso incrementales, que ayudan al equipo de desarrollo y a las partes interesadas en el proyecto a concentrarse en proveer una experiencia positiva para los usuarios del producto de software. El diseño de la UX es más amplio que el diseño de la interfaz de usuario y la ingeniería de usabilidad o accesibilidad. Debe comenzar en las primeras etapas del ciclo de vida del proyecto si queremos que sea efectivo. Los desarrolladores que se esperan hasta el final de un proyecto para agregar la funcionalidad de la interfaz de usuario tendrán pocas probabilidades de proveer una experiencia placentera para los usuarios.

Pressman, R. S., Maxim, B. R.(2021). *Ingeniería de software*. McGraw-Hill Interamericana. <https://www.ebooks7-24.com:443/?il=16414>

**Interfaz de usuario**

El diseño de interfaces para software es análogo a un conjunto de dibujos detallados (y especificaciones) de las puertas, ventanas e instalaciones externas de una casa. Esos dibujos detallados (y especificaciones) de las puertas, ventanas e instalaciones externas nos dicen cómo fluyen las cosas y la información hacia dentro y fuera de la casa, y dentro de los cuartos que son parte del plano de plantas. Los elementos de diseño de interfaz para software representan la información que fluye hacia dentro y fuera de un sistema, además de cómo están comunicados los componentes definidos como parte de la arquitectura.

Hay tres elementos importantes del diseño de interfaces: (1) la interfaz de usuario (UI); (2) interfaces externas; para otros sistemas, dispositivos, redes u otros productores o consumidores de información; y (3) interfaces internas entre varios componentes de diseño. Estos elementos de diseño de interfaz permiten que el software se comunique al exterior y permiten la comunicación y colaboración internas entre los componentes que conforman la arquitectura de software. El diseño de UI (que cada vez se le conoce más como UX o diseño de experiencia del usuario) es una importante acción de ingeniería. El diseño de UX se enfoca en asegurar la facilidad de uso del diseño de UI. Un diseño utilizable incorpora elementos estéticos cuidadosamente elegidos (como distribución, color, gráficos, distribución de la información), elementos ergonómicos (como mecanismos de interacción, colocación de la información, metáforas, navegación de UI) y elementos técnicos (como patrones de UX, componentes reutilizables). En general, la UI es un subsistema único dentro de la arquitectura en general de la aplicación, diseñada para proveer al usuario final una experiencia satisfactoria. El diseño de interfaces externas requiere de información definitiva sobre la entidad a la que se enviará o la que recibirá información. En todo caso, hay que recolectar esta información durante la ingeniería de requerimientos y verificarla una vez que comience el diseño de la interfaz. El diseño de interfaces externas debe incorporar la comprobación de errores y características de seguridad apropiadas.

A computer screen shot of a diagram

Description automatically generated

Pressman, R. S., Maxim, B. R.(2021). *Ingeniería de software*. McGraw-Hill Interamericana. <https://www.ebooks7-24.com:443/?il=16414>

**Experiencia de usuario**

El diseño de la experiencia de usuario es un conjunto de actividades de proceso incrementales, que ayudan al equipo de desarrollo y a las partes interesadas en el proyecto a concentrarse en proveer una experiencia positiva para los usuarios del producto de software. El diseño de la UX es más amplio que el diseño de la interfaz de usuario y la ingeniería de usabilidad o accesibilidad. Debe comenzar en las primeras etapas del ciclo de vida del proyecto si queremos que sea efectivo. Los desarrolladores que se esperan hasta el final de un proyecto para agregar la funcionalidad de la interfaz de usuario tendrán pocas probabilidades de proveer una experiencia placentera para los usuarios. En este capítulo nos concentraremos en los aspectos del diseño de interfaces de usuario en el contexto del diseño de la experiencia de usuario.

El diseño de la experiencia de usuario trata de asegurar que ningún aspecto de su software aparezca en la versión candidata final sin la decisión explícita del equipo de desarrollo y otras partes interesadas de incluirla. Esto implica tener en cuenta cada acción y expectativa razonable del usuario, durante cada paso del proceso de desarrollo. Para realizar la tarea de elaborar una experiencia de usuario positiva más manejable, Garret sugiere desglosarla en sus elementos componentes: estrategia, alcance, estructura, esqueleto y superficie. Las relaciones entre estos elementos componentes y los subcomponentes se muestran en la figura 12.1. La organización de Garret con respecto al diseño de la UX puede interpretarse de la siguiente manera para el desarrollo de productos de software: • Estrategia. identifica las necesidades del usuario y las metas de negocios del cliente que forman la base para todo el trabajo de diseño de la UX • Alcance. Incluye los requerimientos tanto funcionales como de contenido (por ejemplo, información, medios, servicios) necesarios para realizar un conjunto de características consistente con la estrategia del producto.

