# DOCUMENTO DE DISEÑO

Hotel y eventos.

Ingeniería en Software

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 | Primera redacción de las generalidades. |
| 16/05/2024 | Zepeda Minero Manuel | 1.0 | Redacción del punto 6 |
| 16/05/2024 | López Pérez Diego Alexis | 1.0 | Primera redacción del Diseño detallado |
| 16/05/2024 | Carro Gutierrez Jesús | 1.0 | Complementación de punto 2 y redacción del punto 4 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLA DE CONTENIDO**

Contenido

[DOCUMENTO DE DISEÑO 1](#_Toc166809452)

[REVISIÓN Y CONTROL DE CAMBIOS 2](#_Toc166809453)

[Tabla de Imágenes 4](#_Toc166809454)

[Resumen 5](#_Toc166809455)

[1. Generalidades 6](#_Toc166809456)

[1.1. Propósito del sistema. 6](#_Toc166809457)

[1.2. Objetivos del diseño 7](#_Toc166809458)

[1.3. Definiciones, estereotipos UML y abreviaturas 8](#_Toc166809459)

[1.4. Referencias 10](#_Toc166809460)

[2. Arquitectura del sistema. 11](#_Toc166809461)

[2.1. Arquitectura propuesta 11](#_Toc166809462)

[2.1.1. Visión global. 12](#_Toc166809463)

[2.1.2. Diseño de la Arquitectura 14](#_Toc166809464)

[2.1.3. Diseño de los subsistemas 18](#_Toc166809465)

[2.1.4. Vista de casos de uso 20](#_Toc166809466)

[2.1.4.1. Vista lógica 21](#_Toc166809467)

[3. Diseño detallado 23](#_Toc166809468)

[3.1. Introducción. 23](#_Toc166809469)

[3.2. Diseño de casos de uso reales 24](#_Toc166809470)

[3.3. Diseño detallado de las clases. 37](#_Toc166809471)

[4. Diseño de la arquitectura de módulos del sistema 39](#_Toc166809472)

[￼Diccionario de datos. 41](#_Toc166809473)

[5. Diseño físico de datos 42](#_Toc166809474)

[6. Modelo de datos y diccionario de datos 42](#_Toc166809475)

[6.1. Modelo Entidad – Relación. (Diagrama de Clases) 42](#_Toc166809476)

[6.2. Diccionario de datos. 44](#_Toc166809477)

[Diccionario de datos. 44](#_Toc166809478)

[7. Prototipo de pantallas 45](#_Toc166809479)

[7.1. Diccionario de datos. 47](#_Toc166809480)

# Tabla de Imágenes

[Ilustración 1 Diagrama de Actividades 12](#_Toc166809342)

[Ilustración 2 Nodos de Hardware 15](#_Toc166809343)

[Ilustración 3 16](#_Toc166809344)

[Ilustración 4 Despliegue 17](#_Toc166809345)

[Ilustración 5 Diagrama de Casos de Uso 36](#_Toc166809346)

[Ilustración 6 Diagrama de Clases 43](#_Toc166809347)

[Ilustración 7 45](#_Toc166809348)

[Ilustración 8 46](#_Toc166809349)

[Ilustración 9 47](#_Toc166809350)

[Ilustración 10 48](#_Toc166809351)

[Ilustración 11 49](#_Toc166809352)

[Ilustración 12 49](#_Toc166809353)

[Ilustración 13 50](#_Toc166809354)

[Ilustración 14 50](#_Toc166809355)

[Ilustración 15 51](#_Toc166809356)

# 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 .>>

Este apartado proporciona referencias a otros documentos e información de trazabilidad relevante para el diseño del Sistema Integral de Gestión Hotelera y de Eventos (SIGHE) para el “Hotel Chachapa”. Estas referencias son esenciales para comprender el contexto completo del proyecto, sus requisitos, restricciones y decisiones arquitectónicas.

Documentación Referenciada:

Acta de Constitución del Proyecto:

* Descripción: Documento que define el planteamiento principal del proyecto, la metodología a seguir, los roles asignados, y la administración del proyecto.
* Contenido: Metodología RUP, roles y responsabilidades, objetivos del proyecto, riesgos identificados, y plan de gestión.
* Ubicación: Repositorio GitHub del proyecto ([link al documento](https://github.com/ManuelZep/Proyecto-Ing-de-Software)).

Documento de Requisitos:

* Descripción: Documento que especifica los requisitos funcionales y no funcionales del sistema, detallando las funcionalidades esperadas y las restricciones técnicas.
* Contenido: Requisitos de gestión de reservas, gestión de eventos, facturación y pagos, integración con plataformas externas, y otros.
* Ubicación: Repositorio GitHub del proyecto ([link al documento](https://github.com/ManuelZep/Proyecto-Ing-de-Software)).

Prototipo:

* Descripción: Documento que documenta y explica el entregable del prototipo, incluyendo los diseños preliminares de la interfaz de usuario y la interacción del usuario con el sistema.
* Contenido: Bocetos y wireframes del sistema, descripción de la navegación y funcionalidad básica.
* Ubicación: Repositorio GitHub del proyecto ([link prototipo](https://www.figma.com/design/IxAAW4MMWKokvlCo4P4nFp/Untitled?node-id=0-1&t=3Vb4UK37VWjMLEMZ-0)).

Diagrama de Clases:

* Descripción: Documento en formato PDF que contiene el diagrama de clases UML, mostrando la estructura del sistema, las relaciones entre las clases, y los principales atributos y métodos.
* Contenido: Diagrama UML de clases, descripciones de las clases, atributos y métodos, y sus relaciones.
* Ubicación: Repositorio GitHub del proyecto ([link al diagrama](https://lucid.app/lucidchart/bea5f4c3-4425-4da5-88f7-8575a3a45488/edit?viewport_loc=40%2C300%2C6492%2C3328%2CHWEp-vi-RSFO&invitationId=inv_a5a03939-476b-432d-b75c-8c10a82e9c02)).

