



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de
Ingeniería campus Zacatecas

Área de ubicación para el desarrollo del trabajo

Ingeniería en Sistemas Computacionales

Línea de investigación

Desarrollo de Aplicaciones

Título del proyecto de Trabajo Terminal

GPS en 2D y 3D de la UPIIZ

Presenta(n):

Miguel Ángel González Gallegos

Ana Paulina López Cazares

Director:

ISC. Efraín Arredondo Morales



Zacatecas, Zacatecas a 19 de Junio de 2023

Índice de contenido

Propósito.....	2
Arquitectura del sistema	2
Diseño de la base de datos.....	5
Couchbase Lite	6
UnQLite	8
LevelDB.....	9
Base de datos oracle Berkeley	10
RealTime database	10
Diagramas UML	14
Diagrama de casos de uso	14
Diagrama de actividades	30
Diagrama de clases	34
Diagrama de despliegue.....	54
Diseño de prototipos	59
Referencias	74

Índice de tablas

Tabla 1:

Objetos para la elaboración de diagramas de uso	15
Tabla de descripción de caso de uso del diagrama general de visualizar simulación (CU 01)	19
Tabla de descripción de caso de uso del diagrama general de velocidad de cámara (CU 02)	19
Tabla de descripción de caso de uso del diagrama general de descargar mediante QR (CU 03).....	20
Tabla de descripción de caso de uso del diagrama general de intercambio de mapas (CU 04)	21
Tabla de descripción de caso de uso del diagrama general de visualizar menú (CU 05).....	21

Tabla de descripción de caso de uso del diagrama general de ver mapa2D (CU 06)	22
Tabla de descripción de caso de uso del diagrama general de escoger puntos (CU 07)	22
Tabla de descripción de caso de uso del diagrama general de buscar lugar (CU 08)	23
Tabla de descripción de caso de uso del diagrama general de calcular recorrido (CU 09) ..	24
Tabla de descripción de caso de uso del diagrama general de calcular ubicación (CU 10) ..	24
Tabla de descripción de caso de uso del diagrama general de calcular tiempo (CU 11)	25
Tabla de descripción de caso de uso del diagrama general de visualizar tiempo (CU 12) ..	26
Tabla de descripción de caso de uso del diagrama general de mostrar ruta (CU 13)	26
Tabla de descripción de caso de uso del diagrama general de notificar cambios (CU 14) ..	27
Tabla de descripción de caso de uso del diagrama general de iniciar sesión (CU 15)	27
Tabla de descripción de caso de uso del diagrama general de visualizar mapa edición (CU 16)	28
Tabla de descripción de caso de uso del diagrama general de editar información (CU 17) ..	29
Tabla de descripción de caso de uso del diagrama general de actualizar base de datos (CU 18)	29
Descripción de elementos de un diagrama de actividades	31
Descripción de las divisiones en un diagrama de clases	35
Tipos de conexiones en un diagrama de clases	36
Simbología para los diagramas de despliegue	54
Identificador de los diagramas de diseño	68

Índice de figuras

Figura 1: Arquitectura	5
Figura 2: Diseño de base de datos	14
Figura 3: Diagrama de casos de uso general	18
Figura 4: Diagrama de actividades del usuario (DA 01)	33
Figura 5: Diagrama de actividades del usuario administrador (DA 02)	34
Figura 6: Diagrama de clases competo	37
Figura 7: Diagrama de clase BtnCambioEscena(DC 01)	38
Figura 8: Diagrama de clase DatoEntreEscena (DC 02)	39
Figura 9: Diagrama de clases de la escena menú inicio	40
Figura 10: Diagrama de clase GameManagerMenuInicio (DC 03)	41

Figura 11: Diagrama de clase GameManagerMapa3D (DC 04)	42
Figura 12: Diagrama de clase PlayerNavMesh(DC 05)	43
Figura 13: Diagrama de clases de la escena del mapa 3D	44
Figura 14: Diagrama de clases de la escena mapa 2D.....	45
Figura 15: Diagrama de clase PermisoUbicacion(DC 06)	46
Figura 16: Diagrama de clase GPSDatos(DC 07)	46
Figura 17: Diagrama de clase GameManagerMapa2D(DC 08)	47
Figura 18: Diagrama de clase DragenDrop (DC 09).....	48
Figura 19: Diagrama de clases menú inicio y comportamiento de los botos botones.....	49
Figura 20: Diagrama de clase GameManagerMenuBusqueda (DC 10)	50
Figura 21: Diagrama de clase PruebaClaseDato (DC 11)	50
Figura 22: Diagrama de clase IniciarBtn (DC 12).....	51
Figura 23: Diagrama de clase BtnClick (DC 13)	51
Figura 24: Diagrama de clase PrubaLecturaDatos (DC 14).....	52
Figura 25: Diagrama de clase BarraBusqueda (DC 15)	52
Figura 26: Diagrama de clase PruevaLista (DC 16).....	53
Figura 27: Diagrama de clase ExportarBD(DC 17)	53
Figura 28: Notación de un nodo con subnodos	55
Figura 29: Diagrama de despliegue	56
Figura 30: Diagrama de despliegue del Usuario (DES 03)	57
Figura 31: Diagrama de despliegue del Administrador (DES 01).....	58
Figura 32: Diagrama de despliegue de la Base de Datos (DES 02)	59
Figura 33: Diseño del apartado móvil	61
Figura 34: Diseño del apartado de escritorio.....	63
Figura 35: Interacción usuario.....	65
Figura 36: Interacción usuario administrador.....	67



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

CONTROL DE VERSIONES					
Autor(es))	Fecha de modificación	Versión	Descripción del cambio	Revisó	Estado
APLC MAGG	24/10/22	1.0	Creación del Documento	EAM	Pendiente
APLC MAGG	02/10/22	1.0	Terminando el apartado de diagramas UML despliegue	EAM	Revisado con comentarios
APLC MAGG	03/11/22	1.0	Agregando información faltante	EAM	Revisado con comentarios
APLC MAGG	07/11/22	1.0	Corrección en imágenes	EAM	Revisado con comentarios
APLC MAGG	10/11/22	1.0	Modificar algunas palabras en la descripción, agregar más referencia en BD	EAM	Revisión con correcciones
APLC MAGG	18/11/22	1.0	Corrigiendo los diagramas de actividades	EAM	Validado
APLC MAGG	02/02/23	2.0	Corrección del diagrama de Base de Datos	EAM	Validado
MAGG, APLC	10/02/23	2.0	Corrección de las tablas del diagrama de casos de uso	EAM	pendiente
MAGG, APLC	18/05/2023	3.0	Corrección de los diagramas	EAM	pendiente
MAGG, APLC	25/05/2023	3.0	Corrección de los diagramas	EAM	Validado



Documento de diseño

Propósito

El presente documento se realiza con el fin de presentar la arquitectura del sistema, la base de datos que se estará manejando, los diagramas UML correspondientes y los prototipos de diseño, que a la vez son las 4 secciones en las que se divide el documento, con el objetivo de centrar en un solo documento toda la información recabada de la fase de “Diseño” del modelo de cascada que sirve para dar continuación a la etapa de “Implementación”.

Arquitectura del sistema

Una arquitectura de software son lineamientos o patrones que ayudan a la construcción de un programa, permitiendo tener una guía para cumplir con los requerimientos de la aplicación. Ayuda a trazar un camino sobre lo que se va a hacer y cómo se realizara. Se pueden definir partes como servidores, tecnologías, bases de datos, etc. (Cervantes, 2010)

Las arquitecturas cuentan con patrones que son formas de captura estructural de diseño, siendo un paquete de decisiones de diseño. El desarrollo de la arquitectura puede ser visto como un proceso de selección, adaptación y combinación de patrones. (Novoseltseva, 2020)

Mark Richards escribió el libro “software architecture patterns” donde comenta que existen 5 patrones principales de arquitectura de software: microkernel, microservicios, arquitectura en capas, basada en eventos y basada en el espacio. (Novoseltseva, 2020)

La arquitectura basada en capas es una de las más comunes, estructura los atributos del sistema mediante capas separando responsabilidades y administra dependencias. Cada capa tiene una responsabilidad específica, por ejemplo, una capa de presentación se encargaría de manejar la interfaz de usuario y la de lógica. (Novoseltseva, 2020) (Microsoft, 2022)



Documento de diseño

Entre las ventajas que ofrece esta arquitectura es la facilidad para asignar roles separados, fácil de actualizar y mejorar las capas por separado. Es fácil de implementar, ya que no es excesivamente complejo, por lo que suele ser una elección para la mayoría de los desarrolladores de aplicaciones. (Novoseltseva, 2020)

Este tipo de arquitectura puede desarrollarse en 3 capas, 4 capas o n-capas. En una arquitectura de 3 capas se organiza los elementos: nivel de presentación, de aplicación y de datos. (IBM Cloud Education, 2020)

- **Nivel de presentación:** Es la interfaz de usuario y la capa de comunicación de la aplicación, donde el usuario interactúa con la aplicación (IBM Cloud Education, 2020)
- **Nivel de aplicación:** Se procesa la información recopilada en el nivel de presentación. Aquí normalmente se desarrolla utilizando un programa de programación y se comunica con el nivel de datos utilizado (IBM Cloud Education, 2020)
- **Nivel de datos:** Nivel de acceso a datos o back-end, es donde se almacena y gestiona la información procesada de la aplicación. (IBM Cloud Education, 2020)

Uno de los beneficios de la arquitectura de 3 capas es que cada nivel se ejecuta en su propia infraestructura, cada nivel puede ser desarrollado simultáneamente por un equipo de desarrollo independiente y se puede actualizar o escalar. (Novoseltseva, 2020)

Por la simplicidad del diseño se eligió la arquitectura basada en 3 niveles, donde se representan el nivel de presentación, la de aplicación y la de datos.



Documento de diseño

En el primer nivel se encuentra la presentación, son los dos modos en que el usuario podrá interactuar con el producto terminado, que es escritorio y Móvil.

En el segundo nivel se encuentra la sección de aplicación donde se ponen los componentes que conformara el producto. Se hace una separación de componente respecto a lo que tendrá el apartado 2D y el apartado 3D. El producto necesitara el NavMesh, que es la API de inteligencia artificial que brinda Unity, permite realizar recorridos y los cálculos necesarios para mostrar el tiempo y las rutas (Unity, s.f.); el trazo de ruta para indicar el camino hacia el destino, la búsqueda hace referencia a buscar el elemento al cual se desea llegar, y puntos es el componente de la modificación de los puntos para irse actualizando en caso de que se presente un cambio. Todo esto depende de la conexión hacia la base de datos, lo que lleva al último nivel de datos.

En el último nivel de la arquitectura se encuentra la Real Time Database, una base de datos compatible con Unity que ofrece una conexión sencilla. Una de sus características destacadas es la capacidad de notificar a los dispositivos cuando se produce un cambio en la base de datos. (Firebase, s.f.).

A continuación, se presenta una figura que ilustra la arquitectura implementada.

Se realizaron algunos ajustes en la arquitectura del sistema para el apartado del administrador. Se pasó de una plataforma web a una aplicación de escritorio, mientras que el resto de la estructura se mantuvo sin cambios significativos.



Documento de diseño

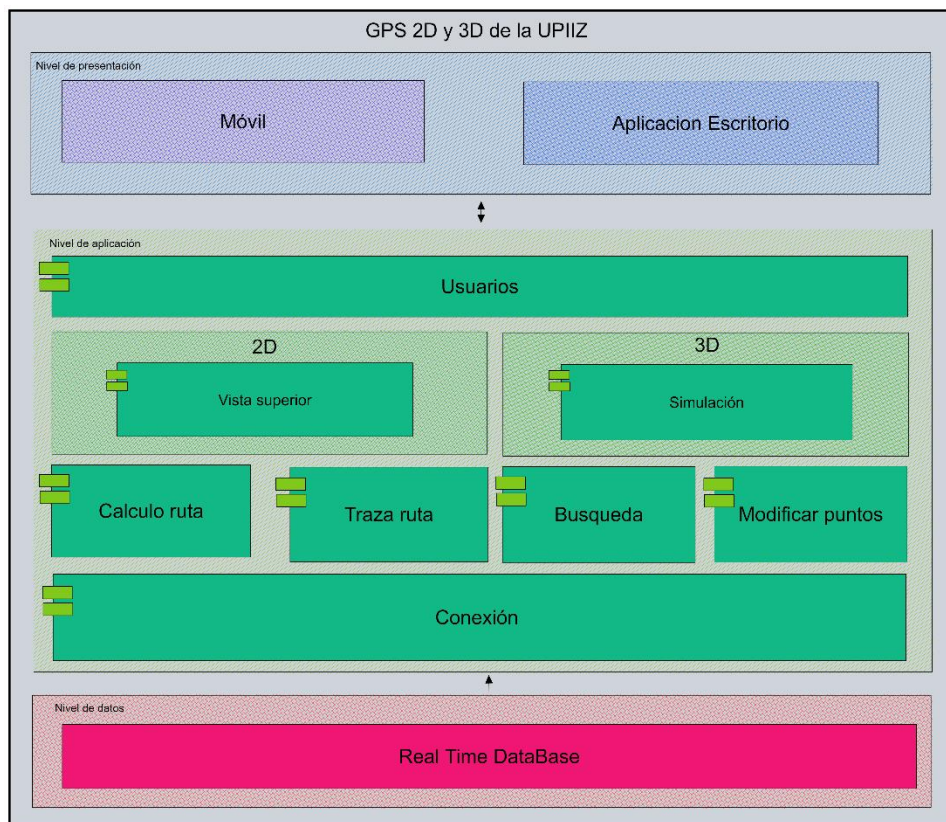


Figura 1: Arquitectura

Fuente: Elaboración Propia

Diseño de la base de datos.