235CAPÍTULO 12 DISEÑO DE LA EXPERIENCIA DE USUARIO

• Estructura Consiste en el diseño de la interacción [por ejemplo, cómo reacciona el sistema en respuesta a las acciones de usuario (sección 12.1.2)] y la arquitectura de la información [es decir, la organización de los elementos de contenido (sección 12.1.1)]

• Esqueleto Consta de tres componentes: diseño de la información (por ejemplo, la presentación del contenido de una forma que sea entendible para el usuario), diseño de la interfaz [es decir, organizar los objetos en la pantalla de la interfaz para que el usuario pueda trabajar con la funcionalidad del sistema (sección 12.5)], diseño de la navegación (esto es, el conjunto de elementos en pantalla que permiten a los usuarios recorrer la arquitectura de la información)

• Superficie Presenta el diseo visual o la apariencia del proyecto terminado a sus usuarios (sección 12.1.4)Varios aspectos transversales del diseño de UX son de especial interés para los ingenieros de software: arquitectura de la información, diseño de la interacción con el usuario, ingeniería de usabilidad y diseño visual.

Pressman, R. S., Maxim, B. R.(2021). *Ingeniería de software*. McGraw-Hill Interamericana. <https://www.ebooks7-24.com:443/?il=16414>

**Backend y frontend**

El frontend o «desarrollo del lado del cliente» se refiere a la práctica de producir HTML, CSS y JavaScript. Estos tres elementos se encargan de dar forma a la parte frontal de un sitio web o aplicación. Esto incluye los fondos, colores, texto, animaciones o efectos. Precisamente de ahí proviene el nombre de «desarrollo del lado del cliente», pues con el frontend se puede construir por completo lo que los usuarios perciben al explorar un sitio y con el que pueden interactuar.

Uso del frontend

El frontend sirve para realizar la interfaz de un sitio web, desde su estructura hasta los estilos, como pueden ser la definición de los colores, texturas, tipografías, secciones, entre otros. Su uso es determinante para que el usuario tenga una buena experiencia dentro del sitio o aplicación.

Elementos del frontend

Estructuras de navegación.

Este elemento se refiere al orden en que se organizan las diferentes páginas de un sitio web y a los componentes que se vinculan entre sí para realizar diferentes funciones dentro del sitio.

 Layout

También nombrado diseño de página, se refiere a todos los componentes de la página web, por ejemplo: ubicación del menú, botones, footer; todo lo necesario para que un sitio sea útil y fácil de navegar.

Contenido web

Todo aquello que brinde información relevante o interesante para los usuarios. Es importante destacar que el contenido no tiene que ser necesariamente texto, puede incluir sonido o materiales interactivos.

 Imágenes

Todos los recursos visuales ayudan a aumentar el interés de los usuarios. Esto también puede incluir videos, animaciones, mapas, gráficas, infografías, GIFs, ilustraciones, diagramas, etc.

Logotipo

Para que un sitio web tenga mayor identidad es vital que contenga el logotipo que represente a la marca o empresa.

Diseño gráfico

Este elemento engloba todo lo relacionado con cómo se ve el sitio web y su apariencia: colores, formas, tipografías, tamaños, etc. Como ya lo mencionamos, el frontend son todos los elementos y componentes visibles para los usuarios, y utilizan lenguajes de diseño como CSS, HTML y JavaScript.

Algunos ejemplos de frontend son los siguientes:

* Optimización de motores de búsqueda (SEO).
* Accesibilidad (reconocimiento de voz, conversión de texto a voz).
* Funcionalidad en todos los navegadores y tamaños de pantalla (computadoras de escritorio, teléfonos móviles y tabletas).
* Velocidad (cuanto más rápido cargue el sitio, mejor).
* Rendimiento del sitio web por medio de la limpieza del código.

**Backend**

El backend es el encargado de procesar toda la información que alimenta a un frontend. Se compone de marcos, bases de datos o códigos. Para que un sitio web o aplicación opere efectivamente, se requiere mucha información y datos que se almacenan en «la parte trasera» de un sistema informático. En oposición al frontend, el usuario no puede ver o acceder a esta información.

