# DOCUMENTO DE DISEÑO

Hotel y eventos.

<<Ingeniería en Software>>

<<Nombre del autor del documento>>

Documento de diseño.

Hotel y eventos.

Puebla, México, 15 de mayo de 2024.

# REVISIÓN Y CONTROL DE CAMBIOS

**Revisión y Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombres y Apellidos** | **Cargo** | **Versión Aprobada** | **Fecha** |
| Carro Gutiérrez Jesús |  |  |  |
| Fernández Jiménez Pedro |  |  |  |
| López Pérez Diego Alexis |  |  |  |
| Reyes Arroyo Gael |  |  |  |
| Zepeda Minero Manuel |  |  |  |

**Control de Cambios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Autor** | **Versión** | **Descripción del cambio** |
| 15/05/2024 | López Pérez Diego Alexis | 1.0 | Creación del documento, de la tabla de contenido y primera redacción del resumen. |
| 16/05/2024 | López Pérez Diego Alexis | 1.0 | Primer redacción de las generalidades. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLA DE CONTENIDO**

Contenido

[DOCUMENTO DE DISEÑO 1](#_Toc166776712)

[REVISIÓN Y CONTROL DE CAMBIOS 2](#_Toc166776713)

[TABLA DE IMÁGENES 3](#_Toc166776714)

[Resumen 4](#_Toc166776715)

[1. Generalidades 5](#_Toc166776716)

[1.1. Propósito del sistema. 5](#_Toc166776717)

[1.2. Objetivos del diseño 6](#_Toc166776718)

[1.3. Definiciones, estereotipos UML y abreviaturas 7](#_Toc166776719)

[1.4. Referencias 9](#_Toc166776720)

[2. Arquitectura del sistema. 9](#_Toc166776721)

[2.1. Arquitectura propuesta 9](#_Toc166776722)

[2.1.1. Visión global. 9](#_Toc166776723)

[2.1.2. Diseño de la Arquitectura 9](#_Toc166776724)

[2.1.3. Diseño de los subsistemas 9](#_Toc166776725)

[2.1.4. Vista de casos de uso 10](#_Toc166776726)

[2.1.4.1. Vista lógica 10](#_Toc166776727)

[3. Diseño detallado 11](#_Toc166776728)

[3.1. Introducción. 11](#_Toc166776729)

[3.2. Diseño de casos de uso reales 11](#_Toc166776730)

[3.3. Diseño detallado de las clases. 13](#_Toc166776731)

[4. Diseño de la arquitectura de módulos del sistema 13](#_Toc166776732)

[5. Diseño físico de datos 13](#_Toc166776733)

[6. Modelo de datos y diccionario de datos 13](#_Toc166776734)

[6.1. Modelo Entidad – Relación. (Diagrama de Clases) 13](#_Toc166776735)

[6.2. Diccionario de datos. 13](#_Toc166776736)

[7. Prototipo de pantallas 14](#_Toc166776737)

## TABLA DE IMÁGENES

<<Agregue aquí la tabla de ilustraciones>>

# Resumen

<<Descripción: En esta sección se debe realizar una presentación del documento especificando el propósito y enumerar las secciones y contenidos que lo componen>>

El presente documento propone un diseño de software para un sistema de reserva de habitaciones y eventos del “Hotel Chachapa”, el principal propósito del documento es poder desglosar y explicar cada punto del diseño propuesto para que el “Hotel Chachapa” pueda agilizar y automatizar la gestión de reservas tanto de sus habitaciones como de sus eventos, ofreciendo así a sus clientes un calendarización rápida y segura en su hotel, el desglose es el siguiente:

1. Generalidades:

En este apartado se describen los objetivos del proyecto, el diseño general del sistema y su correlación con los beneficios esperados. Se detallarán definiciones, estereotipos UML, abreviaturas, y cómo el sistema mejorará la eficiencia operativa y la experiencia del cliente.

1. Arquitectura del sistema:

En este apartado se define la arquitectura a usar, incluyendo los componentes principales del sistema y cómo interactúan entre sí. Se explicará cómo se implementará esta arquitectura para garantizar la escalabilidad, seguridad y rendimiento del sistema.

1. Diseño detallado:

Definido lo anterior, se procederá a detallar el diseño propuesto para solucionar el problema planteado. Esto incluirá diagramas de clases, diagramas de secuencia, y otros modelos UML que representen la estructura y el comportamiento del sistema. Cada clase identificada se describirá con sus atributos y métodos, así como sus interacciones.