Una base de datos es la recopilación de información que puede ser estructurado, aunque en la actualidad ya se están implementando bases de datos conocidas como NoSQL. Una estructura de una base de datos SQL se suele presentar por tener columnas y filas, siendo tablas que se pueden relacionar entre sí, las bases NoSQL pueden representarse de esta misma forma, pero se trabajan como si fueran archivos. (ACENS, 2014; OCI, 2014)



Documento de diseño

Las bases NoSQL aparecen con la llegada de la web 2.0, y llegaron para solucionar los problemas de accesibilidad, por lo tanto, las bases NoSQL son estructuras que permiten almacenar información permitiendo una escalabilidad y rendimiento. (ACENS, 2014)

Una característica de este tipo de base de datos es que no cumplen con el esquema de entidad-relación, no usan estructuras de tablas como se había comentado anteriormente, hacen uso de formatos como clave-valor, mapeo de columnas o grafos. Entre las ventajas que ofrecen las bases de datos NoSQL se encuentran:

- Se ejecuta en máquinas con pocos recursos (ACENS, 2014)
- Escalabilidad horizontal (ACENS, 2014)
- Puede manejar gran cantidad de datos (ACENS, 2014)
- No genera cuellos de botella (ACENS, 2014)

Algunos ejemplos de bases de datos NoSQL son:

Couchbase Lite

Base de datos NoSQL integrada para dispositivos móviles, de escritorio y personalizados integrados

- Primero sin conexión

Couchbase Lite garantiza que las aplicaciones estén siempre disponibles independientemente de las conexiones de red. Los datos se almacenan y procesan sin conexión, y se sincronizan



Documento de diseño

con los centros de datos back-end o Capella App Services cuando la conectividad está disponible. (Couchbase, s.f.)

- Amigable para el desarrollador

Desarrolle más rápido con soporte completo de SQL, API fáciles de usar para CRUD de base de datos, búsqueda de texto completo (FTS) y operaciones de consulta, y soporte integral de plataforma e idioma. El formato JSON sin esquema simplifica las actualizaciones de aplicaciones sin necesidad de una tediosa migración de datos. (Couchbase, s.f.)

- Peso ligero

Couchbase Lite tiene una huella pequeña, que va desde 1MB – 5MB dependiendo de la plataforma. Es adecuado para dispositivos móviles e integrados con recursos limitados. (Couchbase, s.f.)

- Soporte multiplataforma

Soporte para plataformas móviles como iOS, Android y Windows. Desarrolle aplicaciones de escritorio y servidor web con nuestras bibliotecas .NET y Java. El soporte de C-API permite a los desarrolladores crear aplicaciones en C que se ejecutan en casi cualquier plataforma integrada. (Couchbase, s.f.)

Según Couchbase (s.f) algunos SDKs con los que cuenta son:

- Ruby
- Scala
- Java



Documento de diseño

- .NET
- PHP
- C
- Go

Costo

\$6.72 DLS POR DIA (Couchbase, s.f.)

UnQLite

UnQLite es una biblioteca de software en proceso que implementa un motor de base de datos NoSQL transaccional autónomo, sin servidor, de configuración cero. Cuenta con un almacenamiento de documentos similar a MongoDB, Redis, CouchDB, etc., así como un almacén de clave / valor estándar similar a BerkeleyDB, LevelDB, entre otros. Algunas características que menciona Symisc UnQLite (2018) son:

- Motor de base de datos NoSQL sin servidor.
- Base de datos transaccional (ACID).
- Configuración cero.
- Archivo de base de datos único, no utiliza archivos temporales.
- Formato de archivo multiplataforma.
- UnQLite es una biblioteca C autónoma sin dependencia.
- Almacén de clave/valor estándar.
- Base de datos del almacén de documentos (JSON) a través de Jx9.
- Soporta cursores para registros lineales transversales.
- Soporte para bases de datos en disco y en memoria.



Documento de diseño

- API simple, limpia y fácil de usar.
- Admite bases de datos de tamaño terabyte.
- Producto con licencia BSD.
- API EN C/C++
- Amalgamación: Todo el código fuente de C para UnQLite y Jx9 se combina en un solo archivo fuente.
- Soporte en línea de alta disponibilidad.

LevelDB

Es un participante relativamente nuevo en la creciente lista de bibliotecas de bases de datos de claves / valores por lo que sigue en desarrollo. Es un proyecto de código abierto patrocinado por Google que se ha incorporado a una aplicación Erlang y se ha integrado en Riak para el almacenamiento de información clave/valor en disco. Su acceso de lectura puede ser lento cuando hay muchos niveles para buscar. (Riak, s.f.)

El conjunto de tablas ordenadas se organiza en una secuencia de niveles. Cada nivel almacena aproximadamente diez veces más datos que el nivel anterior. La tabla ordenada generada a partir de un flush se coloca en un nivel joven especial (también llamado nivel 0). Cuando el número de archivos jóvenes supera un cierto umbral (actualmente cuatro), todos los archivos jóvenes se fusionan con todos los archivos de nivel 1 superpuestos para producir una secuencia de nuevos archivos de nivel 1 (se crea un nuevo archivo de nivel 1 por cada 2 MB de datos). (Riak, s.f.)



Documento de diseño

Las compactaciones para un nivel particular giran a través del espacio de la clave. Con más detalle, para cada nivel L, LevelDB recuerda la clave final de la última compactación en el nivel L. La siguiente compactación para el nivel L elegirá el primer archivo que comienza después de esta clave (envolviéndose hasta el principio del espacio de claves si no hay tal archivo). (Riak, s.f.)

Base de datos oracle Berkeley

Oracle Berkeley DB 12c proporciona los servicios de almacenamiento de información fundamentales para su aplicación, sin importar cuán exigentes y únicos parezcan ser sus requerimientos. De acuerdo a Oracle (s.f.), con el uso de Berkeley DB se puede:

- Reduzca el tiempo de comercialización
- Reduzca los costos de desarrollo
- Simplifique el almacenamiento de datos en dispositivos móviles
- Reduzca el costo de implementación
- Elimine las costosas reescrituras a medida que su sistema crece
- Elimine los costosos gastos administrativos
- Elimine la pérdida y corrupción de datos
- Proporcionar servicios de alta disponibilidad a escala de Internet

RealTime database

Cloud Firestore es una base de datos NoSQL alojada en la nube a la que pueden acceder tus apps para Apple, Android, Web y escritorio directamente desde los SDK nativos.



Documento de diseño

Cloud Firestore también está disponible en los SDK nativos de Node.js, Java, Python, Unity, C++ y Go, además de las API de REST y RPC. (Firebase, s.f.)

- **Sincronización en tiempo real para datos JSON**

Firebase Realtime Database es una base de datos NoSQL alojada en la nube que te permite almacenar y sincronizar datos entre tus usuarios en tiempo real. (Firebase, s.f.)

- **Colabora entre dispositivos con facilidad**

La sincronización en tiempo real permite que los usuarios accedan a sus datos desde cualquier dispositivo, móvil o de escritorio, con facilidad, y los ayuda a trabajar en conjunto. (Firebase, s.f.)

- **Crea aplicaciones sin servidores**

Realtime Database se incluye en los SDK de escritorio y para dispositivos móviles, de manera que puedas crear apps sin la necesidad de usar servidores. También puedes ejecutar un código de backend que responda a los eventos que activa la base de datos a través de Cloud Functions para Firebase. (Firebase, s.f.)

- **Optimizada para el uso sin conexión**

Cuando los usuarios se desconectan, los SDK de Realtime Database usan la caché local del dispositivo para publicar y almacenar cambios. Cuando el dispositivo se conecta, los datos locales se sincronizan de manera automática. (Firebase, s.f.)



Documento de diseño

- **Seguridad sólida basada en usuarios**

Realtime Database se integra en Firebase Authentication para brindar una autenticación intuitiva y sencilla a los desarrolladores. Puedes usar nuestro modelo de seguridad declarativa para permitir el acceso según la identidad de los usuarios o patrones que coincidan con tus datos. (Firebase, s.f.)

- **Usuarios activos por mes**

Sin costo hasta 50,000 MAU (Métrica de usuario activo) (Firebase, s.f.)

- **Usuarios activos por mes: SAML/OIDC**

Sin costo hasta 50 MAU (Métrica de usuario activo) (Firebase, s.f.)

- **Compatibilidades**

De acuerdo a la documentación de Firebase (s.f.), tiene compatibilidad con:

- C++
- IOS
- ANDROID
- NODE
- JAVA
- UNITY

Por la compatibilidad que tiene con Unity y la característica de sincronización se decidió trabajar con RealTime Database. Aunque es una base de datos NoSQL se estará trabajando con llaves, haciendo que las tablas estén relacionadas. Aunque el esquema del Firebase es no



Documento de diseño

relacional, con el uso de llaves se puede trabajar como relacionales, la única tabla que no estará relacionada sería la tabla de Admin ya que solo se necesita saber las credenciales para poder dar acceso al apartado de administrador.

Se dispone de las siguientes colecciones: Edificios, Docentes, Espacios y Administración, las cuales están interconectadas mediante una clave. La colección de Edificios contiene registros de los edificios pertenecientes a la institución. Cada edificio tiene información sobre su ubicación y puntos dentro de él, donde cada punto puede representar un espacio como un cubículo, laboratorio o área administrativa. Además, en los espacios se puede encontrar una persona.

El diseño de la base de datos se modificó debido a que se trata de una base de datos no relacional, a diferencia del diseño original que era relacional. También se realizaron modificaciones en las colecciones de datos previamente contempladas, con el fin de permitir un orden diferente de los datos.



Documento de diseño

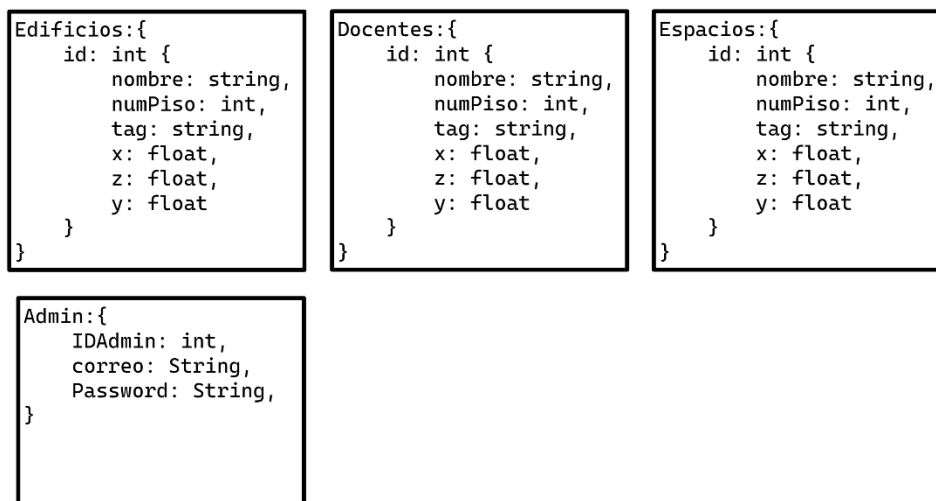


Figura 2: Diseño de base de datos

Fuente: Elaboración Propia

Diagramas UML

Los diagramas UML es un lenguaje común para desarrolladores, permite modelar el sistema de software lo que permite visualizar, especificar, construir y documentar un sistema. (PMOInformatica, 2021)

Algunos ejemplos de diagramas son:

Diagrama de casos de uso

El diagrama de caso de usos es usado para analizar varios sistemas, permitiendo la visualización de los diferentes tipos de roles que tiene el sistema y su interacción con el mismo. (PMOInformatica, 2021)



Documento de diseño

Son modelos que describen como diferentes tipos de usuario interactúan con un sistema informático para resolver un determinado problema o necesidad. Ayuda a describir la interacción entre el usuario y el sistema, como el comportamiento que debe ejecutar el sistema para satisfacerlo. Ayuda a descubrir el comportamiento que tiene el software a eventos realizados por actores externos. (PMOInformatica, 2021)

Los diagramas de caso de uso son usados para especificar los requerimientos funcionales del software, ya que solo describen lo que el software debe hacer y para quien. No entra en detalle sobre como el software será implementado, que componentes tendrá o la interacción entre los componentes. (PMOInformatica, 2021)

Los diagramas de caso de uso cuentan con 4 objetos para su elaboración: actor, caso de uso, sistema y paquete. (Creately, 2022)

Tabla 1

Objetos para la elaboración de diagramas de uso

Objeto	Descripción
Actor	Es cualquier entidad que tenga un papel en el sistema como una persona, organización o un sistema externo.
Caso de uso	Representa una función o una acción dentro del sistema, se dibuja con un ovalo y nombre de la función.
Paquete	Usado más para diagramas complejos, los paquetes se utilizan para agrupar los casos de uso.
Sistema	Usado para definir el alcance de casos de uso y se representa con un rectángulo.



