

Propuesta Técnica Final

Estructurar el Modelo de Datos del Software

Harold Giovanni Peñaloza Alarcon

Dajaryth Yenerys Hernández Cuellar

María José Mendoza Vargas

Juan José Gómez Quintero

Edwin Felipe Montenegro Alfonso

Servicio Nacional de Aprendizaje SENA

Análisis y Desarrollo de Software-ADSO

Centro de Diseño y Metrología-CDM

2873816

Edwin Alexander Duran García

20 septiembre 2024

Bogotá D.C

Tabla de contenido

Introducción	6
Diseño Web.....	7
Front End	7
HTML	7
CSS	7
JavaScript.....	8
Back End.....	8
Base de datos.....	8
Framework	9
Lenguaje de programación.....	9
Programación Orientado a objetos.....	9
• Herencia	9
• Encapsulación:	9
• Abstracción:	9
• Polimorfismo:.....	9
• Csharp.....	10
Protocolos	10
OAuth:.....	10
SSL:.....	10
TLS:	11
HTTP.....	11
HTTPS	11
Arquitectura Web.....	11

Modelo vista controlador	12
• Modelo:	12
• Vista:	12
• Controlador:	12
Modelo Web Cliente-Servidor	12
• Administración Central:	12
• Derechos de acceso controlados globalmente:.....	12
• Un solo servidor para muchos clientes:	13
Patrones de Diseño.....	13
Estructural	13
Comportamental.....	14
Patrón de Comando.....	14
Creacionales	14
Singleton:	14
Bases de Datos	14
Relacional	14
No relacional	15
Diccionario de Datos.....	15
Estructura de datos	16
Descripción detallada del Diccionario de Datos	16
Nombre empresa:	16
Key de acceso	17
Nombre:	17
Informe tipo 1 e informe tipo 2:.....	17

Tipos de equipos:	17
Teléfono y celular:	17
Correo:	17
Bases de Datos Orientada a Objetos	17
Uso Potencial de la Base de Datos.....	18
Gestión de clientes	18
Gestión de técnicos:	18
Registro de servicios:	18
Generación de reportes:	18
Diagrama de Casos de Uso	18
Diagrama de casos de uso alto nivel	19
Descripción.	19
Diagrama de casos de uso (extends e include).....	20
Descripción.	21
Diagramas de Clases	22
Descripción	23
Diagrama de Estados.....	24
Descripción	25
Modelo entidad relación	25
Descripción	26
Diagrama Relacional.....	27
Descripción	28
Conclusión	29
Referencias Bibliográficas	30

Tabla de Ilustraciones

Ilustración 1 Diccionario de datos	16
Ilustración 2 Casos de uso alto nivel.....	19
Ilustración 3 Casos de usos extends e include	21
Ilustración 4 Diagramas de clases	23
Ilustración 5 Diagrama de Estados	24
Ilustración 6 Modelo entidad relación	26
Ilustración 7 Diagrama Relacional.....	28

Introducción

El desarrollo de aplicativo web implica una interacción de tecnologías y conceptos claves. Este trabajo se centra en analizar los elementos importantes que fundamentan la construcción de un aplicativo web moderno, desde su diseño conceptual hasta su implementación técnica.

Comenzaremos con la importancia del diseño web como marco conceptual para organizar y planificar proyectos. Profundizaremos en el desarrollo front-end, que se encarga de la interfaz de usuario y la experiencia del cliente. En esta sección, se detallarán las tecnologías clave: HTML, CSS y JavaScript.

Por último, se abordará aspectos relacionados con la arquitectura web, los protocolos de comunicación y los patrones de diseño. Se presentarán diagramas UML para mostrar conceptos como el modelo vista controlador, el modelo cliente-servidor y el modelo entidad-relación.

Este informe tiene como objetivo brindar una descripción detallada e integral del desarrollo de este aplicativo web. Al comprender todo lo dicho, los desarrolladores podrán construir aplicaciones web robustas, escalables y con una excelente experiencia de usuario.

Diseño Web

El diseño web es muy importante en el desarrollo web ya que permite organizar y planificar en cómo dar forma a las ideas planteadas en un proyecto, no siendo solo en la parte visual sino también en el área de trabajo sea más fácil de desarrollarse entre las distintas personas o grupo de trabajo. (Pressman & Maxim, 2021, p.156)

Front End

El front end de un proyecto de un proyecto web es fundamental para la interacción entre usuario y aplicativo, normalmente es todo aquello que pasa en el lado del usuario. Estas herramientas son las principales a la hora de trabajar esta parte del proyecto, el equipo de trabajo encargado de esta área tendrá que pensar en el usuario para que la experiencia de este sea óptima. Actualmente se está planeando en nuestro proyecto en como brindar una experiencia agradable a los clientes y técnicos ya que es importante que estos tienen que hacer los diferentes procesos de forma sencilla y eficaz, lo que es necesario estar implementando y probando estas herramientas. (Escarcena, 2020, p.13).

HTML

El Lenguaje de Marcado de Hipertexto (HTML) es un código utilizado para estructurar y dar una estructura que funciona como un esqueleto que permite tener todos los contenidos que se van a visualizar en una página web o aplicativo web. Normalmente es usado para estos proyectos y resulta ser muy útil para nuestro proyecto ya que permite la implementación de todos los datos e información que se requieran para el proyecto. (Fernández, 2020, p.24)

CSS

Las hojas de estilo cascada o por sus siglas en ingles Cascading Style Sheets (CSS) es una herramienta muy útil para la creación de páginas y aplicativos ya que permite un mayor manejo de los datos de forma que sean más atractivos para los usuarios y estos puedan tener una mejor interactividad con la página o aplicativo web. Esta herramienta es muy útil para cualquier tipo de proyecto web y por esa razón se va a

implementar para nuestro proyecto ya que hará que esta se vea más atractiva y manejable para los usuarios. (Fernández, 2020, p.24)

JavaScript

Este es un lenguaje de programación muy utilizado en el desarrollo de páginas y aplicativos webs, normalmente es usado para dar interactividad dinámica a un proyecto para que al usuario le sea más fácil interactuar con los distintos elementos que contiene el aplicativo o página web. Este lenguaje es fundamental para el desarrollo de nuestro proyecto ya que permite que los usuarios puedan acceder a los diferentes recursos que ofrece el aplicativo de manera rápida y eficiente. (Escarcena, 2020, p.20).

Back End

El back end normalmente se refiere a la parte sistema servidor, el back end es fundamental para que un aplicativo o app web pueda funcionar, esto funciona a través de un procesamiento de datos que hace el aplicativo para que se mande al servidor y finalmente una base de datos que conecte todo este sistema para que cuando el usuario requiera algún dato o información este pueda acceder sin ningún problema. Los encargados de esta parte cuentan con un amplio conocimiento en bases de datos, frameworks y lenguajes de programación. El front end y el back end trabajan juntos ya que con estos dos se logra que un proyecto se desarrolle y funcione correctamente. Es importante tener en cuenta para nuestro proyecto el uso de estas dos tecnologías ya que básicamente sin estas no podremos lograr desarrollar el proyecto planteado. (Escarcena, 2020, p.23).