Presente Documento:

* Descripción: Este documento proporciona un diseño detallado de la arquitectura del sistema, incluyendo la descripción de los módulos, la interacción entre componentes y el modelo de diseño.
* Contenido: Descripción de la arquitectura propuesta, diagramas de actividades, justificación de decisiones de diseño.
* Ubicación: Repositorio GitHub del proyecto ([link al documento](https://github.com/ManuelZep/Proyecto-Ing-de-Software)).

Sistemas Existentes y Restricciones:

* Sistemas Existentes:

Referencia a los sistemas actualmente en uso en el “Hotel Chachapa” que el nuevo sistema deberá reemplazar o integrar.

Descripción de las interfaces y datos que se deben migrar o integrar con el nuevo sistema.

* Restricciones Técnicas:

Descripción: Restricciones que afectan la arquitectura del software, tales como limitaciones de hardware, software específico, estándares de seguridad, y políticas de la empresa.

Ejemplo: Restricción de uso de determinadas tecnologías debido a compatibilidad con sistemas heredados, requisitos de rendimiento, y políticas de seguridad de datos.

Estas referencias proporcionan un marco integral para el proyecto, asegurando que todas las partes interesadas y desarrolladores puedan acceder a la información necesaria para comprender y construir el Sistema Integral de Gestión Hotelera y de Eventos de manera eficiente y coordinada.

# Arquitectura del sistema.

## Arquitectura propuesta

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

Definir el estilo arquitectónico de acuerdo con la metodología.

Diseña un diagrama de actividades que modele el sistema completo

Hemos elegido la arquitectura Cliente-Servidor para diseñar cada módulo de la arquitectura del sistema, esta arquitectura nos es tanto útil para desarrollar el software en una página web como para poder implementar la metodología RUP.

La arquitectura Cliente-Servidor es compatible con la metodología RUP, que se basa en un enfoque iterativo e incremental. RUP divide el desarrollo en cuatro fases principales: Incepción, Elaboración, Construcción y Transición. Cada módulo del sistema puede ser desarrollado y refinado a lo largo de estas fases, permitiendo entregas incrementales y la incorporación de retroalimentación continua.

**Estilo Arquitectónico**

Esta arquitectura se divide en dos componentes principales:

* ***Cliente:*** La interfaz de usuario (UI) que interactúa con los usuarios del sistema (clientes y personal del hotel) a través de aplicaciones web y móviles.
* ***Servidor:*** Gestiona la lógica de negocio, procesa solicitudes y administra la base de datos del sistema.

**Ventajas de la Arquitectura Cliente-Servidor:**

*Escalabilidad:* Permite añadir más clientes o servidores según la demanda.

*Mantenibilidad:* Facilita la actualización y mantenimiento del sistema.

*Seguridad:* Mejora la protección de datos al centralizar la lógica de negocio y el acceso a la base de datos.

*Eficiencia:* Optimiza el procesamiento de solicitudes al distribuir la carga entre cliente y servidor.

**Diagrama de Actividades**

Para modelar el sistema completo, se utiliza un diagrama de actividades que representa el flujo de operaciones principales del SIGHE. A continuación, se presenta un esquema básico de este diagrama:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza baja

Ilustración 1 Diagrama de Actividades

Este diagrama de actividades ilustra cómo las solicitudes del cliente son recibidas y procesadas por el servidor, y cómo diferentes módulos del sistema interactúan para completar las operaciones solicitadas.

La combinación de la arquitectura Cliente-Servidor y la metodología RUP garantiza un desarrollo estructurado y eficiente del Sistema Integral de Gestión Hotelera y de Eventos, cumpliendo con los requisitos funcionales y no funcionales del “Hotel Chachapa” y asegurando una solución robusta y escalable.

## 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)>>

La arquitectura del SIGHE se diseñó sobre una estructura Cliente-Servidor dividida en dos componentes principales: Cliente y Servidor.

Cliente: Esta componente incluye la interfaz de usuario (UI) que interactúa con los usuarios del sistema, tanto clientes como personal del hotel. Las aplicaciones web y móviles permiten a los usuarios realizar diversas operaciones como reservas, consultas de disponibilidad, check-ins y check-outs, gestión de eventos, entre otros.

Servidor: Este componente gestiona la lógica de negocio, procesa las solicitudes provenientes del cliente y administra la base de datos. Se encarga de operaciones como la validación de datos, la ejecución de procesos de negocio, la actualización de registros y la generación de informes. Además, proporciona una capa de seguridad centralizada que protege los datos sensibles y garantiza el acceso controlado a la información.

**Asignación de Funcionalidad de cada Subsistema**

La arquitectura del sistema se divide en varios subsistemas, cada uno con responsabilidades específicas:

**Subsistema de Gestión de Reservas:**

Funcionalidad: Gestión de reservas de habitaciones y eventos, incluyendo la disponibilidad, confirmación, modificación y cancelación de reservas.

Interfaz: Formularios web y móviles para reservas.

**Subsistema de Gestión de Clientes:**

Funcionalidad: Administración de perfiles de clientes, incluyendo registro, actualización de información y seguimiento de historial.

Interfaz: Formularios de registro y perfiles de usuario.

**Subsistema de Gestión de Eventos:**

Funcionalidad: Organización y administración de eventos, desde la planificación hasta la ejecución, incluyendo la gestión de recursos y logística.

Interfaz: Paneles de control para la planificación y gestión de eventos.