Documento de diseño

Fuente: Creately. (9 de septiembre de 2022). Use Case Diagram Relationships Explained with Examples.

Obtenido de Creately: <https://creately.com/blog/diagrams/use-case-diagram-relationships/>

En un diagrama de casos de usos se reflejan las relaciones, según Creately (2022) existen 5 relaciones en un diagrama de casos de uso, que son:

1. Asociación entre el actor y el caso de uso
 - Un actor debe estar asociado con al menos un caso de uso
 - Un actor puede estar asociado con múltiples casos de uso
 - Se pueden asociar múltiples actores con un solo caso de uso
2. Generalización de un actor: Se refiere a cuando un actor puede heredar el papel de otro actor
3. Ampliar entre dos casos de uso: se refleja como <<extender>> algunas cosas a considerar cuando se usa
 - El caso de uso depende del caso de uso extendido
 - El caso de uso extendido suele ser opcional y se puede activar condicionalmente
 - El caso de uso extendido debe ser significativo por sí mismo
4. Incluir entre dos casos de uso: reutiliza acciones comunes en múltiples casos de uso, ayuda a simplificar comportamientos complejos se escribe como <<include>>
 - El caso de uso está incompleto sin el caso de uso incluido



Documento de diseño

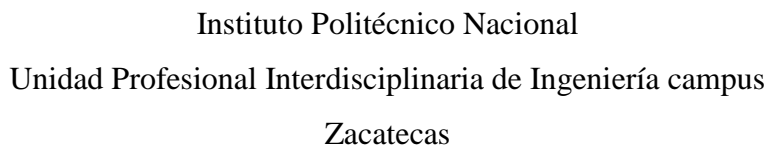
- El caso de uso incluido es obligatorio y no opcional
5. Generalización de un caso de uso: Es cuando hay un comportamiento común entre dos casos de uso y también un comportamiento especializado específico para cada caso de uso.

Para el proyecto se realizó un diagrama de caso general para poner las actividades que los diferentes usuarios pueden realizar. El usuario, desde el menú puede visualizar simulación, ver mapa2D o escoger un punto destino en Visualizar menú. Si decide escoger punto puede buscar el lugar o regresar, el punto realizara una consulta a la base de datos para obtener la ubicación, si se consultan dos puntos se calculará el recorrido y tiempo que se hará entre los puntos elegidos. Si el usuario selecciona luego la simulación o mapa2D visualizara el camino y tiempo.

El usuario administrador deberá iniciar sesión para acceder al menú de edición. En este menú, podrá visualizar un mapa que le permitirá buscar y editar puntos, lo que a su vez actualizará la base de datos.

La base de datos, al ser sincronizada, desempeña un papel importante al notificar los cambios que se realicen.

A continuación, se muestra una imagen que representa el diagrama general con las acciones disponibles para cada usuario, junto con las tablas que indican las funciones correspondientes.

[illegible]

Fuente: Elaboración Propia



Documento de diseño

Tabla 2

Tabla de descripción de caso de uso del diagrama general de visualizar simulación (CU 01)

Caso de uso #1	
Nombre	Visualizar simulación
Actor participante	Usuario
Objetivo	Entrar al mapa en primera persona
Disparador	Botón de simulación
Precondición	Entrar al menú de inicio (cuando se ejecuta la aplicación)
Postcondición	Visualiza la simulación del recorrido dependiendo del punto destino introducido
Flujo básico	Ejecuta la aplicación y entra al menú de inicio de la aplicación. Hacer clic en el botón de simulación una vez introducido un punto destino. Inicia la simulación del recorrido hacia el punto destino introducido.
Flujo alternativo 1	Ejecuta la aplicación y entra al menú de inicio de la aplicación. Hacer clic en el botón de simulación sin haber introducido algún punto destino. Solo se vería lo que ve la cámara sin iniciar un recorrido.
Requerimientos	R02, R04, R12

Fuente: Elaboración Propia

Tabla 3

Tabla de descripción de caso de uso del diagrama general de velocidad de cámara (CU 02)

Caso de uso #2	
Nombre	Velocidad de cámara
Actor participante	Usuario



Documento de diseño

Objetivo	Cambiar la velocidad de recorrido del mapa de simulación
Disparador	Cuando el usuario mueva el scroll de la pantalla
Precondición	Entrar en la simulación con un punto destino definido
Postcondición	La velocidad del recorrido en primera persona
Flujo básico	Introducir un punto destino Seleccionar simulación Cambiar la velocidad con el scroll
Flujo alternativo 1	Si no se selecciona un punto destino queda fijo la cámara en primera persona El scroll inicia con una velocidad por default para iniciar el movimiento
Requerimientos	R14

Fuente: Elaboración Propia

Tabla 4

Tabla de descripción de caso de uso del diagrama general de descargar mediante QR (CU 03)

Caso de uso #3	
Nombre	Descargar mediante QR
Actor participante	Usuario
Objetivo	Escanear el QR para que se redirija a la aplicación para descargar
Disparador	Escanear el QR
Precondición	Tener un celular que pueda leer el QR y el QR a la mano
Postcondición	Visualizar en la PlayStore la aplicación
Flujo básico	Sacar el celular Escanear el QR Instalar
Flujo alternativo 1	Si el QR falla buscar la aplicación por el nombre en la PlayStore
Requerimientos	R13

Fuente: Elaboración Propia



Documento de diseño

Tabla 5

Tabla de descripción de caso de uso del diagrama general de intercambio de mapas (CU 04)

Caso de uso #4	
Nombre	Intercambio de mapas
Actor participante	Usuario
Objetivo	Poder intercambiar de la simulación a la vista superior
Disparador	Botón de cambio
Precondición	Encontrarse en uno de los mapas
Postcondición	Pasar al mapa contrario del que se encontraba
Flujo básico	Seleccionar uno de los mapas Dar clic en el cambio de vista Visualizar el camino en el otro mapa contrario al que se selecciono
Flujo alternativo 1	Se seguirá visualizando el mapa actual Se regresaría al menú inicio y seleccionar el otro mapa
Requerimientos	R06

Fuente: Elaboración Propia

Tabla 6

Tabla de descripción de caso de uso del diagrama general de visualizar menú (CU 05)

Caso de uso #5	
Nombre	Visualizar menú
Actor participante	Usuario
Objetivo	Un menú de inicio cuando el usuario ejecute la aplicación
Disparador	Al abrir la aplicación
Precondición	Abrir la aplicación
Postcondición	Visualizar el menú
Flujo básico	El usuario abrirá la aplicación sea desde el dispositivo móvil o pc



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

Flujo alternativo 1	Haber usado la aplicación y haberla cerrado en alguno de los mapas, y cuando lo abran regresar al menú inicio
Requerimientos	R13

Fuente: Elaboración Propia

Tabla 7

Tabla de descripción de caso de uso del diagrama general de ver mapa2D (CU 06)

Caso de uso #6	
Nombre	Ver mapa2D
Actor participante	Usuario
Objetivo	Visualizar el mapa de la escuela en vista superior
Disparador	Dar clic en el botón de mapa
Precondición	El usuario debe estar en el menú de inicio
Postcondición	Se visualizará el mapa de la institución en una vista superior, indicando el nombre de cada edificio.
Flujo básico	Iniciar la aplicación Ver el menú de inicio Seleccionar punto destino y origen Dar clic en mapa Visualizar la ruta
Flujo alternativo 1	Si no se selecciona el punto destino y origen únicamente se visualizará el mapa sin trazar camino.
Requerimientos	R01, R03

Fuente: Elaboración Propia

Tabla 8

Tabla de descripción de caso de uso del diagrama general de escoger puntos (CU 07)

Caso de uso #7	
Nombre	Escoger puntos



Documento de diseño

Actor participante	Usuario
Objetivo	El usuario pueda introducir el punto inicial y destino
Disparador	Campo de inicio y destino del menú inicio
Precondición	Iniciar la aplicación
Postcondición	Visualizar los campos de inicio y destino de puntos en el menú inicio
Flujo básico	Iniciar la aplicación Visualizar el menú de inicio Seleccionar punto inicio o destino Seleccionar mapa
Flujo alternativo 1	Se puede seleccionar uno de los mapas sin seleccionar algún punto
Requerimientos	R03, R05

Fuente: Elaboración Propia

Tabla 9

Tabla de descripción de caso de uso del diagrama general de buscar lugar (CU 08)

Caso de uso #8	
Nombre	Buscar lugar
Actor participante	Usuario
Objetivo	Buscar el punto al que se dirige o donde inicia.
Disparador	Seleccionar en el menú de inicio el apartado de punto inicio o destino
Precondición	Iniciar la aplicación Visualizar el menú de inicio
Postcondición	Visualizar los elementos que serán los lugares a los que puede llegar el sistema
Flujo básico	Iniciar la aplicación Visualizar menú de inicio Seleccionar uno de los apartados de punto inicio o destino Visualizar el menú de búsqueda



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

Flujo alternativo 1	Si al seleccionar no se pasa a la pestaña de búsqueda quedarse en el menú de inicio
Requerimientos	R04, R07

Fuente: Elaboración Propia

Tabla 10

Tabla de descripción de caso de uso del diagrama general de calcular recorrido (CU 09)

Caso de uso #9	
Nombre	Calcular recorrido
Actor participante	Usuario
Objetivo	Calcular el camino óptimo para trazar o indicar la ruta
Disparador	Cuando se introduce punto destino y se selecciona alguna de las dos opciones: simulación o mapa
Precondición	Estar en el menú inicio y haber insertado el punto destino
Postcondición	En simulación se visualizará un recorrido en mapa se traza una ruta del camino
Flujo básico	Iniciar aplicación Insertar punto destino Seleccionar mapa o simulación
Flujo alternativo 1	Si no se inserta el punto destino no se traza la ruta
Requerimientos	R08

Fuente: Elaboración Propia

Tabla 11

Tabla de descripción de caso de uso del diagrama general de calcular ubicación (CU 10)

Caso de uso #10	
Nombre	Consultar ubicación
Actor participante	Usuario



Documento de diseño

Objetivo	Saber en qué área se encuentra el punto establecido
Disparador	Al seleccionar el punto destino o inicial
Precondición	Iniciar la aplicación, seleccionar un punto de inicio o destino
Postcondición	Realizar la consulta a la base y ubicar el punto seleccionado en el mapa
Flujo básico	Iniciar la aplicación Ir a la búsqueda de punto Seleccionar un punto Realiza la consulta
Flujo alternativo 1	En dado caso de realizar la consulta no se encuentra mandar un mensaje de error
Requerimientos	R03, R05

Fuente: Elaboración Propia

Tabla 12

Tabla de descripción de caso de uso del diagrama general de calcular tiempo (CU 11)

Caso de uso #11	
Nombre	Calcular tiempo
Actor participante	Usuario
Objetivo	Indicar el tiempo que se demorara desde el punto inicio al punto final
Disparador	Al seleccionar alguno de los dos mapas (simulación o mapa)
Precondición	Estar en el menú de inicio, haber seleccionado un punto destino
Postcondición	Seleccionar uno de los dos mapas
Flujo básico	Iniciar la aplicación Seleccionar un punto destino Seleccionar uno de los dos mapas
Flujo alternativo 1	Si no se seleccionara un punto destino e tiempo marcara 0 ya que no existiría un tiempo de camino por recorrer
Requerimientos	R09

Fuente: Elaboración Propia



Documento de diseño

Tabla 13

Tabla de descripción de caso de uso del diagrama general de visualizar tiempo (CU 12)

Caso de uso #12	
Nombre	Visualizar tiempo
Actor participante	Usuario
Objetivo	Mostrar el tiempo en pantalla que se realiza desde el punto inicio introducido y el punto destino
Disparador	Seleccionar alguno de los mapas
Precondición	Haber seleccionado un punto destino
Postcondición	Visualizar el tiempo que se realizaría caminando en cualquiera de los dos mapas
Flujo básico	Iniciar aplicación Insertar un punto destino Seleccionar uno de los mapas
Flujo alternativo 1	Si no se inserta un punto destino el tiempo mostrará 0
Requerimientos	R09,

Fuente: Elaboración Propia

Tabla 14

Tabla de descripción de caso de uso del diagrama general de mostrar ruta (CU 13)

Caso de uso #13	
Nombre	Mostrar ruta
Actor participante	Usuario
Objetivo	Mostrar el camino sea en el mapa o en la simulación
Disparador	Al haber seleccionado el punto destino, y seleccionar alguna de las opciones de mapa
Precondición	Haber seleccionado el punto destino
Postcondición	Se traza la ruta y marque el camino por el cual ir



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

Flujo básico	Iniciar la aplicación Introducir punto inicio y punto destino Elegir una de las opciones del mapa
Flujo alternativo 1	En dado caso que solo se insertara el punto destino
Requerimientos	R08

Fuente: Elaboración Propia

Tabla 15

Tabla de descripción de caso de uso del diagrama general de notificar cambios (CU 14)