Base de datos

Una base de datos es una herramienta que permite recopilar y organizar grandes cantidades de información, en este caso de manera electrónica, estas permiten una mayor facilidad en el acceso, búsqueda, edición y administración de datos en los distintos sistemas informáticos. Esta es la clave fundamental hoy en día para que un proyecto de software funcione ya que la mayoría deben tener en cuenta el aspecto de recopilación y edición de datos. (Marqués, 2009, p.10)

Framework

El framework comúnmente es conocido por ser el marco o esquema de trabajo lo que ofrece una estructura o plantillas que ayuda a la creación y organización de un proyecto de software, utilizar esta herramienta puede facilitar y optimizar esta tarea de desarrollo. (Gutiérrez & López, 2017, p.79)

Lenguaje de programación

Los lenguajes de programación son fundamentales para el desarrollo de proyectos de software para todo tipo de sistemas electrónicos, este es una serie de instrucciones y términos que permite la comunicación entre usuario y máquina, este siendo muy útil a la hora de hacer las distintas tareas para un sistema electrónico. Actualmente hay varios lenguajes de programación orientado a aspectos según la necesidad del cliente o del proyecto a desarrollar, sin esta herramienta sería muy complicado crear un software o un hardware por lo que en todo proyecto se tiene que implementar esta herramienta para que el trabajo sea mucho más fácil y eficiente. (Rinaudo & Pantaleo, 2018, p.46)

Programación Orientado a objetos

Este son los lenguajes son los enfocados en organizar los datos u objetos del software. La programación orientada a objetos tiene cuatro elementos fundamentales para un buen desarrollo son:

- ***Herencia:*** Las clases secundarias heredan datos y comportamientos de la clase principal.
- ***Encapsulación:*** Contener información en un objeto, exponiendo solo la información seleccionada.
- ***Abstracción:*** Exponer solo métodos públicos de alto nivel para acceder a un objeto.
- ***Polimorfismo:*** Muchos métodos puedan realizar la misma tarea. (Sznajdleder, 2017, p.73)

Los principales lenguajes de programación orientados a objetos más usados son, Java, C++, C#, Python, Ruby, PHP, entre otros. Actualmente estos lenguajes son muy usados para este tipo de programación orientado a objetos porque son fáciles y sencillos de usar, este tipo de programación es también muy utilizada en muchos proyectos por eso

la razón de que haya varios lenguajes basados a este tipo de programación. Los lenguajes de programación son muy útiles y hay varios que dependen de nuestro contexto y problema podamos usarlo de buena forma. (Sznajdleder, 2017, p.51)

- **Csharp:** A la hora de programar la creación de clases y objetos nos facilita el proceso de programación ya que vamos a manejar unos módulos dentro de la aplicación web que cuentan con unos sub módulos y con este paradigma así que nos facilita el agruparlos y en la reutilización de código. (Sznajdleder, 2017, p.51)

Protocolos

Los protocolos son una serie de reglas y medidas de un sistema para la comunicación entre estos, siendo importantes también para tener una buena seguridad en la red y a la hora de transmitir datos entre entidades. Estos protocolos son utilizados en muchos proyectos y páginas webs de toda internet por lo que es importante tener esto distintos protocolos para que el desarrollo de nuestro proyecto sea seguro y proteja la información de nuestros clientes y distintos usuarios. (Katz, 2013, p.219).

OAuth: Significa “Open Authorization” (autorización abierta) este básicamente lo que hace es permitir que una página o aplicativo web autorice a acceder a los recursos que se encuentran alojados en estos, como por ejemplo API o datos de usuario. Este concede unos tokens que dan el acceso normalmente estos tokens pueden tener caducidad lo que hace que no se vuelva a acceder a esta información tiempo después. En nuestro proyecto sería muy útil poder tener un mayor control de las autorizaciones que se den ya que pasado un tiempo el cliente no podrá volver a acceder a esta información ya que ya se habrá expirado el tiempo que tenía para poder hacer todos los distintos procesos del aplicativo. (Gutiérrez, 2020, p.20)

SSL: Este es un certificado que significa “Secure Sockets Layer” que en español quiere decir capa de sockets seguros, normalmente son utilizados para establecer una conexión cifrada entre el navegador o el ordenador de un usuario y un servidor o sitio web, protegiendo así la conexión entre estos. Además de que ayuda a que un externo pueda ver o interceptar la información que se está transmitiendo. Actualmente tener este certificado ayuda en la seguridad web y genera una mayor confianza en los usuarios a la hora de utilizar una página web. (Gutiérrez, 2020, p.21)

TLS: Este es un protocolo de seguridad que ofrece privacidad e integridad de los datos para la comunicación en internet, lo que genera webs más seguras y genera una encriptación de datos entre las comunicaciones. Este protocolo es muy útil y fundamental en páginas o aplicativos webs de internet, por lo que hay que investigar en cómo se puede aplicar de forma eficiente en nuestro proyecto. (Katz, 2013, p.115).

HTTP

El protocolo de transferencia de hipertexto es un protocolo de capa aplicación que permite cargar páginas web para transferir información entre los dispositivos conectados a una red. Este protocolo es fundamental en una página web para la carga de datos, por lo que es importante tenerlo en cuenta en cualquier proyecto. (Katz, 2013, p.267)

HTTPS

Este es parecido al anterior con la característica de que hace que sea más seguro que el HTTP, este es importante para proteger los datos y encriptarlos de modo que sea más seguro para el usuario. Si un sitio va a manejar datos importantes de los usuarios como por ejemplo un registro, una cuenta bancaria o un servicio de correo electrónico. Con esto es importante asegurar los datos de nuestros clientes y usuarios así que hay que tener en cuenta cómo proteger estos datos. (Katz, 2013, p.267)

Arquitectura Web

La arquitectura web es muy importante para el desarrollo de una página web, normalmente trata los elementos de la planificación y el diseño de los componentes técnicos, funcionales, y visuales del sitio web mucho antes de ponerlo en práctica. Este es muy importante para que los diseñadores y desarrolladores puedan hacer su trabajo. Contar con una buena arquitectura de diseño web ayuda a tener una mejor experiencia de usuario y que sea Google permita rastrear la página de manera más eficaz, todo esto es importante ponerlo en práctica en nuestro proyecto si queremos tener una mejor experiencia para nuestros usuarios y para poder desarrollar nuestro trabajo de manera más

ordenado y eficaz sin perder tiempo valioso lo que nos permite estar enfocados en otros factores importantes. (López, 2016, p.90)

Modelo vista controlador

El modelo de vista controlador es un tipo de patrón de diseño de software que es muy usado para poder implementar interfaces de usuario, datos y lógica de control. Este modelo es importante para proporcionar una mejor división de trabajo y de una mejora de mantenimiento, normalmente la modelo vista controlador se divide en tres características importantes que son:

- **Modelo:** Manejo de datos y lógica de negocios.
- **Vista:** Se encarga del diseño y presentación.
- **Controlador:** Enruta comandos a los modelos.