**Subsistema de Facturación y Pagos:**

Funcionalidad: Procesamiento de facturas, gestión de pagos y generación de informes financieros.

Interfaz: Formatos de facturación y módulos de pago en línea.

**Subsistema de Mantenimiento y Soporte:**

Funcionalidad: Gestión de solicitudes de mantenimiento, asignación de tareas y seguimiento de incidencias.

Interfaz: Panel de control para la administración de tareas y solicitudes.

**Patrón Arquitectónico**

El patrón arquitectónico Cliente-Servidor es elegido por su capacidad para soportar un enfoque iterativo e incremental, como el que promueve la metodología RUP. Esta arquitectura facilita la escalabilidad, mantenibilidad, seguridad y eficiencia del sistema.

**Beneficios de la Arquitectura Cliente-Servidor en RUP**

Escalabilidad: La capacidad de añadir nuevos clientes y servidores según la demanda garantiza que el sistema pueda crecer junto con las necesidades del hotel.

Mantenibilidad: La separación de la lógica de negocio en el servidor facilita las actualizaciones y el mantenimiento del sistema sin afectar a los usuarios finales.

Seguridad: La centralización de la lógica de negocio y el acceso a la base de datos mejora la protección de datos, asegurando que solo usuarios autorizados accedan a información sensible.

Eficiencia: La distribución de la carga de trabajo entre cliente y servidor optimiza el procesamiento de solicitudes, mejorando la experiencia del usuario.

En resumen, la combinación de la arquitectura Cliente-Servidor con la metodología RUP proporciona una base sólida y estructurada para el desarrollo del SIGHE, asegurando una solución robusta, segura y adaptable a las necesidades del "Hotel Chachapa". Esta visión global sienta las bases para un sistema integral y eficiente, capaz de evolucionar y mejorar con el tiempo, respondiendo adecuadamente a las expectativas y requerimientos de todos sus usuarios.

## 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.>>  
   
  
  
- Subsistema de Gestión de Reservas:

Responsabilidades: Manejo de reservas de habitaciones y eventos, verificación de disponibilidad, confirmación, modificación y cancelación de reservas.

Interfaces: Formularios web y móviles para realizar reservas.

* Subsistema de Gestión de Clientes:

Responsabilidades: Administración de perfiles de clientes, registro de nuevos clientes y seguimiento del historial de clientes.

Interfaces: Páginas de sign up y administración de perfiles.

* Subsistema de Gestión de Eventos (Calendario):

Responsabilidades: Organización y administración de eventos, desde la planificación hasta la ejecución, incluyendo la gestión de recursos y logística.

Interfaces: Paneles de control y herramientas de planificación para el personal del hotel.

* Subsistema de Facturación y Pagos:

Responsabilidades: Procesamiento de facturas, gestión de pagos, gestión de pagos con intermediarios (ej. Paypal) y administración de transacciones.

Interfaces: Módulo de facturación y sistemas de pago en línea.

* Subsistema de Mantenimiento y Soporte:

Responsabilidades: Gestión de solicitudes de mantenimiento, asignación de tareas y seguimiento de incidencias.

Interfaces: Panel de control para la administración de tareas y solicitudes de mantenimiento.