Caso de uso #14	
Nombre	Notificar cambios
Actor participante	BD
Objetivo	Notificar los cambios a los usuarios
Disparador	Al realizar un cambio en la base
Precondición	El administrador debe realizar una modificación en alguno de los puntos
Postcondición	El administrador debe guardar el cambio realizado
Flujo básico	La base de datos detecta el cambio reflejado al momento que el administrador guarda el cambio. Notifica a los usuarios
Flujo alternativo 1	Cuando el usuario no tiene internet en el dispositivo se notificará cuando se detecta que ya está conectado
Requerimientos	R10

Fuente: Elaboración Propia

Tabla 16

Tabla de descripción de caso de uso del diagrama general de iniciar sesión (CU 15)

Caso de uso #15



Documento de diseño

Nombre	Iniciar sesión
Actor participante	Administrador
Objetivo	Asegurar que se una persona con autorización para editar
Disparador	Botón de entrada
Precondición	Estar en el menú de administración
Postcondición	Entrar y ver las opciones para editar los puntos
Flujo básico	Del menú de inicio pasar al menú para poder iniciar sesión Insertar las credenciales Entrar al menú de edición
Flujo alternativo 1	En dado caso que las credenciales son incorrectas no permitir el acceso al menú de edición
Requerimientos	R11

Fuente: Elaboración Propia

Tabla 17

Tabla de descripción de caso de uso del diagrama general de visualizar mapa edición (CU 16)

Caso de uso #16	
Nombre	Visualizar mapa edición
Actor participante	Administrador
Objetivo	Ver el mapa para que pueda buscar el punto a editar de manera más visual
Disparador	Al seleccionar la opción de buscar por mapa
Precondición	Haber iniciado sesión Visualizar el menú de inicio Seleccionar la búsqueda por mapa
Postcondición	visualizar los edificios y planos para encontrar el punto a editar
Flujo básico	Visualizar la escuela en vista superior Seleccionar el edificio donde está el punto que desee editar



Documento de diseño

	Seleccionar el componente a editar
Flujo alternativo 1	Si no se visualiza el mapa se puede buscar el punto por un menú de búsqueda
Requerimientos	R10

Fuente: Elaboración Propia

Tabla 18

Tabla de descripción de caso de uso del diagrama general de editar información (CU 17)

Caso de uso #17	
Nombre	Editar información
Actor participante	Administrador
Objetivo	Visualizar la información que tiene guardada el punto
Disparador	Cuando se selecciona el punto a modificar
Precondición	Buscar el punto a modificar y seleccionarlo
Postcondición	Visualizar la información que tiene guardado ese punto
Flujo básico	Buscar el punto Seleccionar el punto Se visualiza la información
Flujo alternativo 1	Si el punto no tiene información guardada estará en blanco la información
Requerimientos	R10

Fuente: Elaboración Propia

Tabla 19

Tabla de descripción de caso de uso del diagrama general de actualizar base de datos (CU 18)

Caso de uso #18	
Nombre	Actualizar base de datos
Actor participante	Administrador



Documento de diseño

Objetivo	Actualizar la base de datos
Disparador	Cuando guarda un cambio
Precondición	Cambia la información de un punto
Postcondición	Guarda el cambio realizado
Flujo básico	El administrador entrará al apartado de edición, escogerá buscar mediante palabras claves identificando el elemento a modificar
Flujo alternativo 1	El administrador entrara al apartado de edición, escogerá buscar mediante el apartado del mapa 2D identificando la edición a modificar entrara
Requerimientos	R10

Fuente: Elaboración Propia

Diagrama de actividades

Presenta un flujo de acciones que se realizan dentro del sistema, simulan el funcionamiento de un sistema cuando se realiza una acción para llevar a cabo un proceso o actividad completa. Pueden ocurrir actividades que se ejecutan al mismo tiempo. (UNAD, s.f.)

En una fase de requerimientos se pueden usar este tipo de diagramas para mostrar los diferentes casos de uso en un sistema. Para una fase de diseño se puede modelar los elementos de un software como: método, función y operación. (UNAD, s.f.)

Según la Universidad abierta y a distancia de México (s.f.), otros usos de los diagramas de actividades son:



Documento de diseño

- Modelado de procesos de negocios
- Presentación de los aspectos dinámicos de un sistema
- Definición de las reglas o la lógica de un sistema
- Identificación de los casos de uso de un sistema
- Simplificación de flujos de trabajo complejos

La siguiente tabla muestra los elementos que tiene un diagrama de actividades:

Tabla 20

Descripción de elementos de un diagrama de actividades

Nombre elemento	Función
Estado inicial y estado final	Los puntos de inicio y fin de un proceso
Actividad o estado de acción	Consta de una única actividad que inicia una secuencia de acciones
Acciones	Se ejecuta una acción o paso para hacer una tarea, sea que la ejecute el sistema o el usuario
Objeto	Materiales o datos que se crean o se utilizan en una actividad
Decisiones	Son tomadas antes de continuar una acción, las respuestas son sí o no
Sincronización	Consta de dos nodos, el nodo de bifurcación donde se ramifican los flujos concurrentes y el nodo de unión que recoge los flujos concurrentes en uno solo



Documento de diseño

Señales	Ejecución de acciones externas al sistema que pueden modificar la actividad
Carril	Agrupar por medio de columnas o categorías actividades relacionadas que son ejecutadas por actores diferentes

Fuente: UNAD. (s.f.). Diagramas de Actividades. Obtenido de UNAD:

https://unadzurlab.com/UML/U1/diagramas_de_actividades.html

Se realizaron modificaciones en los diagramas de actividades para los usuarios normal y administrador. En el diagrama del usuario normal, se ha ajustado el flujo para que la aplicación inicie directamente desde el mapa 2D en lugar del menú de inicio que se planteó inicialmente. Sin embargo, las actividades y funcionalidades siguen siendo las mismas.

En el diagrama del usuario normal, el flujo comienza en el Mapa2D, donde el usuario podrá escoger el punto destino, o cambiar al menú donde se le brinda la opción al usuario de elegir entre la simulación, el mapa o la selección de puntos iniciales o finales. Si el usuario opta por seleccionar los puntos, el sistema realizará una consulta a la base de datos para obtener la ubicación correspondiente. Por otro lado, si el usuario elige la simulación o el mapa, el sistema verificará si existen puntos guardados previamente. En caso afirmativo, calculará el tiempo y la ruta, y los mostrará según la opción seleccionada (recorrido o no). Si el usuario no elige ningún punto, el sistema consultará la opción seleccionada y mostrará la vista en primera persona o la vista del mapa.

A continuación, se muestra una imagen que representa el proceso descrito:



Documento de diseño

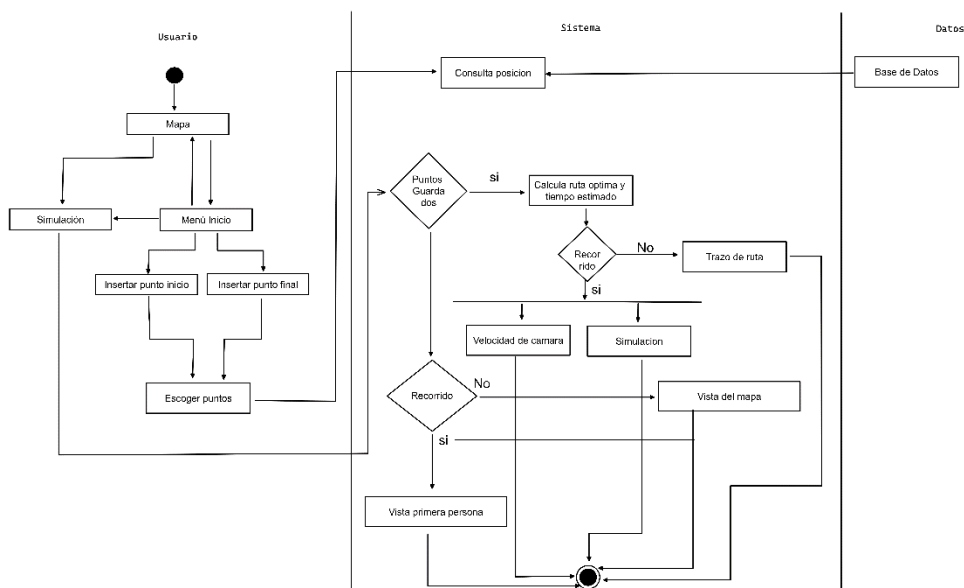


Figura 4: Diagrama de actividades del usuario (DA 01)

Fuente: Elaboración Propia

Las modificaciones que se realizaron modificaciones en la secuencia de actividades del usuario administrador. Inicialmente, se contemplaba que el usuario administrador iniciara desde un menú de inicio, donde debía iniciar sesión. Sin embargo, al ponerlo en práctica, se decidió que el inicio de sesión se realice directamente desde el mapa 2D.

En la nueva secuencia, una vez que el usuario administrador inicia sesión, tiene la opción de buscar el punto directamente o hacerlo a través del mapa. Una vez seleccionado el punto, el sistema consulta la base de datos y guarda las modificaciones realizadas. Finalmente, el usuario administrador puede cerrar sesión.

Estos cambios implicaron la eliminación de las opciones de menú de inicio y menú de búsqueda para el usuario administrador, permitiendo un flujo más directo hacia las acciones



Documento de diseño

de creación, modificación y eliminación de los puntos de los espacios, como se había planteado inicialmente.

A continuación, se muestra una imagen que refleja el proceso descrito:

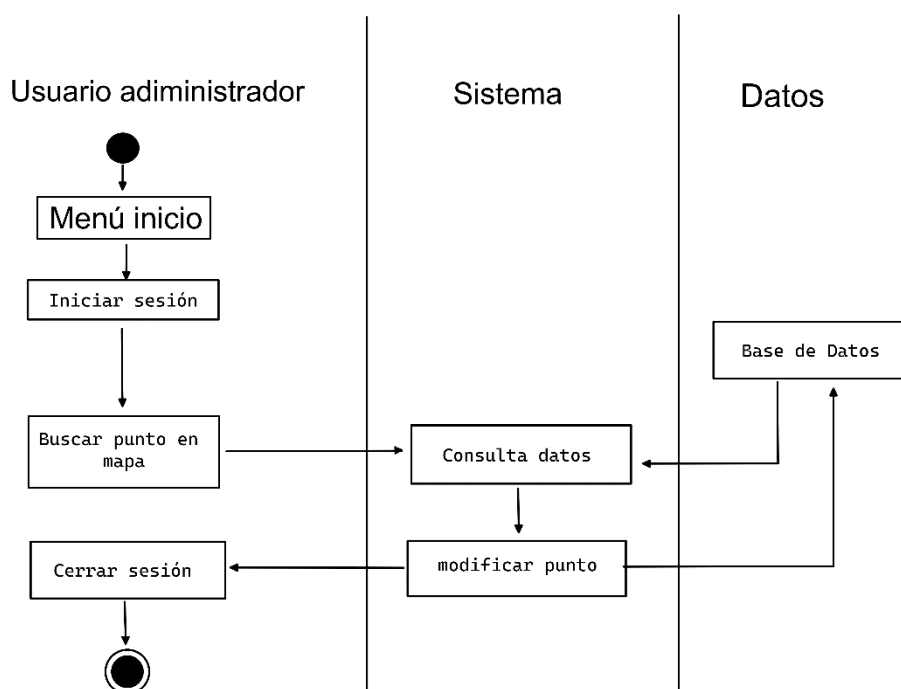


Figura 5: Diagrama de actividades del usuario administrador (DA 02)

Fuente: Elaboración Propia

Diagrama de clases

El diagrama de clases ayuda a comunicar el diseño de un programa orientado a objetos y permite visualizar las relaciones entre diferentes entidades. Para su representación en UML, se usa un rectángulo que posee tres divisiones. (OKDIARIO, 2018) En la siguiente tabla se explican las divisiones:



Documento de diseño

Tabla 21

Descripción de las divisiones en un diagrama de clases

Nivel de división	Descripción
Primera división	Se encuentra hasta arriba del rectángulo, lleva el nombre de la clase.
Segunda división	Es el centro del rectángulo, aquí se agregan los atributos o variables de instancias.
Tercera división	Se escriben las operaciones.

Fuente: OKDIARIO. (09 de 11 de 2018). Qué es un diagrama de clases. Obtenido de OKDIARIO:

<https://okdiario.com/curiosidades/que-diagrama-clases-3323710>

Los diagramas de clase usan símbolos y anotaciones para mostrar la estructura de clase (Fonseca, 2022) . Algunas de las anotaciones usadas según Fonseca (2022) son:

Clase: Se representan por medios de rectángulos con 3 divisiones

Objetos: Representados con óvalos que contienen las clases de nombre dentro de los compartimientos de clase de nombre

Atributos: Se enumeran dentro del compartimiento de atributo en un rectángulo de clase. Pueden ser atributos u operaciones de clase dependiendo si se utiliza en una relación con otro tipo de clase.