1. Diseño de la arquitectura del módulo del sistema:

En este apartado se profundiza en el diseño de la arquitectura de cada módulo del sistema, buscando cumplir con los requisitos funcionales y no funcionales. Se describirán los componentes internos de los módulos, sus interfaces y la comunicación entre ellos.

1. Diseño físico de datos:

Aquí se detallará el modelo de datos físico, incluyendo el diseño de la base de datos, las tablas, las relaciones entre ellas y las restricciones de integridad. Se describirá cómo se almacenarán y accederán los datos para asegurar la consistencia, integridad y eficiencia del sistema.

1. Modelo de Datos y Diccionario de Datos:

En esta sección se presentará el modelo de datos lógico y físico del sistema. También se proporcionará un diccionario de datos detallado que describa cada elemento de datos en el sistema, su tipo de datos, restricciones y cualquier otra información relevante para asegurar una comprensión clara y consistente de la estructura de datos.

1. Prototipo de Pantallas:

Con la ayuda del software Figma, se presenta el primer prototipo de la interfaz de usuario del sistema como entregable. Este prototipo incluye las pantallas principales, los flujos de navegación y las interacciones del usuario con el sistema.

# Generalidades

## Propósito del sistema.

<<Se presenta una visión global y resumida de la arquitectura del sistema y de los objetivos generales del diseño. Se describen las influencias con los requisitos funcionales y no funcionales del sistema y las decisiones y prioridades establecidas – eficiencia vs. portabilidad, por ejemplo.>>

El propósito del Sistema Integral de Gestión Hotelera y de Eventos (SIGHE) para el “Hotel Chachapa” es proporcionar una solución integral y automatizada para gestionar las operaciones diarias del hotel, incluyendo la reserva de habitaciones y la planificación y ejecución de eventos. La arquitectura del sistema está diseñada para optimizar la eficiencia operativa y mejorar la experiencia del cliente, permitiendo una gestión rápida, segura y precisa de todas las actividades del hotel.

La visión global del sistema incluye los siguientes componentes clave:

* *Interfaz de Usuario (Cliente):* Aplicaciones web y móviles accesibles para los clientes y el personal del hotel.
* *Servidor de Aplicaciones:* Gestión de la lógica de negocio y procesamiento de solicitudes.
* *Base de Datos:* Almacenamiento y gestión de datos de reservas, eventos, clientes, facturación y servicios adicionales.

La especificación general de Requisitos Funcionales y No Funcionales es:

|  |  |
| --- | --- |
| Funcionales | No Funcionales: |
| Gestión de reservas de habitaciones y eventos. | Eficiencia: El sistema debe procesar solicitudes y operaciones con rapidez para garantizar una experiencia de usuario fluida. |
| Administración de perfiles de clientes y empleados. | Portabilidad: El sistema debe ser accesible desde diferentes dispositivos y plataformas (navegadores web, dispositivos móviles). |
| Generación de facturas y seguimiento de pagos. | Seguridad: Protección de datos sensibles de clientes y transacciones. |
| Integración con plataformas de reservas externas (Booking.com, Expedia). | Escalabilidad: Capacidad para manejar un aumento en el número de usuarios y transacciones sin degradar el rendimiento. |
| Gestión de servicios adicionales (catering, decoración). | Mantenibilidad: Facilidad para realizar actualizaciones y mantenimiento del sistema. |

Decisiones y Prioridades:

**Eficiencia vs. Portabilidad:** La prioridad se ha dado a la eficiencia para asegurar que las operaciones críticas del hotel se realicen de manera rápida y fiable, mientras que la portabilidad se ha asegurado mediante el desarrollo de aplicaciones web y móviles.

**Seguridad vs. Usabilidad:** Se ha dado prioridad a la seguridad para proteger los datos de los clientes, asegurando al mismo tiempo que la interfaz de usuario sea intuitiva y fácil de usar.

**Escalabilidad:** El sistema está diseñado para escalar horizontalmente, permitiendo la adición de más servidores según sea necesario para manejar cargas de trabajo mayores.