1. Nodos de hardware

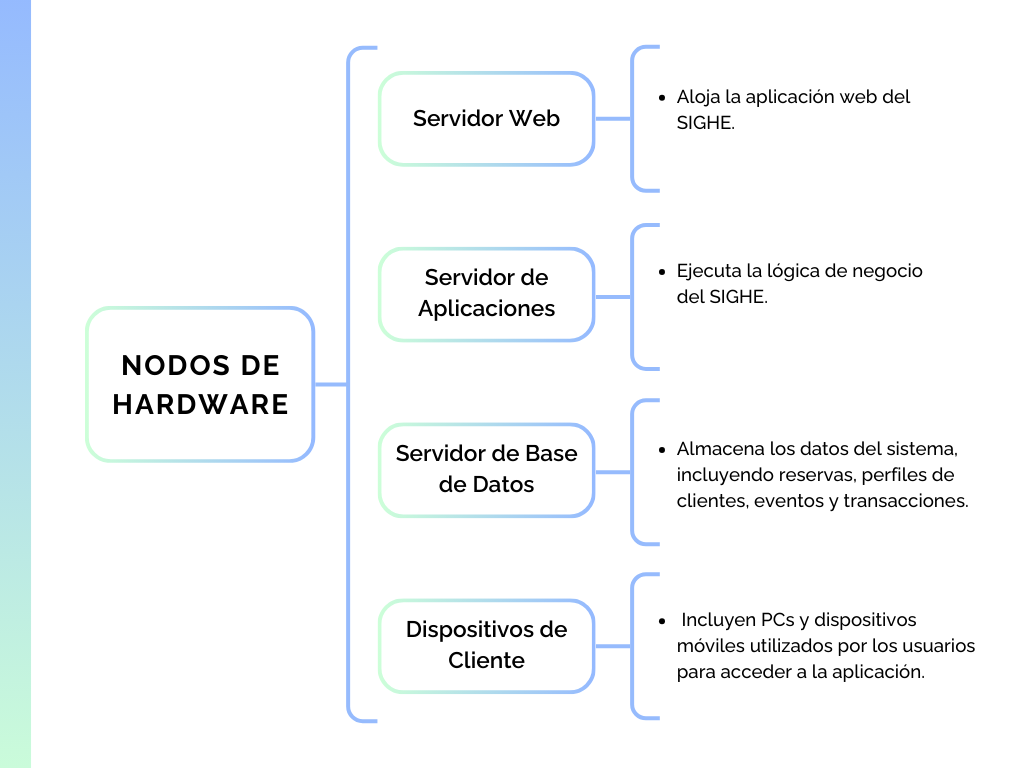


Ilustración 2 Nodos de Hardware

1. Asignación de hardware:

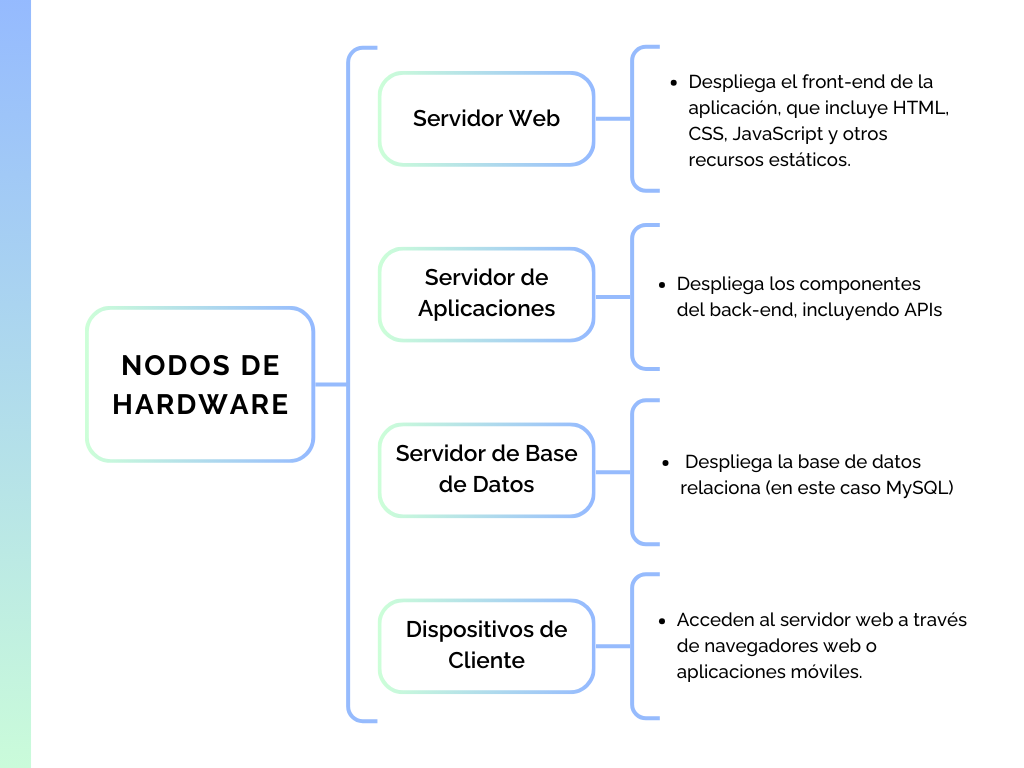


Ilustración 3

1. Diagrama de despliegue:

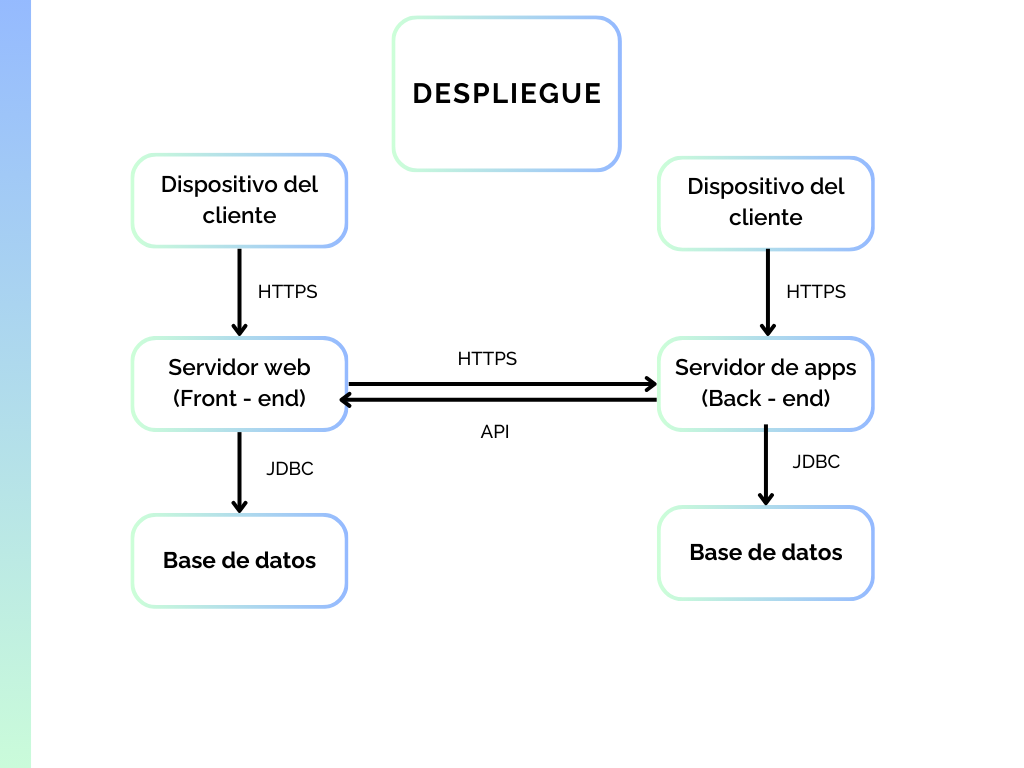


Ilustración 4 Despliegue

Componentes Principales:

Interfaz de Usuario (UI): Formularios web y móviles que interactúan con los usuarios.

Controladores: Gestionan las solicitudes del usuario y las redirigen a los servicios apropiados.

Servicios: Contienen la lógica de negocio del sistema.

Repositorios: Gestionan el acceso a la base de datos.