Los objetos suelen tener una flecha con una línea asociada para indicar el tipo de relación. La línea unirá dos objetos o presentará un único atributo del objeto si está definido en otra









Documento de diseño

clase (Fonseca, 2022). En la siguiente imagen se visualiza algunas de las conexiones que pueden existir:

Tabla 22

Tipos de conexiones en un diagrama de clases

Símbolo	Significado
	Asociación
	Herencia
	Realización/Implementación
	Dependencia
	Agregación
	Composición

Fuente: Fonseca, L. (27 de Junio de 2022). *Cómo crear un diagrama de clases*. Obtenido de Venngage:

<https://es.venngage.com/blog/diagrama-de-clases/>

El diagrama de clases previamente propuesto ha experimentado mejoras y cambios, los cuales se reflejan en la función del proyecto. El diseño original era de carácter general, abarcando el funcionamiento del proyecto en su totalidad. Sin embargo, se han realizado modificaciones en la estructura, se han añadido nuevas clases y se han replanteado otras con el fin de mostrar de manera más precisa el funcionamiento del proyecto.





Documento de diseño

Se mostrará de manera agrupada las clases por escenas y de manera individual de cada escena. Existen dos clases principales que hacen conexión con todas las escenas trabajadas en Unity, ya que nos permite controlar los datos entre escenas y saber de qué escena el usuario viene para la escena de MenuBusqueda.

Una de esas clases es BtnCambioEscena, esta clase es la que nos ayuda a indicarle a cada botón de cambio de escena hacia qué escena se debe dirigir; cuando se hace un cambio de mapa a mapa, obtiene los datos de posición del player, estos datos los manda a guardar a nuestra segunda clase: DatosEntreEscenas.

BtnCambioEscena
-datoEntreEscena: DatoEntreEscenas -player: GameObject -playerGPS: GameObject -bloqueado: bool
-Start() +almMenuBusqueda(int numEscena) +aAlgunaEscena() -AMenuInicio() -AMenuInicioAdmin() -SimulaCION() -Mapa() -Login() -MapaAdmin() -MapaSimulacion() -SimulacionMapa() -GuardarEscena() -EncontrarPosicionY() +getValorBloqueado() +setValorBloqueado()

Figura 7: Diagrama de clase BtnCambioEscena(DC 01)

Fuente: Elaboración Propia

La clase DatosEntreEscena es la que guarda toda la información que se están compartiendo entre escenas, esta clase hace uso del PlayerPrefs, que le indica a Unity que esos datos no los debe borrar entre cada cambio de escena.



Documento de diseño

Dado que playerprefs solo guarda tipos de variables básicos (int, float y string), se tuvo que desglosar la información de los vectores, para guardarse en 3 variables diferentes tipos float.

DatoEntreEscena
-puntoDestino: string -poscicionDatoX: float -poscicionDatoY: float -poscicionDatoZ: float -posicionPlayer: string -posicionPlayerX: float -posicionPlayerY: float -posicionPlayerZ: float -deDondeViene: int -posicionEcena: int -isGPS: int -ayuda: int -textoDatoJson: string
-Awake() -OnDestino() -SaveData() -LoadData() +getpuntoDestino() +getPoscicionDatoX() +getPoscicionDatoY() +getPoscicionDatoZ() +getPosicionPlayer() +getPosicionPlayerX() +getPosicionPlayerY() +getPosicionPlayerZ() +getDeDondeViene() +getPosicionEcena() +getIsGPS() +getAyuda() +getTextoDatoJson(float Dato) +setPuntoDestino(float Destino) +setPoscicionDatoX(float x) +setPoscicionDatoY(float y) +setPoscicionDatoZ(float z) +setPosicionPlayer(float player) +setPosicionPlayerX(float playerx) +setPosicionPlayerY(float playery) +setPosicionPlayerZ(float playerz) +setDeDondeViene(float donde) +setPosicionEcena(float posicion) +setIsGPS() +setAyuda() +setTextoDatoJson()

Figura 8: Diagrama de clase DatoEntreEscena (DC 02)

Fuente: Elaboración Propia



Documento de diseño

La clase de datos de la escena anterior manda los datos obtenidos del destino y posición a DatosEntreEscena para guardar la información y BtnCambioEscena es el auxiliar para saber a qué escena dirigirse acorde al botón que se pulse dentro de la escena.



Figura 9: Diagrama de clases de la escena menú inicio

Fuente: Elaboración Propia



Documento de diseño

Para la escena de MenuInicio tiene un script que es el que controla y actualiza los Text_InputField en base a la información que seleccione el usuario.

GameManagerMenuInicio
-textoPuntoInicio: Text_InputField -textoPuntoDestino: Text_InputField -datoEntreEscenas: DatoEntreEscenas -GameObject: BtnCambioEscena
Start() RotacionEscena() LenarDato()

Figura 10: Diagrama de clase GameManagerMenuInicio (DC 03)

Fuente: Elaboración Propia

En el mapa 3D la clase GameManager3D hace la conexión con los scripts principales para guardar la información del player, para eso, la misma clase tiene relación con el Player para poder obtener su transform cuando se detecta que el usuario cambiara de escena.

Tanto este script como el script principal del mapa2D tienen una función que es RotacionEscena, esta función ayuda a que la escena de los mapas se mantenga de manera horizontal, aunque el dispositivo móvil se encuentre de manera vertical.



Documento de diseño

GameManagerMapa3D
-datosEntreEscena : DatoEntreEscena +payer: PlayerNavMesh -textoDestino: TextoMeshProGui -PlayerCapsula: PlayerNavMesh +btnCambioEscena: btnCambioEscena
Start() RotacionEscena() PosicionPlayer() InicioDestino()

Figura 11: Diagrama de clase GameManagerMapa3D (DC 04)

Fuente: Elaboración Propia

El PlayerNavMesh es una clase que se le asigna al agente que se tiene de player en el proyecto. Dado que en este mapa la cámara está vinculada con el player, el script cuenta con variables que guardan y modifican la posición de la cámara al momento de que el usuario mueve la vista de cámara.

Como el script está siendo asignado a nuestro agente player también cuenta con las variables para calcular el tiempo y trazar el camino.



Documento de diseño

PlayerNavMesh
-agente: NavMeshAgent -lineRender: LineRender -dibujarLinea: bool -destino: vector 3 -posicion: vector 3 -slider: Slider -playerCapsulas: Camara -lookSpeed: float -touch Sta: vector 2 -isTouching: bool -IsClick : bool -verticalRotacion: float -touchDelta: float -touchX: float -touchY: float -mauseX: float -mauseY: float -rotacionX: float -rotacionY:float -tiempo: floart -puntoDestino: float -distancia: float
-Start() -Update() -MovimientoCamara() +CaminoAlDestino() -CalcularDistancia() -DibujarCamino() +setValorDestino() +setValorPosicion() +getValorDestino() +getValorPosicion()

Figura 12: Diagrama de clase PlayerNavMesh(DC 05)

Fuente: Elaboración Propia

La conexión completa de las clases que conforman la escena del mapa3D junto con las clases principales es la siguiente:



Documento de diseño

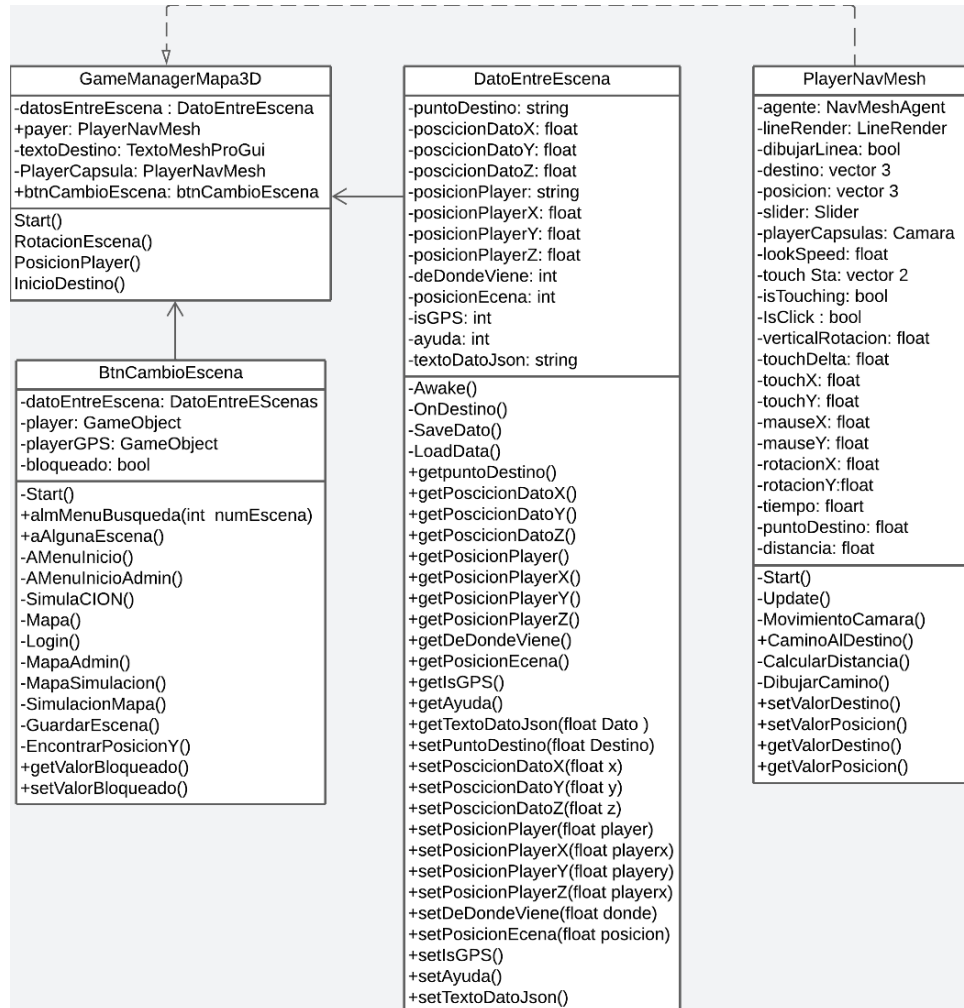


Figura 13: Diagrama de clases de la escena del mapa 3D

Fuente: Elaboración Propia

Para el mapa2D se tiene la siguiente conexión de las clases usadas para esa escena.



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

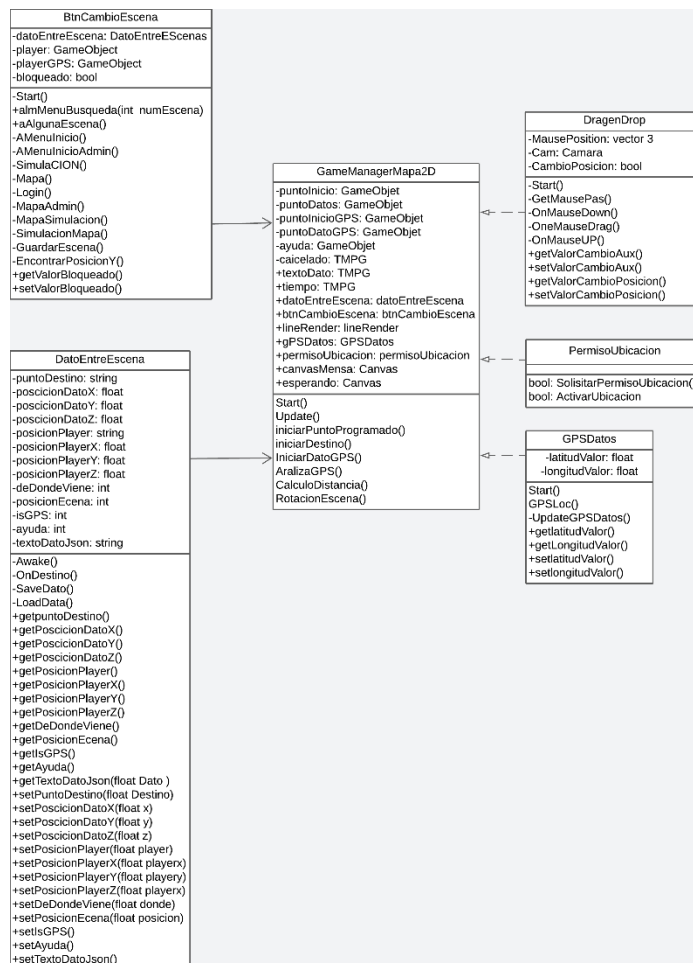


Figura 14: Diagrama de clases de la escena mapa 2D

Fuente: Elaboración Propia

El permiso Ubicación es el que se estará consultando para ver que el usuario haya otorgado los permisos para que la app pueda acceder al GPS del móvil, mediante una variable booleana se puede saber si concedieron los permisos o no.



Documento de diseño

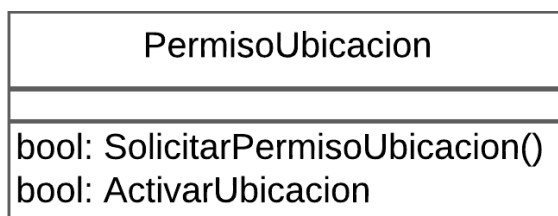


Figura 15: Diagrama de clase PermisoUbicacion(DC 06)

Fuente: Elaboración Propia

GPSDatos se estará consultando esta clase en el Update del script GameManagerMapa2D, cuando el usuario haya dado los permisos y activado el GPS. Si esas condiciones se cumplen el script estará consultando la latitud y longitud que el GPS este proporcionando para poder ir actualizando en tiempo real la posición del punto.