Este modelo de vista controlador es importante para nuestro proyecto ya que nos va a permitir en tener una mejor división y control del trabajo en el front end y back end dividiéndolo en lógicas diferentes para que el proceso sea enfocado y detallado. (Flórez & Hernández, 2021, p.273)

Modelo Web Cliente-Servidor

El modelo cliente servidor es aquel que permite la interacción y comunicación del usuario y el servidor que este acepta las peticiones, las procesa y proporciona la respuesta a la solicitud. Este modelo es muy importante ya que es el que permite que las solicitudes que haga el usuario sean respondidas por lo tanto es importante tener un servidor físico que ayude a que estos procesos sean hechos. Contar con una buena administración del servidor tiene una serie de ventajas que son. (López, 2016, p.93)

- **Administración Central:** Esto simplifica la administración de la base de datos o los diferentes recursos de la página web.
- **Derechos de acceso controlados globalmente:** Además de una buena administración también permite una buena protección de accesos a los datos controlando quien los puede ver y quien los puede administrar.

- ***Un solo servidor para muchos clientes:*** Los usuarios pueden estar simultáneamente en el servidor sin que este tenga problemas por lo que es importante para varias páginas web más públicas.

Patrones de Diseño

Los patrones de diseño nos ayudan a resolver un problema en un contexto dado, lo que nos da plantillas para dar solución a este problema en dicho contexto. Esto nos da soluciones fáciles y sencillas sin importar el lenguaje o métodos que vayamos a utilizar. El patrón de diseño cuenta con unas características importantes para un buen patrón de diseño que son:

- Estandarizar el lenguaje entre programas
- Evitar perder tiempo en soluciones a problemas ya resueltos o conocidos
- Crear código reusable

A lo largo del tiempo ha habido muchos patrones de diseño lo importante es saber en cómo utilizarlos y aprovecharlos al máximo lo que nos permitirá escribir código de la manera correcta y ayudaran a resolver problemas de una manera sencilla y eficaz.

(Pressman & Maxim, 2021, p.295)

Dentro de los patrones de diseño se encuentran:

Estructural

El patrón tipo fachada nos ayuda encapsular toda la lógica de la interfaz, para exponerles de una manera más simple y fácil y amigable. Se tiene que realizar la abstracción para definir todas las operaciones complejas para realizar la encapsulación.

(Pressman & Maxim, 2021, p.291)

Comportamental

Patrón de Comando

Este patrón encapsula una solicitud como un objeto. Permite parametrizar los objetos con operaciones y desacoplar al emisor de las acciones concretas.

Imagina una aplicación de procesamiento de texto. Los comandos (como “Copiar”, “pegar” o “deshacer”) se representan como objetos. El usuario interactúa con estos comandos, y el editor de texto los ejecuta sin saber que exactamente qué hacen. (Pressman & Maxim, 2021, p.291)

Creacionales

Singleton: es una orientación de cómo solucionar un problema como, por ejemplo, que en alguna parte del diseño se necesita acceder a una clase, un objeto de única instancia, es decir, solo necesitamos usar un objeto siempre en todas las invocaciones.

Encapsular nuestro método de creación por defecto de forma privada

Crear en nuestro código un método de construcción de forma estática para que pueda ser accedido por los demás y en la creación estática por dentro del método se hará el llamado al constructor privado. Para que sea únicamente responsabilidad del método estático. (Jiménez De Parga, 2021, p.181)

Bases de Datos

Relacional

Es aquella que almacena y proporciona acceso a los datos relacionados entre sí, estas se basan en el modelo una manera intuitiva y directa de representar datos en tablas. Estas además cuentan con características importantes como, representar datos en tablas que estas a su vez cuentan con filas y columnas, cada tabla cuenta con ID o clave única que las representa que además permite que se relacionen entre otras tablas. Actualmente este tipo de bases de datos es muy útil para muchos proyectos, pero nuestro proyecto

requiere un gran manejo de datos e información por lo que se ha optado en utilizar una base de datos no relacional. (Marqués, 2009, p.42)

No relacional

Las bases de datos no SQL o no relacionales son aquellas que están diseñadas específicamente para modelos de datos específicos y almacenan los datos en esquemas flexibles que se escalan con facilidad para aplicaciones modernas. Esto quiere decir que va a permitir almacenar grandes volúmenes de cantidades de datos, lo que permite una mayor flexibilidad y menor uso de espacio para almacenar estos datos. Este tipo de base de datos nos puede ser muy útil para nuestro ya que como se expuso se van a manejar grandes cantidades de datos necesitamos ahorrar espacio en el servidor, además que nuestro aplicativo web no está enfocado en tener una base de datos relacional. Este tipo de base de datos cuenta con algunos gestores útiles precisamente para el uso en el tipo de proyecto que se va a realizar. (Sarasa, 2016, p.25)

Diccionario de Datos

Para definir nuestro diccionario de datos primero vamos a visualizar más a fondo sobre de que trata este. Reinoso et al. (2010) dice que: "El diccionario de datos, también conocido como catálogo del sistema, es una base de datos que es permanentemente administrada y modificada por el DBMS. Esta base de datos se construye con todas las características del modelo relacional, es decir que es un conjunto de relaciones y éstas poseen claves primarias, foráneas, restricciones, etcétera." (p.38)

Es importante tener en cuenta que un diccionario de datos debe contar con: "Objetos que conforman cada base de datos (vistas, tablas, índices, procedimientos almacenados, etc.), que incluyen la fecha de creación, la definición estructural de cada uno, las relaciones con otros objetos, el propietario del objeto..." (p.38)

Estructura de datos

Es importante tener en cuenta la estructura de los datos y como estos se definen por eso Hernández y Baquero (2021) “Se trata de un conjunto de variables de un determinado tipo, agrupadas y organizadas de alguna manera para representar un comportamiento. Lo que se pretende con las estructuras de datos es facilitar un esquema lógico para manipular los datos en función del problema que haya que tratar y el algoritmo para resolverlo. En algunos casos, la dificultad para resolver un problema radica en escoger la estructura de datos adecuada. Y, en general, la elección del algoritmo y de las estructuras de datos que manipulará estará muy relacionada.” (p.37)

Descripción detallada del Diccionario de Datos

Campo	Tipo de Dato	Descripción
nombre_empresa	Cadena de texto (varchar)	Nombre de la empresa que utiliza el sistema.
key_de_acceso	Cadena de texto (varchar)	Clave única para acceder al sistema.
nombre	Cadena de texto (varchar)	Nombre del usuario (técnico o cliente).
informe_tipo_1	Archivo (blob)	Archivo del primer tipo de informe.
informe_tipo_2	Archivo (blob)	Archivo del segundo tipo de informe.
tipos_de_equipo	Cadena de texto (varchar)	Lista de los tipos de equipos manejados (Minisplit, Pisotecho, etc.).
telefono	Cadena de texto (varchar)	Número de teléfono de contacto.
celular	Cadena de texto (varchar)	Número de celular de contacto.
correo	Cadena de texto (varchar)	Dirección de correo electrónico de contacto.