Modelos: Representan las entidades y datos del sistema.

Interacciones:

UI <-> Controladores: La interfaz de usuario envía solicitudes a los controladores.

Controladores <-> Servicios: Los controladores llaman a los servicios para procesar la lógica de negocio.

Servicios <-> Repositorios: Los servicios interactúan con los repositorios para acceder a los datos.

Repositorios <-> Base de Datos: Los repositorios realizan operaciones de lectura/escritura en la base de datos.

Descripción del Diagrama de Componentes

Interfaz de Usuario:

Formularios web y móviles.

Muestra la información al usuario y captura sus entradas.

Controladores:

Gestionan las solicitudes HTTP/HTTPS.

Dirigen las solicitudes a los servicios correspondientes.

Servicios:

Contienen la lógica de negocio.

Ejecutan operaciones específicas del sistema como gestión de reservas, clientes y eventos.

Repositorios

Interactúan con la base de datos.

Ejecutan consultas y actualizaciones de datos.

Modelos:

Representan las entidades del sistema (e.g., clientes, reservas, eventos).

Utilizados por los servicios y repositorios para manejar los datos.

## 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.>>

Subsistema de Gestión de Reservas

Servicios Proporcionados:

Consulta de Disponibilidad: Permite a los usuarios verificar la disponibilidad de habitaciones y salas de eventos en tiempo real.

Reserva de Habitaciones: Facilita la reserva de habitaciones, permitiendo seleccionar fechas, tipos de habitación y realizar la confirmación.

Modificación de Reservas: Permite a los usuarios modificar sus reservas existentes, como cambiar fechas o tipos de habitación.

Cancelación de Reservas: Gestiona la cancelación de reservas y actualiza el sistema en consecuencia.

Límites e Interfaces:

Interfaces: Formularios web y aplicaciones móviles para gestionar las reservas.

Interacción con Otros Subsistemas:

Gestión de Clientes: Verifica la información del cliente.

Facturación y Pagos: Procesa los pagos relacionados con las reservas.

Gestión de Eventos: Coordina la disponibilidad de salas para eventos.

Subsistema de Gestión de Clientes

Servicios Proporcionados:

Registro de Clientes: Permite a los nuevos clientes registrarse en el sistema.

Actualización de Información: Facilita la actualización de los datos personales de los clientes.

Consulta de Historial: Proporciona acceso al historial de reservas y transacciones de cada cliente.

Gestión de Perfiles: Administra los perfiles de los clientes, incluyendo preferencias y detalles de contacto.

Límites e Interfaces:

Interfaces: Páginas de registro y administración de perfiles accesibles desde aplicaciones web y móviles.

Interacción con Otros Subsistemas:

Gestión de Reservas: Proporciona datos del cliente.

Facturación y Pagos: Accede a la información de transacciones vinculadas a los clientes.

Subsistema de Gestión de Eventos

Servicios Proporcionados:

Planificación de Eventos: Facilita la creación y planificación de eventos, incluyendo la asignación de recursos y salas.

Administración de Recursos: Gestiona los recursos necesarios para los eventos, como equipos, personal y catering.

Seguimiento de Eventos: Permite el seguimiento del estado de los eventos, desde la planificación hasta la ejecución.

Gestión de Invitados: Administra la lista de invitados y envía invitaciones y recordatorios.

Límites e Interfaces:

Interfaces: Paneles de control y herramientas de planificación para el personal del hotel.

Interacción con Otros Subsistemas:

Gestión de Reservas: Coordina la disponibilidad de salas.

Facturación y Pagos: Gestiona los pagos relacionados con los eventos.

Mantenimiento y Soporte: Asegura que los eventos no se vean afectados por problemas de mantenimiento.

Subsistema de Facturación y Pagos

Servicios Proporcionados:

Generación de Facturas: Crea facturas detalladas para reservas y eventos.

Procesamiento de Pagos: Gestiona los pagos realizados por los clientes, incluyendo diferentes métodos de pago.

Emisión de Recibos: Proporciona recibos electrónicos a los clientes tras la confirmación de pago.

Informes Financieros: Genera informes financieros para la administración del hotel.

Límites e Interfaces:

Interfaces: Módulos de facturación y sistemas de pago en línea accesibles desde aplicaciones web y móviles.

Interacción con Otros Subsistemas:

Gestión de Reservas: Facturación de servicios de reserva.

Gestión de Eventos: Facturación de servicios de eventos.

Gestión de Clientes: Vincula pagos a perfiles de clientes.

Subsistema de Mantenimiento y Soporte

Servicios Proporcionados:

Gestión de Solicitudes: Administra las solicitudes de mantenimiento y soporte enviadas por el personal y los clientes.

Asignación de Tareas: Asigna tareas de mantenimiento al personal adecuado y rastrea su progreso.

Seguimiento de Incidencias: Permite el seguimiento de incidencias reportadas y su resolución.

Historial de Mantenimiento: Mantiene un registro de todas las actividades de mantenimiento realizadas.

Límites e Interfaces:

Interfaces: Panel de control para la administración de tareas y solicitudes accesible desde aplicaciones web y móviles.

Interacción con Otros Subsistemas:

Gestión de Clientes: Recibe solicitudes de soporte.

Gestión de Eventos: Coordina el mantenimiento para asegurar que los eventos se desarrollen sin problemas.