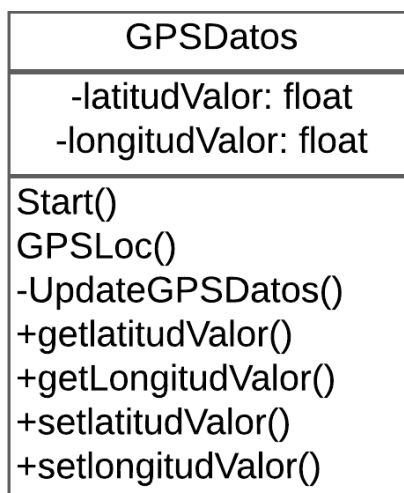


Figura 16: Diagrama de clase GPSDatos(DC 07)

Fuente: Elaboración Propia



Documento de diseño

La siguiente es la clase principal de la escena Mapa2D

GameManagerMapa2D
-puntoInicio: GameObjet -puntoDatos: GameObjet -puntoInicioGPS: GameObjet -puntoDatoGPS: GameObjet -ayuda: GameObjet -caicelado: TMPG +textoDato: TMPG +tiempo: TMPG +datoEntreEscena: datoEntreEscena +btnCambioEscena: btnCambioEscena +lineRender: lineRender +gPSDatos: GPSDatos +permisoUbicacion: permisoUbicacion +canvasMensa: Canvas +esperando: Canvas
Start() Update() iniciarPuntoProgramado() iniciarDestino() IniciarDatoGPS() AralizaGPS() CalculoDistancia() RotacionEscena()

Figura 17: Diagrama de clase GameManagerMapa2D(DC 08)

Fuente: Elaboración Propia

La clase DragAndDrop detecta cuando el usuario pulsa el player, si detecta que se seleccionó el player da aviso de que está siendo arrastrado y va guardando la posición constantemente hasta que detecta que ha sido soltado.



Documento de diseño

DragenDrop
-MausePosition: vector 3
-Cam: Camara
-CambioPosicion: bool
-Start()
-GetMausePas()
-OnMauseDown()
-OneMauseDrag()
-OnMauseUP()
+getValorCambioAux()
+setValorCambioAux()
+getValorCambioPosicion()
+setValorCambioPosicion()

Figura 18: Diagrama de clase DragenDrop (DC 09)

Fuente: Elaboración Propia

Para la escena de MenuBusqueda la estructura de las clases utilizadas es la siguiente:



Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria de Ingeniería campus

Zacatecas



Documento de diseño

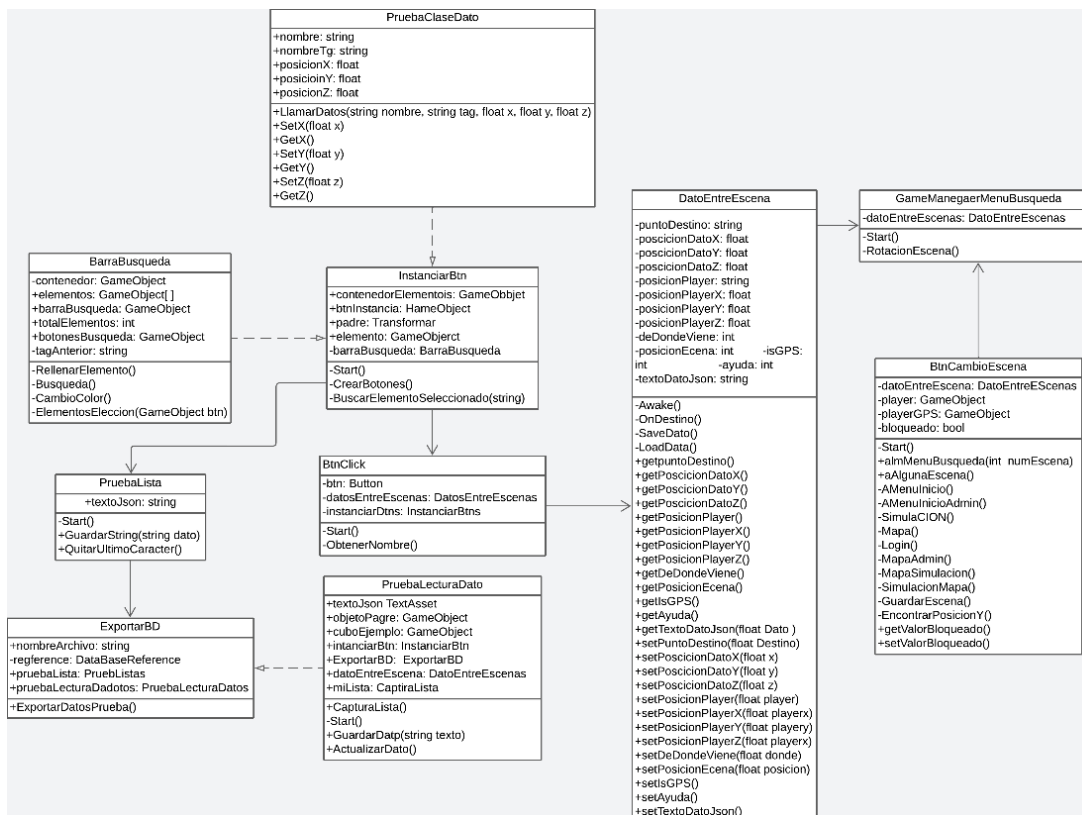


Figura 19: Diagrama de clases menú inicio y comportamiento de los boto botones

Fuente: Elaboración Propia

EL script principal de la escena solo contiene la rotación de la misma, que fija la escena en vista vertical para evitar que el usuario pueda rotarla a una vista horizontal



Documento de diseño

GameManegaerMenuBusqueda
-datoEntreEscenas: DatoEntreEscenas
-Start() -RotacionEscena()

Figura 20: Diagrama de clase GameManagerMenuBusqueda (DC 10)

Fuente: Elaboración Propia

La siguiente imagen hace referencia a la clase de PruebaClaseDatos, esta clase es instanciada al momento de crear todos los elementos para que cada elemento guarde sus datos en el script.

PruebaClaseDato
+nombre: string +nombreTg: string +posicionX: float +posicioinY: float +posicionZ: float
+LlamarDatos(string nombre, string tag, float x, float y, float z) +SetX(float x) +GetX() +SetY(float y) +GetY() +SetZ(float z) +GetZ()

Figura 21: Diagrama de clase PruebaClaseDato (DC 11)

Fuente: Elaboración Propia

Cuando se consulta la base de datos y se obtienen los datos, esos datos se utilizan para ir construyendo los botones que aparecen en el scroll del menú de búsqueda.



Documento de diseño

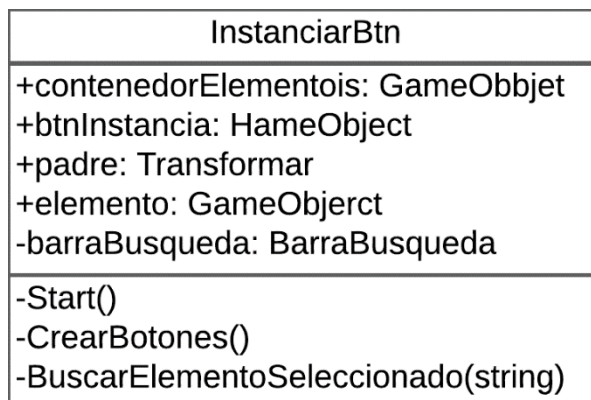


Figura 22: Diagrama de clase IniciarBtn (DC 12)

Fuente: Elaboración Propia

BtnClick es una clase que se instancia al momento de crear el botón. Cuando uno de los botones es pulsado la información que se encuentra en ese botón es mandada a DatosEntreEscenas para poder saber qué punto fue el consultado.

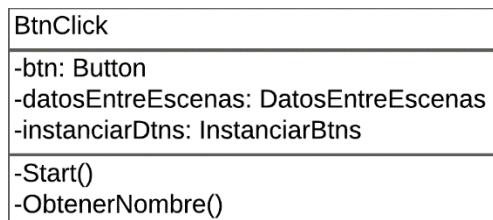


Figura 23: Diagrama de clase BtnClick (DC 13)

Fuente: Elaboración Propia

El siguiente script, toma el string existente de DatosEntreEscenas para hacer los botones en lo que recibe una señal de la exportación de datos por si existió algún cambio en la base de datos.



Documento de diseño

PruebaLecturaDato
+textoJson TextAsset +objetoPagre: GameObject +cuboEjemplo: GameObject +instanciarBtn: InstanciarBtn +ExportarBD: ExportarBD +datoEntreEscena: DatoEntreEscenas +miLista: CaptiraLista
+CapturaLista() -Start() +GuardarDatp(string texto) +ActualizarDato()

Figura 24: Diagrama de clase PrubaLecturaDatos (DC 14)

Fuente: Elaboración Propia

Esta clase está asociada con el componente del scroll de elementos de búsqueda, el script ayuda a detectar si alguno de los botones de tags fue presionado y si fue así mostrar los elementos relacionados con ese tag, de igual forma es el responsable de actualizar los elementos acordes a lo que el usuario introduzca en la entrada de búsqueda.

BarraBusqueda
-contenedor: GameObject +elementos: GameObject[] +barraBusqueda: GameObject +totalElementos: int +botonesBusqueda: GameObject -tagAnterior: string
-RellenarElemento() -Busqueda() -CambioColor() -ElementosEleccion(GameObject btn)

Figura 25: Diagrama de clase BarraBusqueda (DC 15)

Fuente: Elaboración Propia



Documento de diseño

La clase PruebaLista guarda y actualiza el string que se extrae de la base de datos.

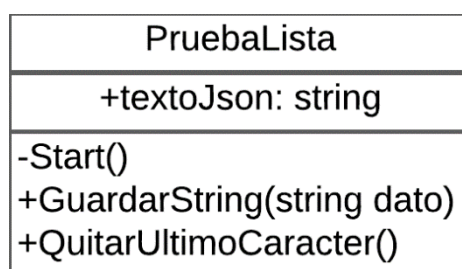


Figura 26: Diagrama de clase PruebaLista (DC 16)

Fuente: Elaboración Propia

Hace la conexión con la base de datos, si la conexión es exitosa se estará guardando los datos para la creación de los botones.

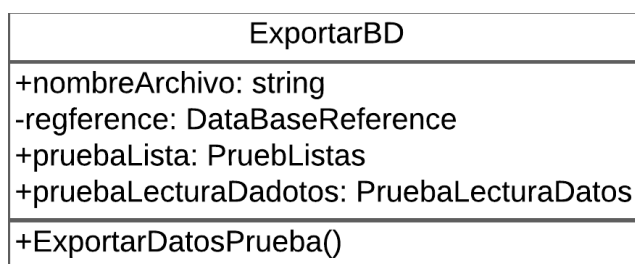


Figura 27: Diagrama de clase ExportarBD(DC 17)

Fuente: Elaboración Propia



Documento de diseño

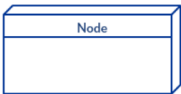

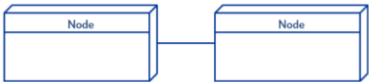
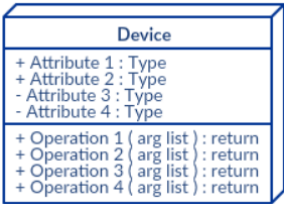
Diagrama de despliegue

Estos diagramas se utilizan para visualizar los procesadores/nodos/dispositivos de hardware de un sistema. Muestran la arquitectura de ejecución del sistema como los entornos de ejecución y el middleware que los conecta. (creately, 2022)

Para dibujar un diagrama de despliegue se usan las siguientes anotaciones y elementos:

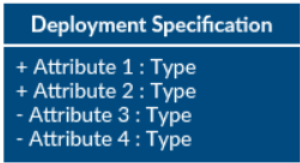
Tabla 23

Simbología para los diagramas de despliegue

Nombre	Símbolo	Función
Nodos		Es una entidad física que se ejecuta uno o más componentes, subsistemas o ejecutables
Artefactos		Son elementos concretos que son causados por un proceso de desarrollo
Asociación de comunicación		Línea entre dos nodos que indican el camino de la comunicación entre los nodos
Dispositivos		Se utiliza para representar un recurso físico.



Documento de diseño

Especificaciones de despliegue		Especifica los despliegues, son un archivo de configuración, como un archivo de texto o documento XML.
--------------------------------	---	--

Fuente: creately. (18 de Octubre de 2022). *La Guía Fácil de los Diagramas de Despliegue UML*. Obtenido de creately: <https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>

Un nodo puede tener nodos incluidos en su interior, indicando que son sistemas separados incluidos dentro del mismo físico. Es esto se le conocería como nodo compuesto (DiagramasUML, s.f.)