Ilustración 1 **Diccionario de datos**

El diccionario de datos describe la estructura de una base de datos está relacionada con un sistema de gestión de información para una empresa que brinda servicios de mantenimiento o instalación de equipos, de tipo aire acondicionado.

Cada campo del diccionario representa un atributo de los datos almacenados y su significado es el siguiente:

Nombre empresa: Este campo almacena el nombre de la empresa que utiliza el sistema. Es un dato de tipo cadena de texto tipo varchar.

Key de acceso: Esta clave es única y sirve para identificar y autenticar a cada usuario del sistema, ya sea un técnico o un cliente. Al igual que el nombre de la empresa, es un dato de cadena de texto tipo varchar.

Nombre: En este campo se registra el nombre del usuario, lo que permite personalizar la experiencia del usuario dentro del sistema.

Informe tipo 1 e informe tipo 2: Estos campos almacenan archivos de tipo blob (Binary Large Object). Los archivos blob se utilizan para guardar datos binarios, como imágenes, documentos y en este caso, informes. La existencia de dos campos para diferentes tipos de informes que el sistema.

Tipos de equipos: Este campo contiene una lista de los tipos de equipos que la empresa maneja, como minisplits o equipos de piso techo. Esta información es útil para tener en cuenta los servicios y generar reportes específicos según el servicio prestado.

Teléfono y celular: Estos campos almacenan los números de teléfono de contacto del usuario, tanto el fijo como el móvil.

Correo: En este campo se registra la dirección de correo electrónico del cliente, lo que permite la comunicación entre el técnico y el cliente para la respectiva revisión de los informes.

Bases de Datos Orientada a Objetos

Las bases de datos orientada a objetos son muy utilizadas hoy en día ya que permiten encontrar una solución a un problema fácilmente, según Reinoso et al (2010) dice que: "El objetivo principal de este tipo de desarrollo es dejar la etapa en la que la construcción del software es una labor de artesanos y pasar a la etapa en la que se pueda tener fábricas de software, con gran capacidad de reutilización de código y con metodologías eficientes y efectivas que se apliquen al proceso de producción"(p.239)

Uso Potencial de la Base de Datos

Con base en esta estructura de datos, se puede saber que la base de datos puede utilizarse para las siguientes funcionalidades:

Gestión de clientes: Almacenar información de contacto de los clientes, como nombre, teléfono, correo electrónico y dirección.

Gestión de técnicos: Registrar los datos de los técnicos que trabajan para la empresa, incluyendo su identificación o key de acceso.

Registro de servicios: Documentar los servicios realizados, incluyendo el tipo de equipo, las acciones realizadas y los informes generados.

Generación de reportes: Crear reportes personalizados para analizar el rendimiento de la empresa, la satisfacción del cliente y otras características importantes.

Diagrama de Casos de Uso

Estos diagramas son muy importantes en cualquier proyecto ya que nos permiten definir qué se hará y como interactúa con los demás usuarios, para definir mejor que son los casos de uso según Fontela (2010):

"Los casos de uso son herramientas de modelización de requisitos funcionales, que preceden (en el tiempo) y exceden (en alcance) a UML. Pero fueron UML y UP los que les dieron mayor difusión. Tal vez por esa asociación con UP, los métodos ágiles evitan hablar de casos de uso". (p.28)



Diagrama de casos de uso alto nivel

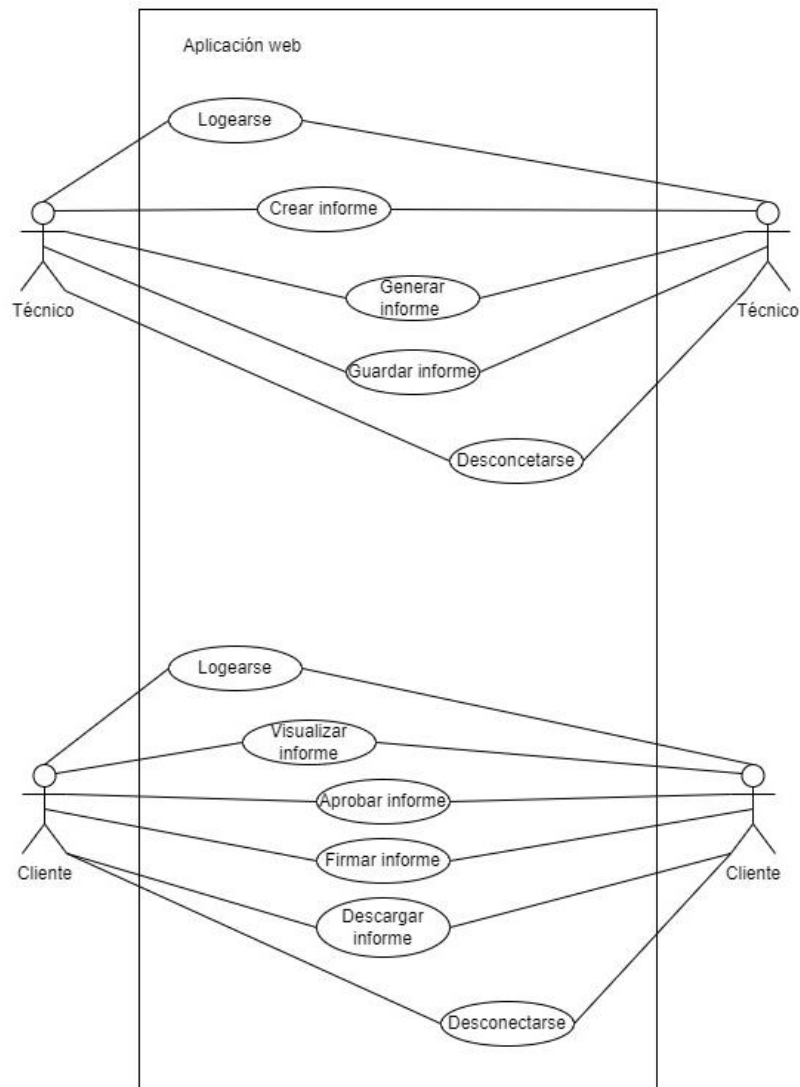


Ilustración 2 **Casos de uso alto nivel**

Descripción.