## Vista de casos de uso

Caso de uso 1: Reservar una habitación

Descripcion: un cliente reserva una habitación en el sistema

Actores:

* Cliente
* Recepcionista

Precondiciones:

* El cliente debe estar registrado
* Debe haber al menos 1 habitacion disponible

Flujo de eventos:

* El cliente selecciona una opcion de lista de habitaciones disponibles
* El sistema muestra la lista de habitaciones
* El cliente decide que tipo de habitacion elije
* El sistema verifica la disponibilidad
* Si esta disponible, se entrega la llave y se guarda la informacion del cliente asi como su dia de llegada

Casos de exepcion:

* No hay habitaciones disponibles
* El cliente no esta registrado

## Vista lógica

El Sistema de Gestión para Hotel y Eventos está diseñado para manejar múltiples aspectos operativos del hotel y la planificación de eventos. El sistema se descompone en varios subsistemas o paquetes, cada uno responsable de un conjunto específico de funcionalidades. A continuación, se detallan los paquetes significativos y su descomposición en clases.

* EncargadoDeEventos:
  + Atributos:
    - Nombre :String
    - Eventos : String
    - Fecha: date
    - TipoDeEvento :String
  + Operaciones:
    - PlanificarEvento() :void
    - CordinarServicios() :void

Responsabilidades:

Es el que esta acargo de la coordinacion de eventos y los servicios de estos, ademas de planificarlos

* Eventos:
  + Atributos:
    - NombreDelEvento :String
    - FechaDelEvento : Date
    - EspacioReservado :String
    - ListaDeServicios :String
  + Operaciones:
    - AgregarServicios() :void
    - ConfirmarEventos() :void
* Factura:
  + Atributos:
    - Cliente :Cliente
    - Items : ItemFactura
    - Total :Float
  + Operaciones:
    - AgregaItem(item: itemFactura) :void
    - GenerarFactura() :void
* CLIENTE:
  + Atributos:
    - Nombre :String
    - Numero de reserva : Int
    - MetodoDePago :String
    - FechaDeEntrada :Date
    - FechaDeSalida :Date
  + Operaciones:
    - RealizarPago() :void
    - ReservarHabitacion() :void
* Habitacion:
  + Atributos:
    - NumeroDeHabitacion :Int
    - TipoDeHabitacion : String
    - Estado :Bolean
  + Operaciones:
    - Reservar() :void
    - Liberar() :void
* Reserva:
  + Atributos:
    - NombreDeReserva :String
    - Cliente : int
    - Habitacion :String
    - Fecha de entrada :Date
    - Fecha de salida :Date
  + Operaciones:
    - CalcularTotal() :void
    - ConfirmarReserva() :void
* PersonalDeRecepcion:
  + Atributos:
    - Nombre :String
    - Habitacion : Date
    - TipoDeServicio :String
  + Operaciones:
    - RealizarCheckIn() :void
    - RealizarChecOut() :void
    - GenerarFactura() :Voi

# 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.>>

En esta sección, se describen las elecciones de diseño específicas que guiarán el desarrollo del Sistema Integral de Gestión Hotelera y de Eventos del “Hotel Chachapa”. Se detallan aspectos cruciales como el tipo de desarrollo, las consideraciones de rendimiento, y los estándares de desarrollo y codificación que se seguirán. Estas directrices asegurarán la coherencia y calidad del desarrollo del sistema.

**Tipo de Desarrollo**

Para este proyecto, hemos optado por el desarrollo a medida del software. Esta elección se debe a la necesidad de personalizar las funcionalidades del sistema para que se ajusten específicamente a los procesos operativos y de gestión del “Hotel Chachapa”. Aunque la adquisición de software existente podría ofrecer una implementación más rápida, no proporcionaría el mismo nivel de adaptación y flexibilidad que un desarrollo personalizado.

Rendimiento

* El rendimiento del sistema se ha considerado cuidadosamente, equilibrando el uso del espacio en memoria y el tiempo de respuesta. Las decisiones de diseño incluyen:
* Optimización del Tiempo de Respuesta: Prioridad en la rapidez de las operaciones críticas como la reserva de habitaciones y eventos. Se utilizarán técnicas de caching y consultas optimizadas a la base de datos para mejorar la velocidad.
* Gestión del Espacio en Memoria: Uso eficiente de la memoria a través de técnicas de gestión de recursos y liberación de memoria no utilizada. Se evitará la sobrecarga de datos en memoria mediante la paginación y la carga diferida (lazy loading) de datos.

Estándares de Desarrollo

Para garantizar un desarrollo coherente y de alta calidad, se seguirán los siguientes estándares de desarrollo:

Asignación de Nombres:

* Variables y Métodos: Se utilizará el formato camelCase para nombres de variables y métodos (ejemplo: customerName, calculateTotal).
* Clases y Componentes: Se utilizará el formato PascalCase para nombres de clases y componentes (ejemplo: ReservationService, CustomerRepository).
* Constantes: Se utilizará el formato UPPER\_CASE con guiones bajos para constantes (ejemplo: MAX\_RETRY\_COUNT).

Mecanismos de Gestión de Excepciones:

* Captura y Manejo: Todas las excepciones serán capturadas y manejadas adecuadamente para evitar fallos inesperados del sistema. Se utilizarán bloques try-catch y se definirán excepciones personalizadas para casos específicos.
* Registro de Errores: Todos los errores y excepciones serán registrados utilizando un sistema de logging centralizado para facilitar la detección y resolución de problemas.

Estándares de Interfaces:

* APIs: Las interfaces de programación de aplicaciones (APIs) seguirán el estándar RESTful, con endpoints claros y bien documentados.
* UI/UX: Las interfaces de usuario serán diseñadas siguiendo principios de usabilidad y accesibilidad, garantizando una experiencia de usuario intuitiva y eficiente.

Estándares de Codificación:

* Formato del Código: Se utilizarán herramientas de formateo de código (como Prettier para JavaScript o ktlint para Kotlin) para asegurar un estilo de código consistente.
* Revisiones de Código: Se implementará un proceso de revisión de código (code reviews) en el que al menos otro desarrollador revise cada cambio antes de ser integrado en el código base.
* Documentación del Código: Todo el código deberá estar adecuadamente documentado utilizando comentarios y anotaciones para facilitar el mantenimiento y la comprensión del mismo.