Uso

El backend son todos los códigos ocultos que sirven para que una página web o aplicación funcione correctamente. Además, de su estructura y organización depende la experiencia de usuario. De igual forma, el backend se encarga de optimizar otros elementos y recursos como la seguridad y privacidad en un sitio web o aplicación.

Elementos

El backend se constituye por lenguajes de programación como PHP, Python y C++ y frameworks. Los servidores controlan cómo los usuarios acceden a los archivos. Las bases de datos son colecciones de datos organizadas y estructuradas.

La seguridad es uno de los elementos más importantes dentro de un sitio web, pues garantiza que los visitantes y su información estén seguros. Esto también incluye evitar, en lo posible, ciberataques.

Ejemplos de aplicación

Algunos ejemplos para terminar de comprender el término son los siguientes:

* Inicio de sesión. Cuando una persona accede a un sitio web o aplicación utiliza un correo electrónico y contraseña, esta información es validada y resguardada por el servidor, que consulta su base de datos y así identifica y permite el acceso al usuario.
* Carrito de compras. Este elemento permite la compra de productos en línea y sirve para facilitar la selección de diferentes productos o servicios que algún usuario desee comprar.
* Cookies. Muchos sitios utilizan cookies para realizar un seguimiento de aquello que los usuarios vieron anteriormente, lo que les permite sugerir otros contenidos (o productos) de interés.
* CMS. Un sistema de gestión de contenidos permite al propietario de un sitio web actualizar la información sin tener que modificar el código HTML.
* Formularios de contacto. Si un visitante del sitio web se interesa por recibir más información o ponerse en contacto, se debe contar con un elemento que sea capaz de vincular al usuario con la empresa.

Diferencias entre frontend y backend

Por sus aplicaciones y características podemos afirmar que el frontend comprende todas las acciones relacionadas con el diseño de experiencia que tendrá un visitante a una página web, mientras que el backend se refiere a la estructuración del sitio y la programación de sus funcionalidades principales.

<https://blog.hubspot.es/website/frontend-y-backend>

## Aplicación web

Una aplicación web es un software que se ejecuta en el navegador web. Las empresas tienen que intercambiar información y proporcionar servicios de forma remota. Utilizan aplicaciones web para comunicarse con los clientes cuando lo necesiten y de una forma segura. Las funciones más comunes de los sitios web, como los carros de compra, la búsqueda y el filtrado de productos, la mensajería instantánea y los canales de noticias de las redes sociales, tienen el mismo diseño que las aplicaciones web. Le permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software.

## Beneficios

Las aplicaciones web tienen muchos beneficios, y casi todas las empresas grandes las utiliza como parte de sus ofertas para usuarios. A continuación, se muestran alguno de los beneficios comunes asociados a las aplicaciones web.

### **Accesibilidad**

Las aplicaciones web son accesibles desde todos los navegadores web y desde diferentes dispositivos personales y empresariales. Equipos de diferentes ubicaciones pueden acceder a documentos compartidos, sistemas de administración de contenidos y otros servicios empresariales a través de aplicaciones web basadas en suscripciones.

### **Desarrollo eficiente**

Tal y como se detalló, el proceso de desarrollo para aplicaciones web es relativamente sencillo y rentable para las empresas. Los equipos pequeños pueden lograr ciclos de desarrollo cortos, lo que hace que las aplicaciones web sean una manera eficiente y asequible de desarrollar programas de computación. Además, dado que la misma versión funciona en todos los navegadores y dispositivos modernos, no tendrá que crear un número elevado de iteraciones diferentes para varias plataformas.

### **Simplicidad para el usuario**

Los usuarios no tienen que descargar las aplicaciones web, lo que hace que sean fáciles de acceder a la vez que se prescinde de mantenimiento y capacidad en el disco duro por parte del disco duro. Las aplicaciones web reciben actualizaciones de software y seguridad de manera automática, lo que significa que siempre están actualizadas y presentan menor riesgo de sufrir brechas de seguridad.

### **Escalabilidad**

Las empresas que utilizan aplicaciones web pueden agregar usuarios cuando sea necesario, sin necesidad de infraestructura adicional o hardware costoso. Además, la mayor parte de los datos de las aplicaciones web se almacena en la nube, lo que significa que su empresa no tendrá que invertir en capacidad de almacenamiento adicional para ejecutar aplicaciones web.

## Algunas aplicaciones web más comunes

Hay muchos tipos de aplicaciones web. Estos son algunas de las más comunes.