Primero el técnico inicia sesión, luego tiene la pestaña de crear informe que se divide en generar informe y guardar informe y por último el técnico se desconecta. El cliente inicia sesión, luego tiene la pestaña de visualizar informe que este se divide en aprobar informe, firmar informe y descargar informe, luego que el cliente haya hecho todos sus procesos se desconecta.

Diagrama de casos de uso (extends e include)

La extensión y la inclusión son importantes para definir un sistema ya que a medida que más se profundice estos casos de uso se extenderán y crecerán. Aquí es donde entran las inclusiones y extensiones que según Schmuller (2000) la inclusión se encarga de eliminar la duplicación de pasos en un caso al otro, por lo que permite crear un nuevo caso de uso adicional. Y la extensión se encarga de extender un caso de uso siendo una instancia del caso de uso, por ejemplo, un caso de uso de comprar comida y su extensión sería comprar según lo que haya en la alacena. Con esto podemos decir que un caso de uso se puede agrandar cada vez más y por esto es importante tener un orden para no tener equivocaciones y confusiones. (p.91)

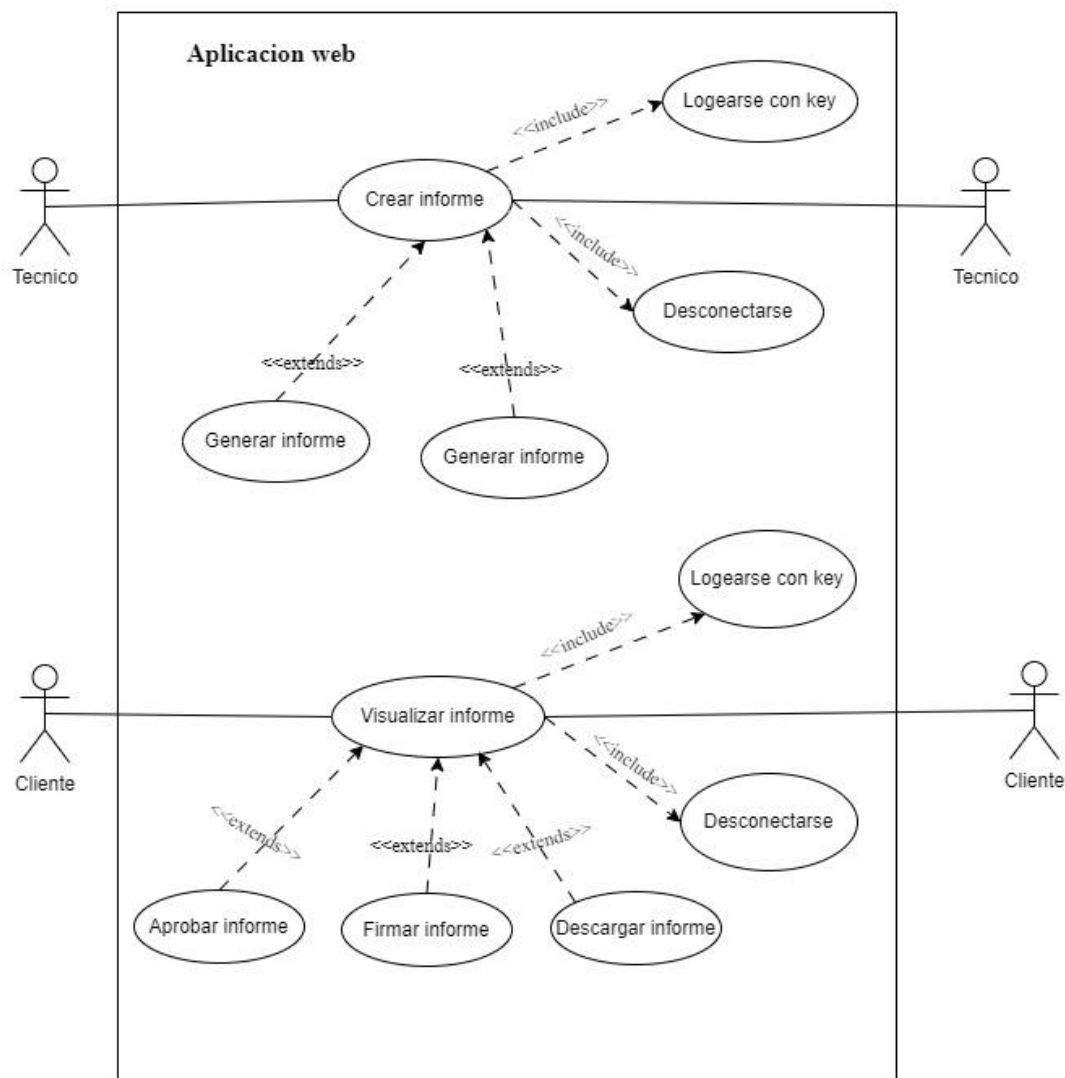


Ilustración 3 Casos de usos extends e include

Descripción.

En este diagrama de caso de uso se evidencia los extends, ampliaciones con funcionalidades adicionales, y los include, descomposición de los casos de uso complejos en casos de uso más simples.

Primero el técnico inicia sesión, <<include>> logearse con key, luego tiene la pestaña de crear informe que se divide en generar informe (<<extends>>) y guardar informe (<<extends>>) y por último el técnico se desconecta (<<include>>).

Por último, el cliente inicia sesión (<<include>> Logearse con key), después tiene la pestaña de visualizar informe que este se divide en aprobar informe (<<extends>>), firmar informe (<<extends>>) y descargar informe (<<extends>>), por último, que el cliente haya hecho todos sus procesos se desconecta (<<include>>).

Diagramas de Clases

El diagrama de clases es fundamental en todo proyecto ya que es la base para desarrollar cualquier proyecto, normalmente este se encuentra orientado a objetos y según Fontela (2010):

"La clase es una construcción de todos los lenguajes orientados a objetos. Esto hace que el diagrama de clases sea el diagrama estructural más importante a la hora de modelar diseño detallado y programación." (p.108)