Estos estándares y directrices proporcionarán una base sólida para el desarrollo del sistema, asegurando que todos los diseñadores e implementadores trabajen de manera coherente y eficiente hacia la consecución de los objetivos del proyecto.

## 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>>

**Crear Reservas de Habitaciones CU1**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Crear Reservas de Habitaciones CU1** |
| Descripción | Este caso de uso permite a los clientes y al personal de recepción crear una nueva reserva de habitación, verificando la disponibilidad y registrando la reserva. |
| Actores Principales | Cliente, Personal de Recepción |
| Actores Secundarios | Personal de mantenimiento |
| Precondiciones | El cliente debe estar registrado en el sistema. |
| Requerimientos. especiales  Po | El sistema debe permitir a los actores buscar disponibilidad de habitaciones.  El sistema debe permitir a los actores seleccionar fechas de check-in y check-out.  El sistema debe confirmar la reserva y actualizar la disponibilidad.  El sistema debe enviar una confirmación por email al cliente. |
| Poscondiciones | La reserva de la habitación es registrada en el sistema y el cliente recibe una confirmación por email. |
| Requerimientos no funcionales | El sistema debe responder en menos de 3 segundos para búsquedas de disponibilidad.  La confirmación del evento debe enviarse dentro de 1 minuto de la creación. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Cliente/ Personal de Recepción | El actor selecciona la opción para crear una nueva reserva. |
| Paso 2 | Sistema | El sistema muestra la disponibilidad de habitaciones. |
| Paso 3 | Cliente/ Personal de Recepción | El actor selecciona la habitación, fecha de check-in y check-out. |
| Paso 4 | Sistema | El sistema verifica la disponibilidad y confirma la reserva. |
| Paso 5 | Cliente/ Personal de Recepción | El sistema envía una confirmación de la reserva al cliente por email. |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si no hay disponibilidad de la habitación en las fechas seleccionadas: (Paso 4 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de no disponibilidad. |
| Paso 2 | Cliente/ Personal de Recepción | El actor puede seleccionar otras fechas o una habitación diferente. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que los clientes tienen acceso a internet para crear reservas.  Se asume que el personal de recepción tiene permisos adecuados para crear reservas. |

**Crear Eventos CU2**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Crear Eventos CU2** |
| Descripción | Este caso de uso permite al encargado de eventos y a los clientes corporativos crear un nuevo evento, organizando los recursos necesarios y reservando el espacio. |
| Actores Principales | Encargado de Eventos, Cliente Corporativo |
| Actores Secundarios | Personal de mantenimiento |
| Precondiciones | El cliente corporativo debe estar registrado en el sistema. |
| Requerimientos. especiales  Po | El sistema debe permitir a los actores buscar disponibilidad de espacios para eventos.  El sistema debe permitir a los actores seleccionar fechas y horas del evento.  El sistema debe permitir la configuración de recursos necesarios para el evento.  El sistema debe confirmar la reserva del espacio y enviar una confirmación por email al cliente. |
| Postcondiciones | El evento es registrado en el sistema y el cliente recibe una confirmación por email. |
| Requerimientos no funcionales | El sistema debe responder en menos de 3 segundos para búsquedas de disponibilidad.  La confirmación del evento debe enviarse dentro de 1 minuto de la creación. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Encargado de Eventos/Cliente Corporativo | El actor selecciona la opción para crear un nuevo evento. |
| Paso 2 | Sistema | El sistema muestra la disponibilidad de espacios para |
| Paso 3 | Encargado de Eventos/Cliente Corporativo | El actor selecciona el espacio, fecha y hora del evento. |
| Paso 4 | Sistema | El sistema verifica la disponibilidad y confirma la reserva del espacio. |
| Paso 5 | Sistema | . El sistema permite al actor configurar los recursos necesarios (equipamiento, catering, etc.). |
| Paso 6 | Sistema | El sistema envía una confirmación del evento al cliente por email |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si no hay disponibilidad del espacio en las fechas seleccionadas: (Paso 4 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de no disponibilidad. |
| Paso 2 | Encargado de Eventos/Cliente Corporativo | El actor puede seleccionar otras fechas o un espacio diferente. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que los clientes corporativos tienen acceso a internet para crear eventos.  Se asume que el encargado de eventos tiene permisos adecuados para crear eventos. |

**Registrar Nuevo Cliente CU3**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Registrar Nuevo Cliente CU3** |
| Descripción | Este caso de uso permite a los clientes y al personal de recepción registrar un nuevo cliente en el sistema, almacenando toda la información necesaria del cliente. |
| Actores Principales | Cliente, Personal de Recepción |
| Actores Secundarios | Personal de mantenimiento |
| Precondiciones | Ninguna (Ganas de registrarse). |
| Requerimientos. especiales  Po | El sistema debe permitir la entrada de datos completos del cliente.  El sistema debe validar los datos ingresados.  El sistema debe almacenar la información del cliente de manera segura.  El sistema debe confirmar el registro exitoso del cliente. |
| Postcondiciones | El cliente es registrado en el sistema y sus datos son almacenados. |
| Requerimientos no funcionales | El sistema debe validar y registrar al cliente en menos de 5 segundos.  La interfaz debe ser fácil de usar y accesible desde dispositivos móviles y de escritorio. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Cliente/Personal de Recepción | El actor selecciona la opción para registrar un nuevo cliente. |
| Paso 2 | Sistema | El sistema muestra un formulario para ingresar los datos del cliente. |
| Paso 3 | Encargado de Eventos/Cliente Corporativo | El actor ingresa los datos del cliente (nombre, dirección, correo electrónico, teléfono, etc.). |
| Paso 4 | Encargado de Eventos/Cliente Corporativo | El actor envía el formulario. |
| Paso 5 | Sistema | El sistema valida los datos y registra al nuevo cliente. |
| Paso 6 | Sistema | El sistema confirma el registro y muestra un mensaje de éxito. |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si los datos ingresados no son válidos: (Paso 5 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de error indicando los campos incorrectos. |
| Paso 2 | Encargado de Eventos/Cliente Corporativo | El actor corrige los datos y reenvía el formulario. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que el personal de recepción tiene los permisos adecuados para registrar nuevos clientes.  Se asume que los clientes tienen acceso a internet para registrarse ellos mismos si lo desean. |