### **Aplicaciones web para el trabajo colaborativo**

Las aplicaciones web para el trabajo colaborativo permiten a miembros de equipos acceder a documentos, calendarios compartidos, servicios de mensajería instantánea empresarial y otras herramientas empresariales.

### **Aplicaciones web de comercio electrónico**

Las aplicaciones web de comercio electrónico, como Amazon.com, permiten a los usuarios navegar, buscar y pagar productos en línea.

### **Aplicaciones web de correo electrónico**

El uso de aplicaciones web de correo electrónico para acceder al correo electrónico está muy extendido entre empresas y usuarios personales. A menudo, estas aplicaciones incluyen otras herramientas de comunicación, como mensajería instantánea y videoconferencia.

### **Aplicaciones web de banca en línea**

Los usuarios empresariales y personales utilizan con frecuencia aplicaciones web de banca en línea para acceder a sus cuentas y a otros productos financieros, como préstamos e hipotecas.

### **Documentación técnica**

### **Puede utilizar aplicaciones web para crear y compartir documentación técnica, como manuales de usuario, guías prácticas y especificaciones de dispositivos**

## Funcionamiento de las aplicaciones web

Las aplicaciones web tienen una arquitectura cliente-servidor. Su código se divide en dos componentes: scripts del lado del cliente y scripts del lado del servidor.

### **Arquitectura del lado del cliente**

El script del lado del cliente se encarga de la funcionalidad de la interfaz de usuario, como los botones y los cuadros con menús desplegables. Cuando el usuario final hace clic en el enlace de la aplicación web, el navegador web carga el script del lado del cliente y renderiza los elementos gráficos y el texto para la interacción del usuario. Por ejemplo, el usuario puede leer contenidos, ver videos o cumplimentar la información de un formulario de contacto. Las acciones como hacer clic en el botón de enviar se dirigen al servidor como una solicitud del cliente.

### **Arquitectura del lado del servidor**

El script del lado del servidor se encarga del procesamiento de datos. El servidor de la aplicación web procesa las solicitudes del cliente y envía una respuesta de vuelta. Las solicitudes suelen ser obtener más datos, editar datos o guardar nuevos datos. Por ejemplo, si el usuario hace clic en el botón Leer más, el servidor de la aplicación web enviará contenido al usuario. Si el usuario hace clic en el botón Enviar, el servidor de la aplicación guardará los datos del usuario en la base de datos. En algunos casos, el servidor completa la solicitud de datos y envía la página HTML completa al cliente. Esto se llama renderizado del lado del servidor.

## Diferencia entre una aplicación web y un sitio web

Cuando se inventó el Internet, los sitios web tenían mucho menos funcionalidad que las aplicaciones web. Únicamente podían brindar información a los usuarios a través de contenido estático. Era necesario instalar y ejecutar software con funcionalidad compleja. Las aplicaciones web fueron diseñadas para cubrir ese vacío existente entre el software y los sitios estáticos. Contaban con la funcionalidad y elementos de usuario interactivos del software, pero se entregaban mediante una URL de navegador web.

Sin embargo, la tecnología ha evolucionado de manera significativa desde entonces. La mayoría de los sitios web modernos son aplicaciones web compleja s en cuanto a su diseño.

<https://aws.amazon.com/es/what-is/web-application/>

HTML

HTML es el acrónimo de Hypertext Markup Language, que es el lenguaje estándar para desplegar documentos en un navegador web. Los navegadores web reciben documentos

HTML de un servidor web o del almacenamiento local. Los documentos almacenados por un servidor web pueden ser accedidos mediante un dominio, es decir, estos documentos conforman un sitio web en producción. Los documentos locales pueden ser accedidos mediante un servidor de aplicaciones local con dominio estándar localhost, es decir, estos documentos conforman un sitio web en desarrollo.

HTML describe la estructura de una página web. Los elementos HTML son los componentes básicos de las páginas HTML. En la página web, se puede incrustar diferentes componentes HTML, tales como imágenes y formularios. Además, HTML proporciona un medio para crear documentos estructurados para texto como encabezados, párrafos, lis-tas, enlaces, citas etc. Los elementos HTML están descritos por etiquetas que se encierran por los símbolos: < y >. Algunas etiquetas incrustan información y requieren el símbolo / antes del símbolo > (por ejemplo <img />) mientras que otras encierran el contenido entre dos etiquetas (por ejemplo <p></p>) [12).

HTML puede incrustar programas escritos en un lenguaje de secuencias de comandos como JavaScript, lo que afecta el comportamiento y el contenido de las páginas web.