## Objetivos del diseño

<<Enumeración de los objetivos de diseño del sistema. Algunos corresponderán con los requisitos no funcionales establecidos en la fase de requisitos y otros con los requisitos generales de calidad del diseño. Se justificarán las decisiones tomadas en la elección entre objetivos deseables pero incompatibles, así como las prioridades establecidas en el diseño e implementación del sistema.>>

Objetivos Generales:

* *Optimización de la Gestión Operativa*: Automatizar y agilizar la gestión de reservas de habitaciones y eventos para reducir errores y mejorar la eficiencia.
* *Mejora de la Experiencia del Cliente:* Proveer una interfaz de usuario intuitiva y accesible desde múltiples dispositivos, facilitando el proceso de reserva y gestión de eventos.
* *Integración con Plataformas Externas:* Facilitar la sincronización con plataformas de reservas externas para maximizar la ocupación de habitaciones y mejorar la gestión de disponibilidad y tarifas.

Requisitos No Funcionales Relacionados:

Eficiencia: El sistema debe manejar consultas y transacciones en menos de 2 segundos para garantizar una experiencia de usuario fluida.

Portabilidad: La interfaz de usuario debe ser accesible desde navegadores web y aplicaciones móviles en plataformas iOS y Android.

Seguridad: Implementar autenticación y autorización robustas, cifrado de datos sensibles y cumplimiento con regulaciones de privacidad de datos.

Escalabilidad: La arquitectura del sistema debe permitir el escalamiento horizontal para manejar incrementos en el volumen de usuarios y transacciones.

Mantenibilidad: El código debe ser modular y documentado, facilitando la implementación de actualizaciones y nuevas funcionalidades sin afectar la estabilidad del sistema.

**Justificación de Decisiones y Prioridades:**

**Eficiencia vs. Portabilidad:** Se ha priorizado la eficiencia para asegurar operaciones rápidas y fiables, implementando tecnologías y prácticas de desarrollo que también permitan la portabilidad del sistema.

**Seguridad vs. Usabilidad:** Aunque la seguridad es crítica, se ha diseñado una interfaz de usuario que sea fácil de navegar y utilizar, asegurando que las medidas de seguridad no interfieran con la usabilidad.

**Escalabilidad y Mantenibilidad**: La arquitectura modular del sistema facilita el mantenimiento y la expansión, asegurando que el sistema pueda crecer con las necesidades del hotel sin comprometer el rendimiento o la estabilidad.

Este enfoque asegura que el diseño del sistema no solo cumpla con los requisitos funcionales y no funcionales establecidos, sino que también se alinee con los objetivos estratégicos del "Hotel Chachapa" para ofrecer un servicio excepcional a sus clientes y mejorar la eficiencia operativa del hotel.

## Definiciones, estereotipos UML y abreviaturas

<<Este apartado proporciona las definiciones de todos los términos, acrónimos y abreviaturas utilizadas a lo largo del documento y que permiten una interpretación correcta del mismo. Se han de incluir los términos técnicos, caso de uso por ejemplo, y los específicos del entorno del sistema, lector de bandas por ejemplo. Es conveniente ordenarlos alfabéticamente>>

Este apartado proporciona las definiciones de todos los términos, acrónimos y abreviaturas utilizadas a lo largo del documento, facilitando una interpretación correcta del mismo. Se incluyen tanto los términos técnicos como los específicos del entorno del sistema. Los términos se presentan en orden alfabético y, en ocasiones, agrupados por categorías para una mejor comprensión.

Términos y Abreviaturas

API (Application Programming Interface): Conjunto de funciones y procedimientos que permiten la creación de aplicaciones que acceden a las características o datos de un sistema operativo, aplicación u otro servicio.

Base de Datos: Colección organizada de datos que puede ser fácilmente accesada, gestionada y actualizada.

CRUD (Create, Read, Update, Delete): Operaciones básicas de almacenamiento persistente.

Eficiencia: Medida de la capacidad del sistema para procesar y responder a las solicitudes de los usuarios de manera rápida.

Escalabilidad: Capacidad del sistema para manejar un aumento en la carga de trabajo, añadiendo recursos adicionales.

Pasarela de Pago: Servicio que automatiza el proceso de pago entre el cliente y el comerciante.

Portabilidad: Capacidad del software para ser utilizado en diferentes entornos y plataformas.

RUP (Rational Unified Process): Metodología de desarrollo de software que enfatiza un enfoque iterativo y centrado en el cliente.

Seguridad: Medidas y controles implementados para proteger los datos y la funcionalidad del sistema contra accesos no autorizados y vulnerabilidades.

SIGHE: Acrónimo de Sistema Integral de Gestión Hotelera y de Eventos.