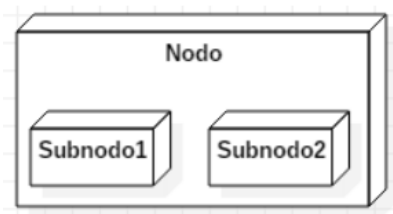


Figura 28: Notación de un nodo con subnodos

Fuente: DiagramasUML. (s.f.). *Diagrama de despliegue*. Obtenido de DiagramasUML: <https://diagramasuml.com/despliegue/>

Mediante el diagrama de despliegue podemos visualizar los dispositivos a utilizar junto con sus tecnologías correspondientes para dar un funcionamiento a la herramienta planteada la cual se divide en 3 partes las cuales se comunican mediante el protocolo HTTP, las cuales son:

- Apartado usuario
- Apartado Administrado



Documento de diseño

- Apartado de la Base de Datos

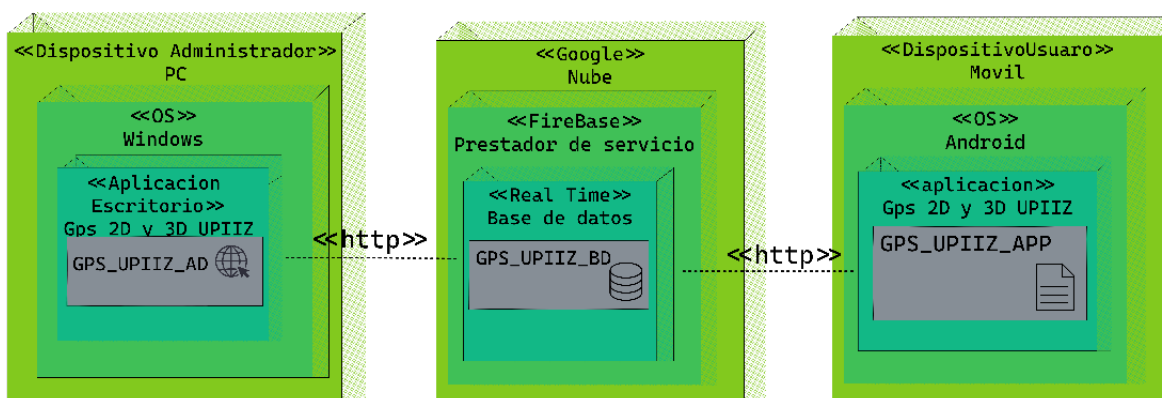


Figura 29: Diagrama de despliegue

Fuente: Elaboración Propia

Diagrama de despliegue del usuario: el usuario necesitará un dispositivo móvil el cual deberá tener el sistema Android, donde podrá ejecutar la aplicación del “GPS DE LA UPIIZ”, en dicha aplicación el usuario podrá consultar y visualizar la ubicación de los diferentes espacios de la institución junto con una ruta a seguir para llegar al lugar solicitado si así lo desea el usuario.



Documento de diseño

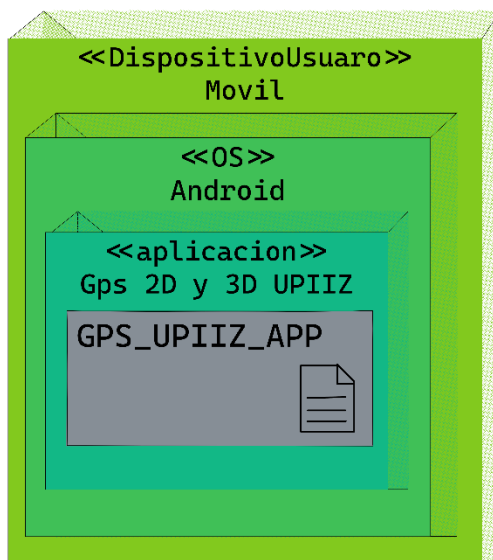


Figura 30: Diagrama de despliegue del Usuario (DES 03)

Fuente: Elaboración Propia

En este diagrama, se muestra que el administrador utilizará una computadora con sistema operativo Windows para abrir la aplicación de escritorio. El administrador accederá al apartado correspondiente ingresando un usuario y contraseña válidos, lo que le permitirá acceder a la plataforma del administrador. Desde esta plataforma, el administrador podrá realizar modificaciones en la información de los puntos, las cuales se reflejarán en la base de datos.



Documento de diseño

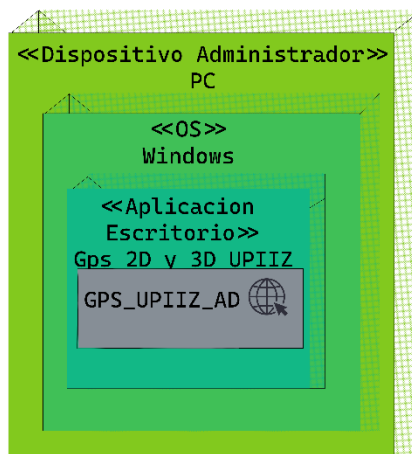


Figura 31: Diagrama de despliegue del Administrador (DES 01)

Fuente: Elaboración Propia

Diagrama de despliegue de la Base de Datos: la base de datos se alojará en la nube donde el prestador del servicio será Firebase, donde el servicio que se usará en este caso será Real Time Database, la cual nos permite tener actualizada la información en todas las aplicaciones que cuenten con la base de datos proporcionada para tener los datos lo más actualizados posibles donde de igual manera no será necesario tener el acceso constante al internet para hacer las consultas a la herramienta y solo se actualizará cuando tenga el acceso a internet y la base de datos notifique que los datos de dicha BD se actualizaron,



Documento de diseño



Figura 32: Diagrama de despliegue de la Base de Datos (DES 02)

Fuente: Elaboración Propia

Diseño de prototipos

El prototipo hace referencia a una representación aparente de una idea de producto, servicio o negocio. Se le puede considerar una simulación de lo que pudiera ser la idea que se desea reflejar. (EmprenderAConciencia, s.f.)

Un prototipo ayuda a:

- Entender la interacción del usuario (EmprenderAConciencia, s.f.)
- Mejorar la experiencia de uso (EmprenderAConciencia, s.f.)
- Demostrar el alcance del producto (EmprenderAConciencia, s.f.)
- Validar la apariencia (EmprenderAConciencia, s.f.)



Documento de diseño

La figura 27 muestra la visualización del producto en móvil. La pantalla principal muestra el menú de inicio donde se visualizará el logo, el apartado para indicar los puntos de inicio y destino y dos botones: simulación (mostrará el modelo en 3D y un recorrido en primera persona) y el de mapa (que se muestra la vista 2D de modelo).

Cuando el usuario selecciona el inserta los puntos de inicio y destino, ambas opciones dirigen a la otra ventana de búsqueda de puntos donde se puede buscar el elemento en una lista o directamente en una barra de búsqueda.

El botón de simulación dirige a la simulación en primera persona donde la cámara se estará moviendo hacia el punto destino introducido. Se visualizará el tiempo de caminata que se realiza hacia el punto introducido, en la parte inferior se verá hacia donde se dirige la caminata, que también será un botón para cambiar el punto destino. Desde la ventana de simulación el usuario puede regresar al menú de inicio o cambiar el tipo de vista a una 2D. Esta opción cuenta con un scroll que permite modificar la velocidad de cámara.

La opción de mapa dirigirá a la vista en 2D que tiene las mismas características que la opción de simulación, salvo que en esta opción no se encuentra la velocidad de cámara.



Documento de diseño

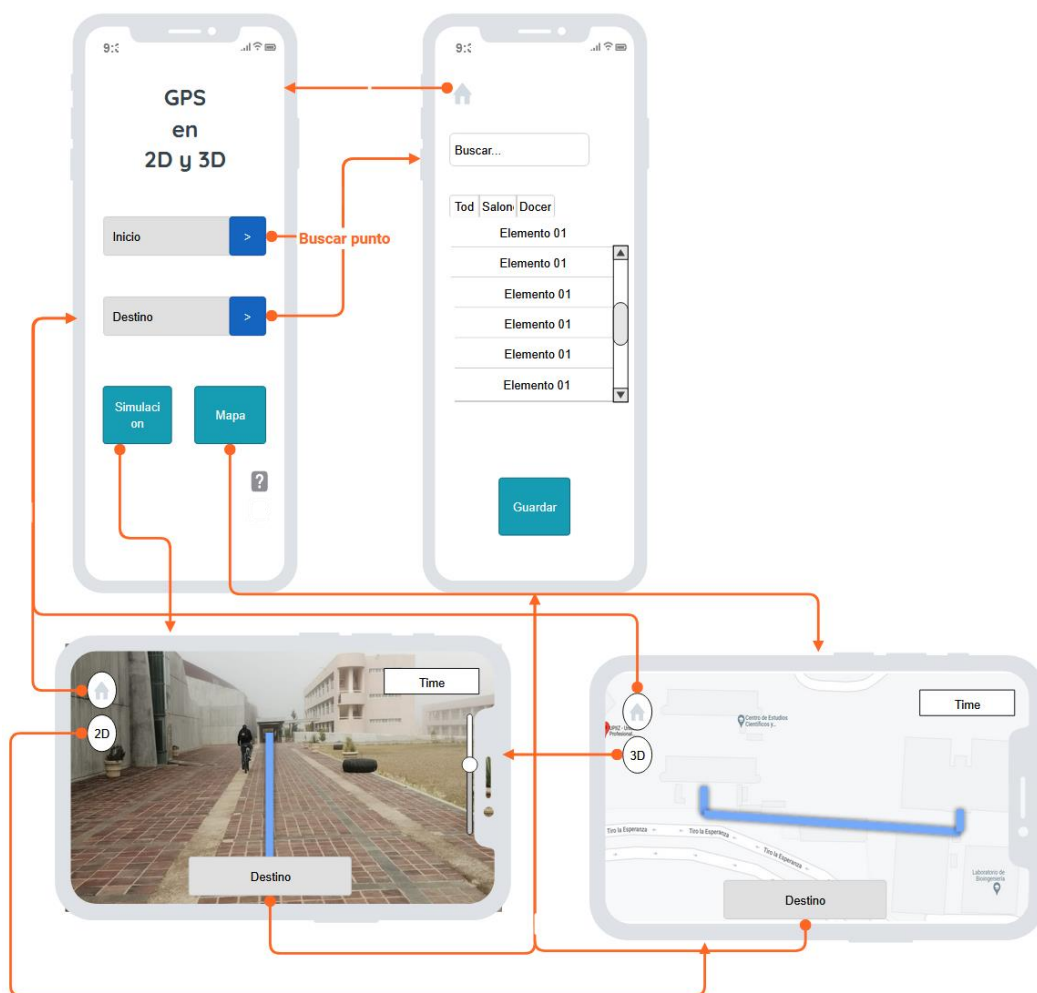


Figura 33: Diseño del apartado móvil

Fuente: Elaboración Propia



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

En la figura 34 se visualiza el apartado de escritorio y las conexiones que tiene. Cuenta con las mismas características que el apartado móvil, su principal diferencia es el apartado del administrador:



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño



Figura 34: Diseño del apartado de escritorio

Fuente: Elaboración Propia



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

La primera parte funciona igual que el apartado móvil, teniendo las mismas conexiones, como se ve a continuación en la figura 35. La única diferencia es el icono en el menú principal para entrar al menú administrador.



Documento de diseño

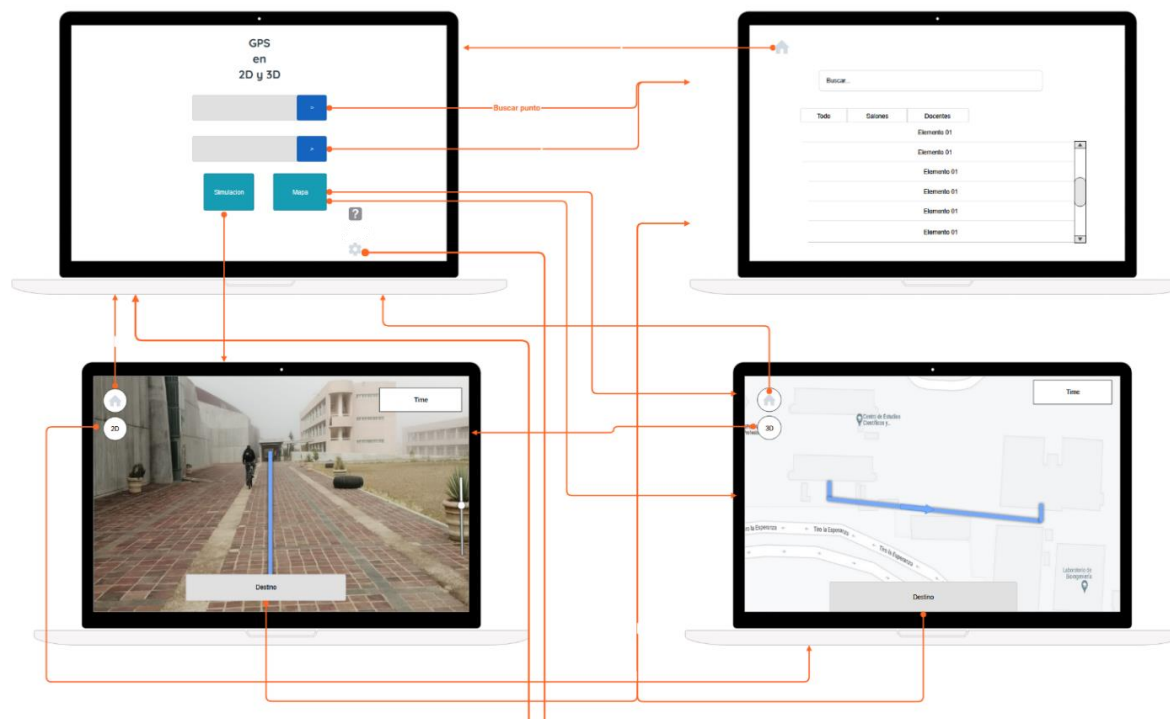


Figura 35: Interacción usuario

Fuente: Elaboración Propia



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

Cuando el usuario entra al menú inicio de sección se solicitarán las credenciales. Introducido las credenciales correspondientes se dirige a el mapa en una vista superior.