La inclusión de CSS define el aspecto y el diseño del contenido. El World Wide Web Consortium' (W3C) fue la organización que inicialmente mantuvo HTML y mantiene

actualmente de los estándares CSS desde 1997.  
HTML ha evolucionado con varias versiones actualizadas. Cada versión ha permitido crear páginas web de una manera mucho más fácil, estética y eficiente. HTML 1.0 se lanzó en 1993 con la intención de compartir información que pueda ser legible y accesible a través de navegadores web. HTML 2.0 fue publicado en 1995 y contiene todas las características de HTML 1.0 junto con algunas características adicionales que permaneció como el lenguaje de marcado estándar para diseñar y crear sitios web hasta enero de 1997.

HTML 3.0 incluyó nuevas características mejoradas de HTML brindando características más potentes para el diseño de páginas web. Sin embargo, estas nuevas características ralentizaron el navegador al aplicar mejoras adicionales. HTML 4.01 es ampliamente utilizado y era una versión exitosa de HTML antes de HTML 5.0, el cual actualmente se usa en todo el mundo. Se puede decir que HTML 5 es una versión extendida de HTML 4.01 que se publicó en el año 2012.

CSS

Cascading Style Sheets (CSS) es un lenguaje de hojas de estilo utilizado para describir la presentación de un documento escrito en HTML. CSS está diseñado para permitir la separación del contenido, colores y fuentes en una página web. Esta separación puede mejorar la accesibilidad al contenido, proporcionar más flexibilidad y control en la especificación de las características de presentación. Además, permitir que múltiples páginas web compartan el formato al especificar el CSS relevante en un archivo con extensión css y reducir la complejidad y la repetición en el contenido estructural CSS. La W3C mantiene y proporciona las especificaciones de CSS.

JavaScript

JavaScript es un lenguaje de secuencias de comandos interpretado Junto con HTML y CSS, JavaScript es una de las tecnologías principales de la World Wide Web Foundation.

JavaScript habilita páginas web interactivas y es una parte esencial de las aplicaciones web.

Los navegadores web tienen un motor de JavaScript dedicado para ejecutarlo. JavaScript admite estilos de programación impulsados por eventos, funcionales e imperativos. Tiene API para trabajar con texto, matrices, fechas, expresiones regulares y DOM.

Inicialmente solo se implementó en el lado del cliente en los navegadores web, sin em-bargo, los motores de JavaScript ahora están integrados en muchos otros tipos de software incluido el lado del servidor en servidores web, y/o bases de datos. JavaScript surgió gracias a la influencia de lenguajes de programación como Self y Scheme. Además, el formato de texto JSON, el cual es utilizado para almacenar estructuras de datos en archivos o transmitirlos a través de redes, se basa en JavaScript.

PHP

PHP3 es un lenguaje de programación que permite incorporar HTML, el cual se usa principalmente para aplicaciones web dinámicas. De esta manera, PAP puede intercalarse con HTML, lo que simplifica la construcción de páginas web. PHP es un lenguaje que se interpreta en un explorador mediante Apache, el cual actúa como servidor de aplicaciones.

Entonces, PAP no es un lenguaje que se compila y genera archivos ejecutables independientes. La sintaxis de PHP es conocida debido a que toma la mayor parte de C, Java y Perl. PHP es un lenguaje de código abierto y se ejecuta en la mayoría de los sistemas operativos y con la mayoría de los servidores web.

PHP fue escrito en el lenguaje de programación C por Rasmus Lerdorf en 1994 para ser usado en el monitoreo de su currículum en línea e información personal. Por esta razón, originalmente PHP era el acrónimo de "Personal Home Page". Lerdorf combinó PHP con su propio intérprete de formularios (generalmente conocido como PHP 2.0) el 8 de junio de 1995. Sin embargo, Zeev Suraski y Andi Gutmans reconstruyeron el núcleo de PHP liberando el resultado actualizado como PHP/FI 2.0 (Personal Home Page/Forms Inter-preter) en 1997. En ese momento, el acrónimo PHP se cambió formalmente a "Hyper Text Preprocessor".