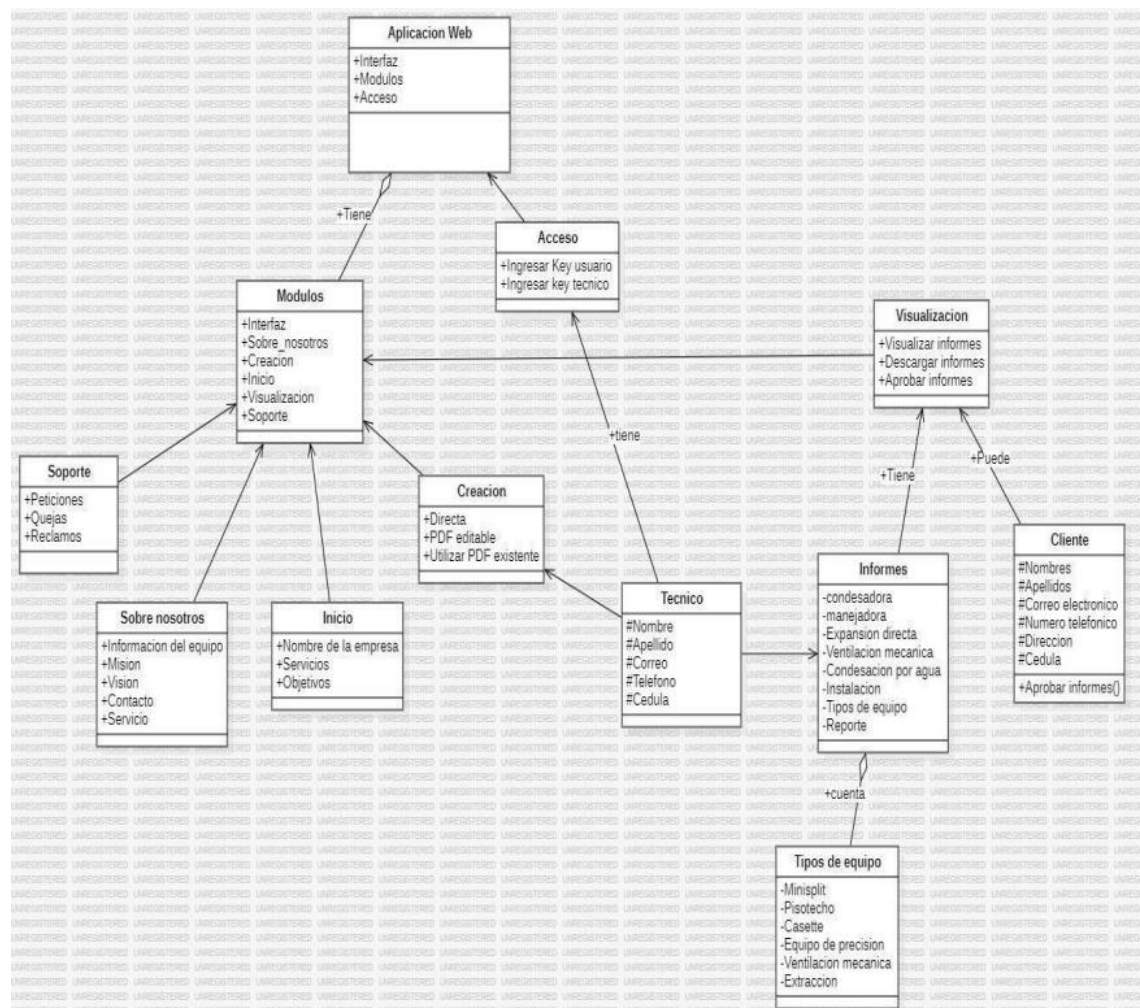


Ilustración 4 Diagramas de clases

Descripción

En este diagrama se observan las clases y sus interacciones. Cada clase es un rectángulo el cual en la parte de arriba muestra el nombre de la clase y en la parte inferior los atributos y métodos de cada clase. Por medio de los diferentes tipos de flechas se evidencia el tipo de relación que tienen las clases.

Por ejemplo, las clases Soporte, Inicio, Creación, visualización y Sobre nosotros se asocian con la clase Módulos esto debido a que estas son atributos de Modulo, pero al tener sus propios atributos se vuelven clases.

Como se muestra en el diagrama la clase genérica es Aplicación web ya que es la estancia principal.

Diagrama de Estados

Segun Schmuller (2000):

“El diagrama de estados del UML proporciona una gran variedad de símbolos y abarca varias ideas (todas para modelar los cambios por los que pasa un objeto). Este tipo de diagrama tiene el potencial de convertirse en algo complejo con pasmosa rapidez...

Es necesario contar con diagramas de estados dado que permiten a los analistas, diseñadores y desarrolladores comprender el comportamiento de los objetos de un sistema” (p.91)

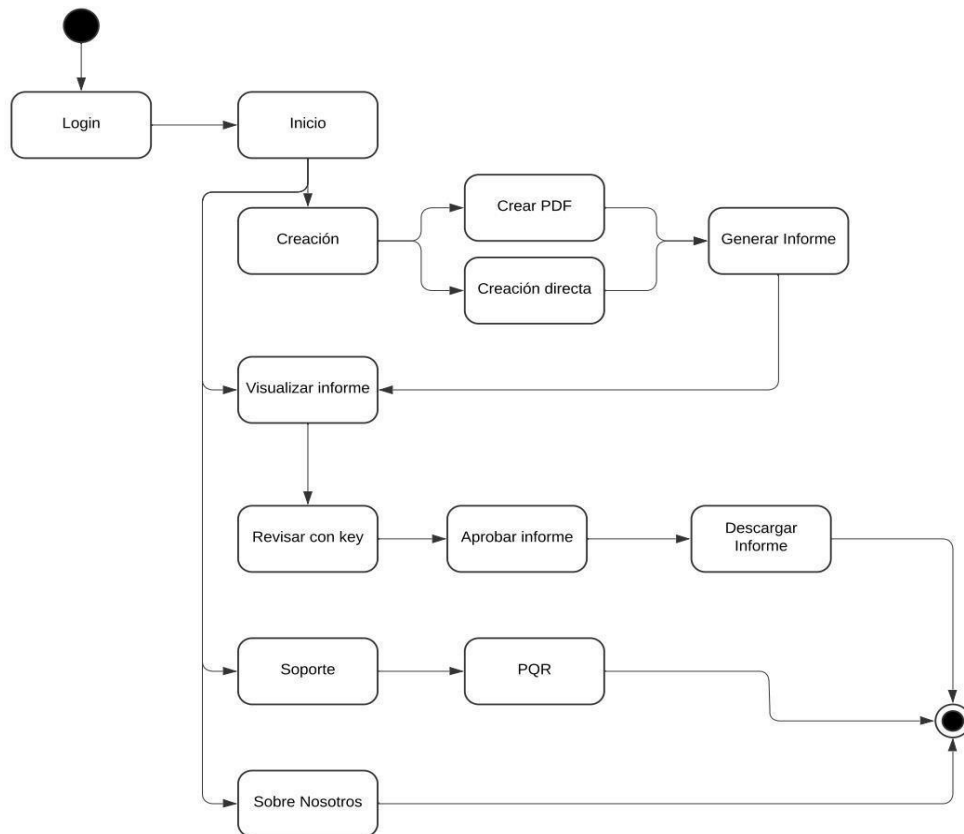


Ilustración 5 Diagrama de Estados

Descripción

En este diagrama se muestran los siguientes estados en este caso al estar en el estado de login se inicia sesión, da la opción de dirigirse a un nuevo estado como lo es el de inicio, este estado nos dirige a creación y a su vez, este nos dirige a dos opciones de estados, crear PDF o creación directa.