El mapa es similar al mapa del usuario normal con algunas diferencias que nos permiten seleccionar un edificio y mostrar en la parte inferior de la pantalla el número de piso que deseamos visualizar. De esta manera, podremos agregar, eliminar o editar los puntos que pertenecen a dicho piso.

En la parte de edición, podremos agregar un nombre y una descripción si es un docente, un espacio o un aula. También podemos asignar etiquetas que indiquen qué tipo de espacio es, como aula, oficina, cubículo o Áreas comunes.



Documento de diseño

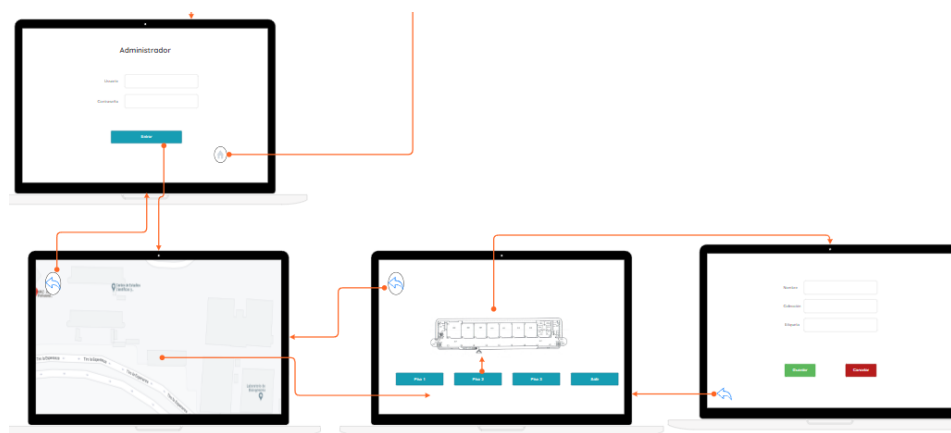


Figura 36: Interacción usuario administrador

Fuente: Elaboración Propia



Documento de diseño

La siguiente tabla identifica los elementos de diseño con un ID y la descripción.

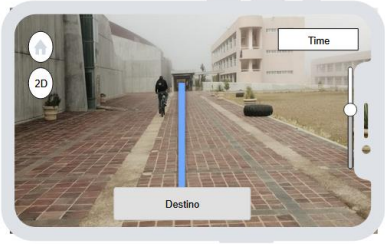
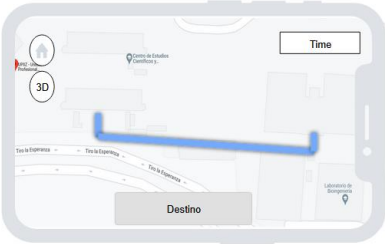
Tabla 24

Identificador de los diagramas de diseño

Diagrama de diseño		Descripción
DD 01		Movil Menu inicio, sera la primera pantalla que visualizara el usuario, puede seleccionar el introducir un punto inicio o destino, seleccionar la simulacion (recorrido en primera persona) o el mapa (mapa 3D)
DD 02		Movil Menu de busqueda Se muestra esta opcion cuando el usuario selecciona introducir punto destino o inicial. Aparece un apartado de busqueda para que encuentre la opcion el elemento que necesita, puede ser mediante busqueda o buscando el elemento en la lista.





Documento de diseño

DD 03		<p>Movil</p> <p>Visualizacion de simulacion del recorrido. En la esquina superior derecha se estara viendo el tiempo que se hace desde el punto inicial al punto destino introducido, un poco mas abajo se encontrara un scroll que permite cambiar la velocidad de la camara.</p> <p>Centrado en la parte inferior se estara señalando el destino introducido, que tambien puede llevar al menu de busqueda para cambiar el destino.</p> <p>Los botones que se encuentran en la esquina superior izquierda son para regresar al menu inicio y el segundo para el cambio de vista.</p>
DD 04		<p>Movil</p> <p>La vista mapa o vista 2D</p> <p>Se presenta una visualizacion de la superficie, donde se traza un camino de edificio a edificio de acuerdo al punto inicio y destino introducido, cuenta con los mismos botones de regreso, cambio, destino e igual se puede visualizar el tiempo, la</p>

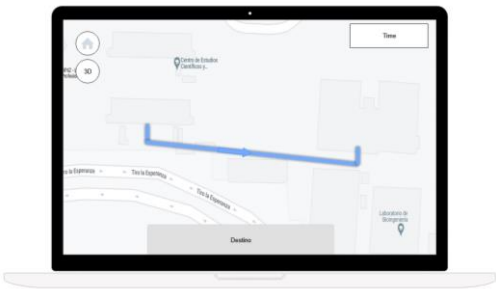


Documento de diseño

		diferencia que no cuenta con un scroll.
DD 05		<p>Escritorio</p> <p>Menu inicio, sera la primera pantalla que visualizara el usuario, puede seleccionar el introducir un punto inicio o destino, seleccionar la simulacion (recorrido en primera persona) o el mapa (mapa 3D)</p> <p>A diferencia del movil este contara con una ranura que dirige al apartado de administrador.</p>
DD 06		<p>Escritorio</p> <p>Menu de busqueda</p> <p>Se muestra esta opcion cuando el usuario selecciona introducir punto destino o inicial. Aparece un apartado de busqueda para que encuentre la opcion el elemento que necesita, puede ser mediante busqueda o</p>



Documento de diseño

		buscando el elemento en la lista.
DD 07		<p>Escritorio</p> <p>Visualizacion de simulacion del recorrido. En la esquina superior derecha se estara viendo el tiempo que se hace desde el punto inicial al punto destino introducido, un poco mas abajo se encontrara un scroll que permite cambiar la velocidad de la camara.</p> <p>Centrado en la parte inferior se estara señalando el destino introducido, que tambien puede llevar al menu de busqueda para cambiar el destino.</p> <p>Los botones que se encuentran en la esquina superior izquierda son para regresar al menu inicio y el segundo para el cambio de vista.</p>
DD 08		<p>Escritorio</p> <p>La vista mapa o vista 2D</p> <p>Se presenta una visualizacion de la superficie, donde se</p>

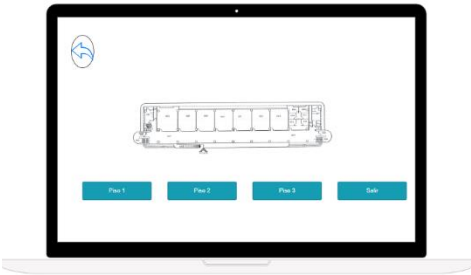
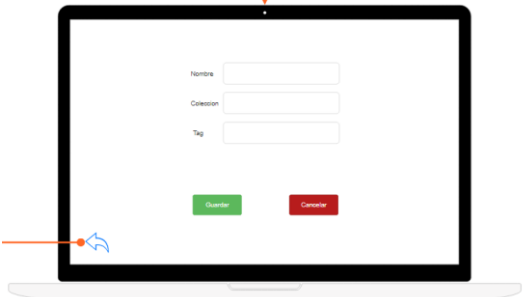


Documento de diseño

		traza un camino de edificio a edificio de acuerdo al punto inicio y destino introducido, cuenta con los mismos botones de regreso, cambio, destino e igual se puede visualizar el tiempo, la diferencia que no cuenta con un scroll.
DD 09		El inicio de sesion del administrador, pedira correo y contraseña para poder ingresar para editar los puntos, cuenta con um boton de regreso.
DD 10		En el mapa se visualizara la parte superior de los modelos, donde puede seleccionar alguno para mostrar las zonas internas que puede editar.



Documento de diseño

DD 11		Al seleccionar los pisos, aparecerán unos botones que nos ayudarán a identificar el piso del edificio en el que queremos agregar un punto. Dependiendo del botón seleccionado, se mostrará el piso correspondiente, así como los espacios y puntos previamente creados que lo conforman.
DD 12		Cuando se selecciona un punto en el mapa, se mostrará la información almacenada del mismo, permitiendo al usuario administrador editar dicha información. En caso de crear un nuevo punto, la información estará vacía para que el usuario pueda agregarla según sea necesario.

Fuente: Elaboración propia



Documento de diseño

Referencias

- ACENS. (28 de Febrero de 2014). *Bases de datos NoSQL*. . Obtenido de ACENS part of telefónica tech:
https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiujuzFuYv7AhUyMEQIHYP8DTkQFnoECAwQAQ&url=https%3A%2F%2Fwww.acens.com%2Fwp-content%2Fimages%2F2014%2F02%2Fbddd-nosql-wp-acens.pdf&usg=AOvVaw3YuKFtHI9TOeX_na4xapoW
- Cervantes, H. (02 de Marzo de 2010). *Arquitectura de Software*. Obtenido de Buzz:
<https://sg.com.mx/revista/27/arquitectura-software>
- Couchbase. (s.f.). *Couchbase Lite*. Obtenido de Couchbase:
<https://www.couchbase.com/products/lite>
- Couchbase. (s.f.). *Pricing*. Obtenido de Couchbase: <https://www.couchbase.com/pricing>
- creately. (18 de Octubre de 2022). *La Guía Fácil de los Diagramas de Despliegue UML*. Obtenido de createy: <https://createy.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/>
- Creately. (9 de Septiembre de 2022). *Use Case Diagram Relationships Explained with Examples*. Obtenido de Createy: <https://createy.com/blog/diagrams/use-case-diagram-relationships/>
- DiagramasUML. (s.f.). *Diagrama de despliegue*. Obtenido de DiagramasUML:
<https://diagramasuml.com/despliegue/>
- EmprenderAConciencia. (s.f.). *Diseño de prototipos*. Obtenido de EmprenderAConciencia:
<https://www.emprendeaconciencia.com/diseno-prototipo>



Documento de diseño

Firebase. (s.f.). *Almacena y sincroniza*. Obtenido de Firebase:

https://firebase.google.com/products/realtime-database?gclid=Cj0KCQjwnP-ZBhDiARIsAH3FSRfo9TUXNCaCpYnatq8lklic30Hei4FIfmG2VIYu90ygJE5nrfb-5zIaAoiAEALw_wcB&gclsrc=aw.ds

Firebase. (s.f.). *Firestore*. Obtenido de Firebase: <https://firebase.google.com/docs/firestore>

Firebase. (s.f.). *Pricing*. Obtenido de Firebase: <https://firebase.google.com/pricing>

Fonseca, L. (27 de Junio de 2022). *Cómo crear un diagrama de clases*. Obtenido de Venngage: <https://es.venngage.com/blog/diagrama-de-clases/>

IBM Cloud Education. (28 de Octubre de 2020). *Arquitectura de 3 niveles*. Obtenido de IBM: <https://www.ibm.com/mx-es/cloud/learn/three-tier-architecture>

Microsoft. (29 de Septiembre de 2022). *Estilo de arquitectura de n niveles*. Obtenido de Microsoft: <https://learn.microsoft.com/es-es/azure/architecture/guide/architecture-styles/n-tier>

Novoseltseva, E. (2 de junio de 2020). *Los 5 principales patrones de Arquitectura de Software*. Obtenido de Apiumhub: <https://apiumhub.com/es/tech-blog-barcelona/principales-patrones-arquitectura-software/>

OCI. (31 de Julio de 2014). *¿Qué es una base de datos?* Obtenido de Oracle Cloud: <https://www.oracle.com/mx/database/what-is-database/>

OKDIARIO. (09 de 11 de 2018). *Qué es un diagrama de clases*. Obtenido de OKDIARIO: <https://okdiario.com/curiosidades/que-diagrama-clases-3323710>

Oracle . (s.f.). *Oracle Berkeley DB*. Obtenido de Oracle : <https://www.oracle.com/mx/database/technologies/related/berkeleydb.html>

PMOInformatica. (18 de Febrero de 2021). *Diagrama de casos de uso: Definición*. Obtenido de PMOInformatica: <http://www.pmoinformatica.com/2021/02/diagrama-de-casos-de-uso-definicion-elementos-ejemplos-como-hacer.html>



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería campus
Zacatecas



Documento de diseño

Riak. (s.f.). *LevelBD*. Obtenido de Riak:

<https://docs.riak.com/riak/kv/latest/setup/planning/backend/leveldb/index.html>

Symisc UnQLite. (2018). *About UnQLite*. Obtenido de Symisc UnQLite:

<https://unqlite.org/>

UNAD. (s.f.). *Diagramas de Actividades*. Obtenido de UNAD:

https://unadzsurlab.com/UML/U1/diagramas_de_actividades.html

Unity. (s.f.). *Construyendo un NavMesh*. Obtenido de UnityDocumentation:

<https://docs.unity3d.com/es/2019.4/Manual/nav-BuildingNavMesh.html>