En 1998, se lanzó PHP 3.0 y esta se convirtió en la versión más utilizada. Una de las mayores fortalezas de PHP 3.0 fue su fuerte capacidad de extensibilidad. Además de proporcionar a los usuarios finales una interfaz madura para múltiples bases de datos.

protocolos y API, la facilidad de extender el lenguaje motivó a los desarrolladores para que desarrollaran y enviaran nuevos módulos. Otras características clave introducidas en PHP 3.0 incluyen soporte de programación orientada a objetos y una sintaxis de lenguaje mucho más potente y consistente. En junio de 1998, con muchos nuevos desarrolladores de todo el mundo uniéndose al esfuerzo, PHP 3.0 fue anunciado por el nuevo equipo de desarrollo de PHP como el sucesor oficial de PHP /FI 2.0.

Después de aproximadamente nueve meses de pruebas públicas abiertas, cuando se anunció el lanzamiento oficial de PHP 3.0, ya estaba instalado en más de 70.000 dominios en todo el mundo y ya no estaba limitado a los sistemas operativos compatibles con POSIX.

Una parte relativamente pequeña de los dominios que informan que PHP está instalado se alojó en servidores que ejecutan Windows 95, 98, NT y Macintosh. En su apogeo, PHP 3.0 se instaló en aproximadamente el 10% de los servidores web en Internet. De esta forma, PHP en todas sus versiones se encontraba instalado en más del 80% de los servidores web

del mundo.

PHP 4.0 fue lanzado en mayo de 2000 con un nuevo núcleo conocido como Zend Engine

1.0. PHP 4.0 mejoró la velocidad y la confiabilidad con respecto a PHP 3.0. Además, PHP

4.0 agregó referencias, el tipo primitivo de dato Boolean, el soporte COM en Windows, el búfer de salida, muchas nuevas funciones de matrices, sesiones HTTP, entre servicios que fueron muy importantes en su época.

PHP 5.0 se lanzó en julio de 2004 con la versión actualizada de Zend Engine 2.0. Entre las muchas características nuevas en PHP 5.0 están: SQLite, soporte para nuevas funciones de MySQL, manejo de excepciones usando estructura try.. catch, soporte SOAP integrado, biblioteca de filtros (en PHP 5.1), iteradores, entre muchos otros servicios que son de gran interés en el desarrollo de aplicaciones web en la actualidad.

El desarrollo de PHP 6.0 se inició en octubre de 2006. El cambio más significativo en esta nueva versión es el soporte nativo para Unicode. Aunque PHP todavía se usa principalmente para la generación de páginas web del lado del servidor, también se puede usar para ejecutar la línea de comandos Scripting o para crear aplicaciones gráficas con la

ayuda de GTK+\*.

1.7 Bootstrap

Bootstrap® fue creado en Twitter a mediados de 2010. Inicialmente, Bootstrap era conocido como Twitter Blueprint. Tras unos pocos meses de desarrollo, Twitter celebró su primera semana de hackeo y el proyecto explotó a medida que los desarrolladores de todos los niveles de habilidad saltaban sin ninguna guía externa. Sirvió como la guía de estilo para el desarrollo de herramientas internas en la compañía durante más de un año antes de su lanzamiento público.

Bootstrap fue lanzado originalmente en agosto de 2011. En enero de 2012, se lanzó Bootstrap 2.0 que agregó soporte incorporado para Glyphicons. Bootstrap 2 soporta diseño web responsivo que significa que el diseño de las páginas web se ajusta dinámicamente, teniendo en cuenta las características del dispositivo utilizado. Bootstrap 3, lanzada en agosto de 2013, fue reescrita con el fin de que responda de forma predeterminada para desarrollo en dispositivos móviles. En agosto de 2015 se lanzó la versión alfa de Bootstrap

4. En agosto de 2017, se lanzó la versión beta de Bootstrap 4. Finalmente, Bootstrap 4 fue terminado en enero de 2018. Bootstrap 4 es compatible con las últimas versiones de Google Chrome, Firefox, Internet Explorer, Opera y Safari (excepto en Windows). Además, es compatible con IE9 y la última versión de soporte extendido de Firefox.

Fernández, H. F., & Rodríguez, J. H. (2021). *Aplicaciones web con PHP*. Ediciones de la U.

# Tercera Parte: Metodología

## Sujetos y Fuentes de Información

### Sujetos

### Fuentes de Información

## Tipo de investigación

### Diseño de investigación

## Instrumentos y Técnicas utilizadas en la recopilación de los datos.

### Método de investigación

# Cuarta Parte: Conclusiones y Recomendaciones

## Conclusiones

## Recomendaciones

# Quinta Parte: Propuesta

## Análisis de requerimientos

## Fase de preparación

## Fase de instalación

### Sub faces necesarias

## Costos Estimados

## Cronograma

# Bibliografía

# Glosario

# Anexos