Estereotipos UML

Actor: Representa un rol jugado por un usuario o cualquier otro sistema que interactúa con el sistema bajo desarrollo.

Clase: Representa una entidad con atributos y métodos en el sistema.

Caso de Uso: Descripción de un conjunto de interacciones entre un actor y el sistema para lograr un objetivo específico.

Diagrama de Clases: Representación estática de las clases en un sistema y sus relaciones.

Diagrama de Secuencia: Representación dinámica que muestra cómo los objetos interactúan en un escenario particular de un caso de uso.

Entidad: Objeto del mundo real o conceptual que tiene una existencia independiente y es relevante para el sistema.

<<La definición del <término> se incluye a continuación. Se debe incorporar tanta información como el lector necesite para comprender el concepto presentado>>

<<En ocasiones es conveniente organizar los términos en grupos para facilitar su lectura. Por ejemplo si el dominio del problema incluye términos relacionados con aspectos de tipo técnico y de gestión se pueden agrupar los términos para evitar confusión>>

Grupos de Términos

Términos Técnicos:

API (Application Programming Interface)

CRUD (Create, Read, Update, Delete)

Pasarela de Pago

Base de Datos

Términos del Sistema:

Eficiencia

Escalabilidad

Portabilidad

Seguridad

Términos de Metodología:

RUP (Rational Unified Process)

Actor

Clase

Caso de Uso

Diagrama de Clases

Diagrama de Secuencia

Entidad

Definiciones Detalladas

API (Application Programming Interface):

Un conjunto de herramientas y protocolos utilizados para construir software y aplicaciones. Permite que diferentes aplicaciones se comuniquen entre sí.

Base de Datos:

Sistema organizado de datos estructurados que permite el almacenamiento, gestión y recuperación de información de manera eficiente. Ejemplos incluyen MySQL, PostgreSQL y MongoDB.

CRUD (Create, Read, Update, Delete):

Conjunto de operaciones básicas necesarias para trabajar con una base de datos u otro sistema de almacenamiento de datos.

Eficiencia:

La capacidad del sistema para utilizar sus recursos de manera óptima para proporcionar respuestas rápidas y realizar tareas de manera efectiva.

Escalabilidad:

La propiedad de un sistema que le permite crecer y manejar una mayor carga de trabajo agregando recursos adicionales sin comprometer el rendimiento.

Pasarela de Pago:

Un servicio que autoriza pagos realizados a través de tarjetas de crédito, débito y otras formas de pago electrónico, y actúa como intermediario entre el vendedor y el comprador.

Portabilidad:

La capacidad del software para funcionar en diferentes entornos de hardware o sistemas operativos con pocas o ninguna modificación.

RUP (Rational Unified Process):

Metodología de desarrollo de software que utiliza un enfoque iterativo y está centrada en la arquitectura. Fomenta la participación del cliente y la entrega de productos incrementales.

Seguridad:

Conjunto de medidas diseñadas para proteger un sistema contra amenazas tales como el acceso no autorizado, el robo de datos, y otras vulnerabilidades.

SIGHE:

Acrónimo de Sistema Integral de Gestión Hotelera y de Eventos, el software desarrollado para el “Hotel Chachapa” que optimiza la gestión de reservas de habitaciones y eventos.

Actor (UML):

Representa cualquier entidad externa que interactúa con el sistema, puede ser un usuario humano, otro sistema o un dispositivo.

Clase (UML):

Una plantilla para crear objetos, proporcionando valores iniciales para el estado de los objetos y las implementaciones de comportamiento.

Caso de Uso (UML):

Descripción de un conjunto de acciones que un sistema realiza para producir un resultado observable de valor para un actor en particular.

Diagrama de Clases (UML):

Representación estática de las clases del sistema, sus atributos, métodos y las relaciones entre ellas.

Diagrama de Secuencia (UML):

Diagrama que muestra cómo los objetos interactúan en una secuencia de mensajes en un escenario de un caso de uso específico.

Entidad:

Algo que tiene una existencia independiente en el sistema, puede ser un objeto físico o una abstracción como un evento o un concepto.

Este apartado asegura que todos los términos y conceptos utilizados a lo largo del documento son claros y comprensibles para el lector, facilitando una mejor comprensión del diseño y funcionalidad del sistema propuesto.