Los estados siempre están ligados a otros como se ve en el diagrama, el estado generar informe depende de los estados creación esto se debe, a que el diagrama de estados es como un paso a paso, no se puede realizar cierta acción si no se han llevado a cabo las anteriores, pero, esto no quiere decir que hay que repetir estas acciones cada vez que se quiera visualizar el informe, por ejemplo, si las etapas de creación y Generar informe ya se han realizado, no hay la necesidad de volver a pasar por estas para eso está la etapa de Visualizar informe.

Modelo entidad relación

Este Modelo es importante para entender las relaciones entre las diferentes entidades, además este junto con el diagrama de clases son importantes tener en cuenta, según Valderrey (2011):

“Un modelo entidad-relación (a veces denominado por sus siglas, E-R “Entity Relationship”, o, “DER”, Diagrama de Entidad Relación) es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, así como sus interrelaciones y propiedades.” (p.85)

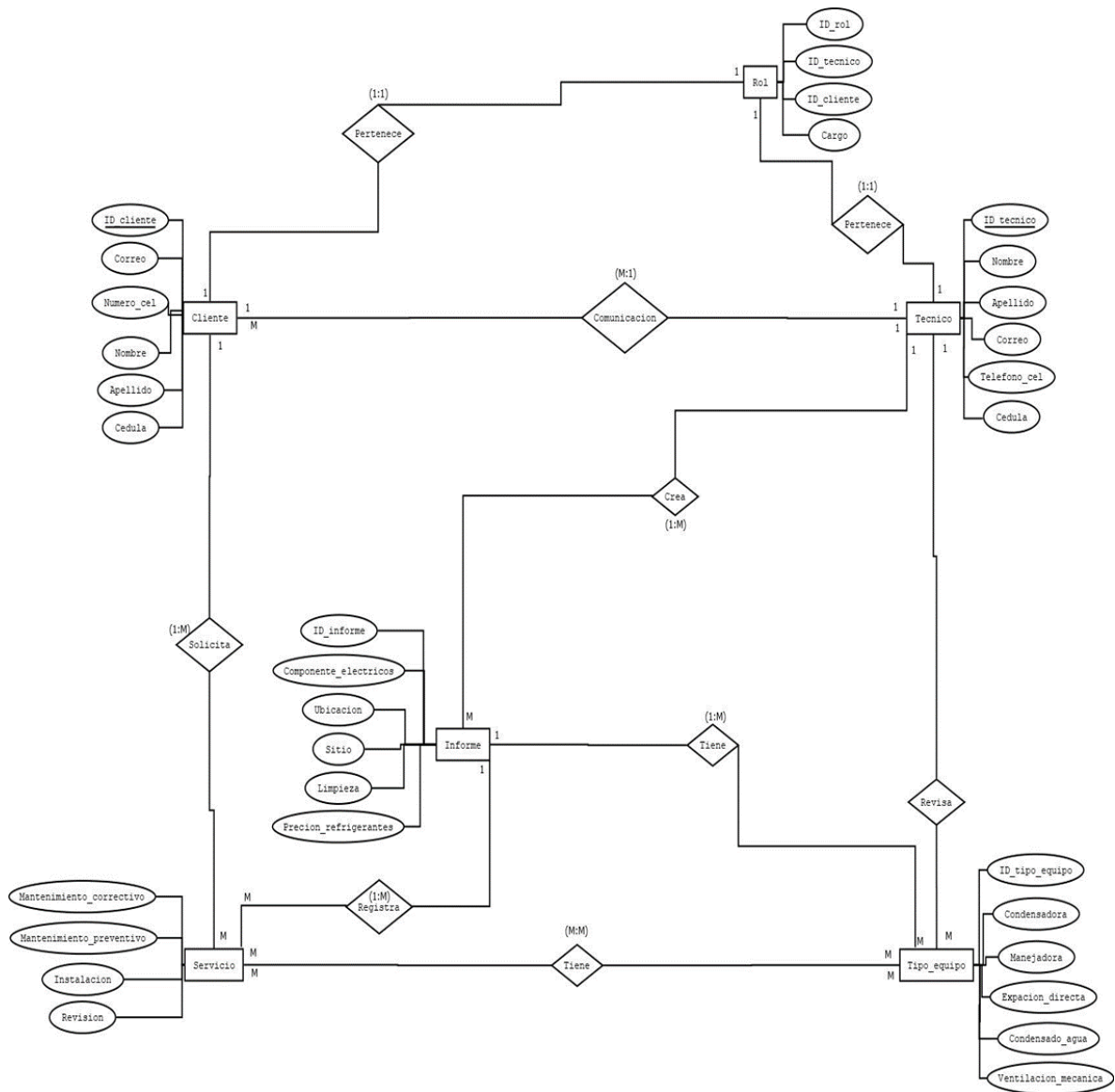


Ilustración 6 **Modelo entidad relación**

Descripción

En este diagrama se muestran las entidades y sus atributos juntos con las relaciones entre entidades, dependiendo de la relación que tengan las entidades se define las cardinalidades ya se han se muchos a uno (M: 1) como se muestra en la relación entre Cliente y Técnico o, de muchos a muchos (M: M) como en el caso de Servicio y Tipo equipo.

El tipo de cardinalidad o relación depende mucho de las entidades y como se manejen, por ejemplo:

Un cliente se contacta con un técnico mientras que aún técnico se contacta con muchos clientes de ahí esa relación de Muchos a uno (M: 1).

Diagrama Relacional

Para entender mejor de que trata este diagrama relacional vamos a ver lo que dijo Capacho y Nieto (2017):

“Un Modelo Relacional (MR). El MR representa la estructura de la base de datos como una colección de relaciones. Las relaciones en la base de datos contienen del M E-R tanto las entidades con sus atributos como las relaciones representadas a través de tablas. Luego, el contenido de una tabla relacionales una estructura de n filas, llamadas tuplas, por m columnas. Las columnas idénticas los atributos de las entidades, y las tuplas o filas almacenan los datos concretos de la tabla relacional.” (p.80)