## Referencias

<<Referencias a otros documentos e información de trazabilidad – documento/s de análisis de requisitos relacionados, sistemas existentes, restricciones que afectan a la arquitectura software, etc .>>

# Arquitectura del sistema.

## Arquitectura propuesta

<<Documenta el modelo de diseño del sistema para el nuevo sistema.>>

Definir el estilo arquitectónico de acuerdo a la metodología.

Diseña un diagrama de actividades que modele el sistema completo

## Visión global.

<<Presenta la estructura global de la arquitectura del sistema y una breve descripción de la asignación de funcionalidad de cada subsistema. Descripción de la arquitectura software elegida para el sistema (patrón arquitectónico)>>

## Diseño de la Arquitectura

<<En este apartado se establece la descomposición en subsistemas y las responsabilidades asignadas a cada uno. Corresponde a los productos obtenidos de las fases de análisis y diseño arquitectónico. En la sección de topología del sistema se describirán la asignación del software al hardware – diagrama de despliegue- y a los componentes – diagrama de componentes. La sección de “Gestión de la Persistencia” describe la infraestructura de almacenamiento requerida y los datos que serán almacenados por el sistema. La sección de “Aspectos Globales y de Seguridad” se describen las políticas de seguridad y acceso que serán establecidas en el sistema – mecanismos de autentificación, encriptación, certificados, etc.- y los aspectos de sincronización, concurrencia e inicialización de subsistemas. Por último, la sección de “Aspectos de rendimiento y tamaño” establecerá los acuerdos de nivel de servicio establecidos con el usuario en relación a tiempos de respuesta aceptables, productividad y capacidad de almacenamiento requerido.>>

## Diseño de los subsistemas

<<En esta sección se describirán los servicios proporcionados por cada subsistema. Se establecen los límites entre los subsistemas y se especifican sus interfaces.>>

## Vista de casos de uso

<<En esta sección se enumeran los casos de uso incluidos en el modelo de casos de uso que representan las partes de funcionalidad central y que son significativos para el sistema final o que cubren una parte importante de la arquitectura del sistema final.

La documentación a incluir en esta sección corresponde a la obtenida como consecuencia de la actividad “Realización de casos de uso”:

## Vista lógica

<<Esta sección describe las partes arquitectónicamente significativas del modelo de diseño, la descomposición de los subsistemas en paquetes por ejemplo. Para cada paquete significativo, su descomposición en clases. Se deberán incluir las clases relevantes para la arquitectura y describir sus responsabilidades así como sus relaciones, atributos y operaciones.

Para cada paquete relevante, incluir una subsección con el nombre del paquete, una breve descripción y un diagrama de con todas las clases y paquetes contenidas en dicho paquete.>>

# Diseño detallado

## Introducción.

<<Se describen las elecciones de diseño específicas: tipo de desarrollo - adquisición o desarrollo –, rendimiento – espacio en memoria vs tiempo de respuesta – etc. Se incluyen los estándares de desarrollo a seguir – asignación de nombres, mecanismos de gestión de excepciones – los estándares de interfaces y los de codificación que constituyen la guía de desarrollo para los diferentes diseñadores e implementadores.>>

## Diseño de casos de uso reales

<<En esta sección se enumeran los casos de uso incluidos en el modelo de casos de uso que representan las partes de funcionalidad central y que son significativos para el sistema final o que cubren una parte importante de la arquitectura del sistema final.

La documentación a incluir en esta sección corresponde a la obtenida como consecuencia de la actividad “Realización de casos de uso”:

Flujos de eventos(Plantillas de caso des uso) - Diseño: descripción textual de cómo se realiza el caso de uso en términos de los objetos que colaboran. Resumen de los diagramas conectados con el caso de uso y explicación de sus relaciones.

Diagramas de interacción: Diagramas de secuencia ó Diagramas de colaboración, objetos participantes, Diagramas de clases.

Requisitos derivados: Descripción textual que recoge todos los requisitos, normalmente los no funcionales, de la realización del caso de uso no que han de tenerse en cuenta durante la implementación>>

**Tabla 1. Formatos de casos de uso (Tomar como referencia el formato propuesto en clase)**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | < Escriba aquí el nombre del caso de uso [CU número del caso de uso]> |
| Descripción | <Breve descripción de lo que hace el Caso de uso, el objetico que logra> |
| Actores Principales | <Nombre del Rol>:<Descripción del Rol del  actor que esta interactuando con el sistema> |
| Actores Secundarios | <Aquí se puede incluir, por ejemplo, a los actores de soporte> |
| Precondiciones | <Lo que debe cumplirse para ejecutar el caso de uso> |
| Requerimientos. especiales | Reglas de Negocio: <son las especificaciones de los procesos que maneja el usuario> |
| Formatos (de captura) asociados al Caso: <se |

|  |  |
| --- | --- |
|  | debe especificar el número de la forma1 a la que se hace referencia > |
| Formatos de “Salida” asociados al Caso: <la relación de las formas de salida del sistema en este caso de uso> |
| Poscondiciones | <Lo que debe cumplirse una vez finalizado el caso de uso con éxito> |
| Requerimientos no funcionales | De Comunicación: <Detalle aquí los requerimientos no funcionales de comunicaciones, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De Persistencia: <Detalle aquí los requerimientos no funcionales de bases de datos, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De Seguridad: <Detalle aquí los requerimientos no funcionales de seguridad, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De Escalabilidad: <Detalle aquí los requerimientos no funcionales de escalabilidad, que hacen que el caso de prueba funcione con éxito o no aplica (N/A) si es el caso>  De rendimiento: <Detalle aquí los requerimientos no funcionales de rendimiento, que hacen que el caso de prueba funcione con  éxito o no aplica (N/A) si es el caso> |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| <Paso No.> | <Cuál es la condición que desencadena la extensión o variante> | <Descripción de lo que ocurre en la variación o extensión> |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| <Paso No.> | <Cuál es la condición que desencadena la extensión o  variante> | <Descripción de lo que ocurre en la variación o extensión> |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| <Id Punto> | <Descripción de las dudas o indefiniciones existentes en el caso de uso> |

1 Forma: se refiere a la especificación de los campos que hacen parte de ella. (referirse al documento formato de formas)

## Diseño detallado de las clases.

<<Describe completamente las clases (nombre, atributos y métodos) CRC y sus interfaces. Se incluye la documentación de las dependencias de cada clase con otras clases y paquetes.>>

# Diseño de la arquitectura de módulos del sistema

<<Describe completamente los módulos y submódulos que componen el sistema, con la definición de cada uno de ellos. Puede acompañarse de un diagrama con el cual se puedan ver y la relación entre ellos.>>

# Diseño físico de datos

# Modelo de datos y diccionario de datos

<<En esta sección del documento se agrega el diagrama clases con el diagrama de entidades de la base de datos. Adicionalmente se deberá complementar el diagrama entidad relación con la respectiva descripción detallada de cada una de las entidades y atributos como se muestra a continuación..>>

## Modelo Entidad – Relación. (Diagrama de Clases)

<< En esta sección se deberá incluir el diagrama entidad relación desarrollado.>>

## Diccionario de datos.

**Tabla 2 Diccionario de datos**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - <#> | < Escriba aquí el nombre de la entidad (Tabla) > | | | | | |
| Descripción | | < Escriba aquí la descripción de la nombre de la entidad (Tabla) > | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| <#> | < Escriba aquí | < Escriba aquí la | | < | < | < | < | < |
|  | el nombre de | descripción de la | | Escriba | Marque | Indiqu | Indique | Marqu |
|  | la columna > | columna > | | aquí el | SI o NO | e PK – | aquí | e SI o |
|  |  |  | | tipo de | permite | Para | cual es | NO es |
|  |  |  | | datos de | valores | llaves | el valor | un |
|  |  |  | | la | NULL > | primari | predeter | valor |
|  |  |  | | columna |  | as o | minado | auto |
|  |  |  | | > |  | FK – | si aplica | numér |
|  |  |  | |  |  | para | > | ico > |
|  |  |  | |  |  | llaves |  |  |
|  |  |  | |  |  | secun |  |  |
|  |  |  | |  |  | darias |  |  |
|  |  |  | |  |  | > |  |  |

# Prototipo de pantallas

<<En esta sección se incluirá la descripción de navegación a nivel de interfaz de usuario (Diagrama de navegación entre pantallas), se deberá mostrar el prototipo de pantallas del mismo. Se podrá usar algún software de prototipos previamente pactado con la SDIS. Se incluirá la propuesta de diseño de interfaz de usuario con los colores y logos definidos conjuntamente con la SDIS y acorde con la imagen corporativa y las disposiciones de la Comisión Distrital de Sistemas y la normatividad vigente.>>

Realiza un mapa de navegación y posteriormente agrega las imágenes del prototipo elaborado en Figma.