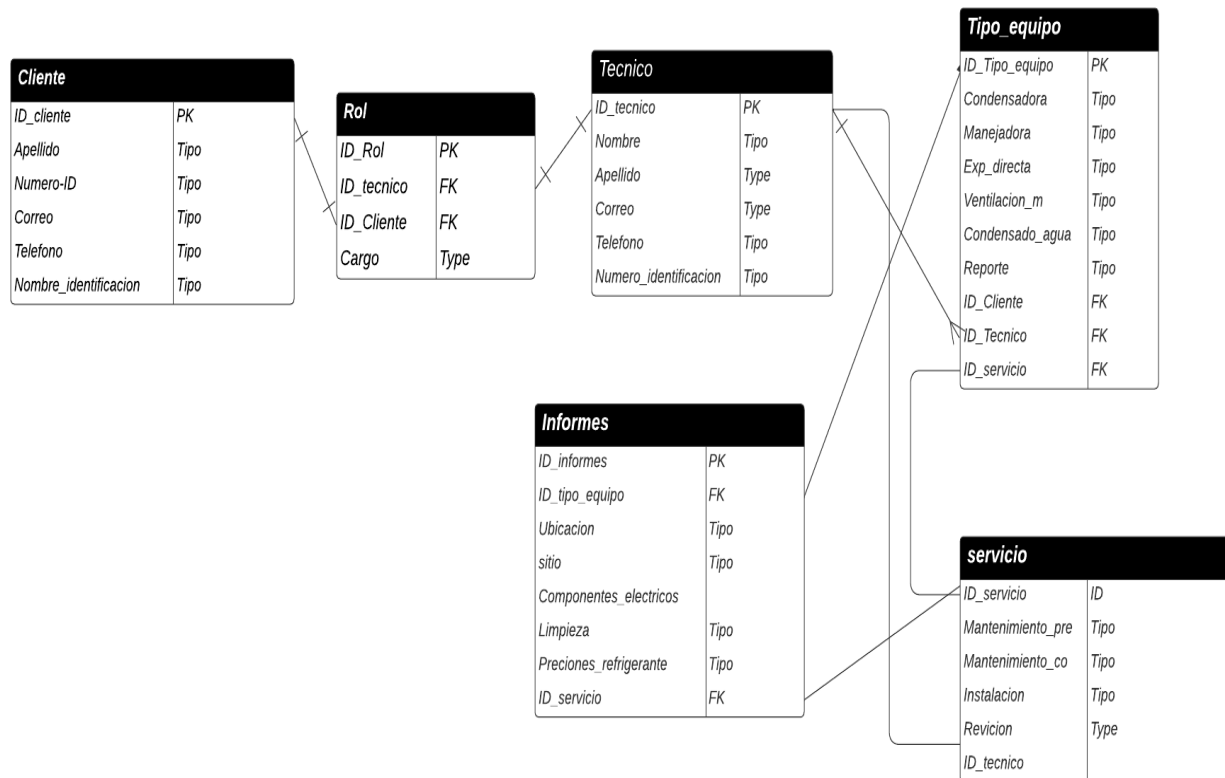


Ilustración 7 Diagrama Relacional

Descripción

En este diagrama se evidencian las clases, atributos, relaciones, tipos de relaciones y el tipo de sus llaves. Las llaves son el medio por el que se conectan las entidades, existen dos tipos de llaves, las llaves primarias y las secundarias. Las primarias son aquellas que permiten la relación de la tabla que la posee con las tablas que necesitan la información de esta, por ejemplo, en la tabla cliente existe la clave primaria ID cliente esta clave primaria transporta todos los atributos de su tabla lo cual es esencial ya que, no hay necesidad de volver a escribir toda esa información en las otras tablas que la necesiten.

Conclusión

Para concluir, el desarrollo web requiere un conocimiento de diversas tecnologías y conceptos. A lo largo de este informe, hemos visto los elementos importantes que intervienen en la construcción de un aplicativo web, desde el diseño hasta la implementación.

Hemos visto cómo el diseño web proporciona un marco de trabajo para organizar proyectos, y cómo las tecnologías front-end (HTML, CSS y JavaScript) son esenciales para crear interfaces de usuario interactivas y funcionales. Además, hemos analizado la importancia del back-end, las bases de datos y los lenguajes de programación para gestionar la lógica del servidor y los datos.

Finalmente, hemos visto conceptos de como la arquitectura web, los patrones de diseño y los modelos de datos. Estos conceptos son importantes para construir aplicativos webs de la mejor manera.

Referencias Bibliográficas

- Escarcena, M. (2020). *Programación páginas web JavaScript y PHP*. Ra-Ma Editorial.
<https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=17194>
- Fernández Casado, P. E. (2020). *Diseño y construcción de páginas web*. Ediciones de la U & Ra-Ma Editorial. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=17225>
- Marqués, M. (2009). Bases de datos: (ed.). Castelló de la Plana, Spain: D - Universitat Jaume I. Servei de Comunicació i Publicacions.
<https://elibro.net/es/ereader/senavirtual/51645?page=10>
- Rinaudo, L, & Pantaleo, L. (2018). Ingeniería de Software. (1ª Ed.). Alfaomega. <https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/ingenieria-de-software?location=8>
- López, J, & Gutiérrez, A. (2017). Desarrollo y programación en entornos web. (1ª Ed.). Alfaomega. <https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/desarrollo-y-programacion-en-entornos-web?location=7>
- Katz, M. (2013). Redes y seguridad. (1ª Ed.). Alfaomega. <https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/redes-y-seguridad?location=10>
- Gutiérrez, P. (2020). *El libro blanco del hacker*. Ediciones de la U. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=10280>
- Sznajdleder, P. (2017). Programación Orientada a Objetos y Estructura de Datos a Fondo. (1ª Ed.). Alfaomega. <https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/programacion-orientada-a-objetos-y-estructura-de-datos-a-fondo?location=8>
- Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB: (ed.). Barcelona, Spain: Editorial UOC.
<https://elibronet.bdigital.sena.edu.co/es/ereader/senavirtual/58524?page=25>
- López, M. (2016). programación web en el entorno del cliente. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=7955>

- Flórez, H & Hernández, J. (2021). Aplicaciones web con PHP. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=14060>
- Capacho, J & Nieto, W (2017). *Diseño de base de datos*. Editorial Universidad del Norte. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=9740>
- Hernández M. y Baquero, L. (2021). *Estructuras de datos: fundamentación práctica*. Ediciones de la U. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=16014>
- Reinosa, M. et al. (2010). *Bases de datos*. (1ª Ed.). Alfaomega. <https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/bases-de-datos?location=52>
- Sommerville, I. (2011). *Ingeniería de software*. Pearson Educación. <https://www-ebooks724-com.bdigital.sena.edu.co/?il=3313>
- Schmuller, J. (2000) *Aprendiendo UML en 24 Horas*. Prentice Hall. [aprendiendo-uml-en-24-horas.pdf - Google Drive](https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/uml-modelado-de-software-para-profesionales?location=27)
- Fontela, C. (2010). *UML Modelado de Software para Profesionales*. (1ª Ed.). Alfaomega. <https://www-alphaeditorialcloud-com.bdigital.sena.edu.co/reader/uml-modelado-de-software-para-profesionales?location=27>
- Valderrey Sanz, P. (2011). *Gestión de bases de datos*. Starbook Editorial. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=17222>
- Capacho, J. & Nieto, W. (2017). *Diseño de base de datos*. Editorial Universidad del Norte. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=9740>