**Procesar Facturación y Pagos CU4**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Procesar Facturación y Pagos CU4** |
| Descripción | Este caso de uso permite al cliente y al personal del hotel procesar la facturación y los pagos asociados a reservas de habitaciones y eventos, incluyendo la generación de facturas y el registro de transacciones de pago. |
| Actores Principales | Cliente, Personal de Recepción, Personal de Contabilidad |
| Actores Secundarios | Personal de mantenimiento |
| Precondiciones | El cliente debe tener una reserva confirmada o un evento planificado. |
| Requerimientos. especiales  Po | El sistema debe permitir la selección de reservas y eventos pendientes de pago.  El sistema debe generar facturas detalladas.  El sistema debe permitir la selección de diversos métodos de pago.  El sistema debe procesar pagos y actualizar el estado de las reservas y eventos.  El sistema debe enviar confirmaciones de pago y facturas por email. |
| Postcondiciones | La factura es generada y el pago es procesado, con la transacción registrada en el sistema. |
| Requerimientos no funcionales | El sistema debe procesar pagos en menos de 10 segundos.  La interfaz debe ser intuitiva y accesible desde dispositivos móviles y de escritorio.  La información financiera debe ser manejada con altos estándares de seguridad.. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Cliente/Personal de Recepción/ Personal de Contabilidad | El actor selecciona la opción para procesar la facturación y los pagos. |
| Paso 2 | Sistema | El sistema muestra las reservas y eventos pendientes de pago. |
| Paso 3 | Cliente/Personal de Recepción/ Personal de Contabilidad | El actor selecciona una reserva o evento para facturar. |
| Paso 4 | Sistema | El sistema genera una factura detallada. |
| Paso 5 | Cliente/Personal de Recepción/ Personal de Contabilidad | El actor selecciona el método de pago (tarjeta de crédito, PayPal, transferencia bancaria, etc.). |
| Paso 6 | Sistema | El sistema procesa el pago y actualiza el estado de la reserva o evento a pagado. |
| Paso 7 | Sistema | El sistema envía una confirmación del pago y la factura al cliente por email. |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si el método de pago seleccionado no es válido: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de error indicando el problema. |
| Paso 2 | Encargado de Eventos/Cliente Corporativo | El actor puede seleccionar un método de pago diferente. |

|  |  |  |
| --- | --- | --- |
| Si el pago es rechazado por el sistema de pago: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema notifica al actor del rechazo. |
| Paso 2 | Encargado de Eventos/Cliente Corporativo | El actor puede intentar nuevamente con el mismo u otro método de pago. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que los clientes tienen acceso a métodos de pago válidos.  Se asume que el personal de recepción y contabilidad tiene permisos adecuados para procesar facturación y pagos. |

**Gestionar Solicitud de Mantenimiento CU5**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Gestionar Solicitud de Mantenimiento CU5** |
| Descripción | Este caso de uso permite al personal del hotel crear, gestionar y actualizar solicitudes de mantenimiento, incluyendo la asignación de tareas y el seguimiento de incidencias. |
| Actores Principales | Personal de Mantenimiento, Personal de Recepción, Personal de Eventos |
| Actores Secundarios | Testers. |
| Precondiciones | El sistema debe estar operando y accesible para los actores. |
| Requerimientos. especiales  Po | El sistema debe permitir la entrada de detalles completos de la solicitud de mantenimiento.  El sistema debe validar los datos ingresados.  El sistema debe registrar la solicitud y asignarla automáticamente a un técnico disponible.  El sistema debe permitir la actualización del estado de la solicitud.  El sistema debe notificar al personal relevante sobre el estado de la solicitud por email. |
| Postcondiciones | La solicitud de mantenimiento es registrada, asignada a un técnico y su estado es actualizado en el sistema. |
| Requerimientos no funcionales | El sistema debe procesar y registrar la solicitud en menos de 5 segundos.  La interfaz debe ser fácil de usar y accesible desde dispositivos móviles y de escritorio.  La información debe ser manejada con altos estándares de seguridad y privacidad. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Personal de Recepción | El actor selecciona la opción para crear una nueva solicitud de mantenimiento |
| Paso 2 | Sistema | El sistema muestra un formulario para ingresar los detalles de la solicitud. |
| Paso 3 | Personal de Recepción | El actor ingresa los detalles de la solicitud (descripción del problema, ubicación, prioridad, etc.). |
| Paso 4 | Personal de Recepción | El actor envía el formulario. |
| Paso 5 | Sistema | El sistema valida los datos y registra la solicitud. |
| Paso 6 | Sistema | El sistema asigna automáticamente la solicitud a un técnico disponible. |
| Paso 7 | Sistema | El sistema actualiza el estado de la solicitud y notifica al personal de mantenimiento asignado. |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si los datos ingresados no son válidos: (Paso 5 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de error indicando los campos incorrectos. |
| Paso 2 | Personal de Recepción | El actor corrige los datos y reenvía el formulario. |

|  |  |  |
| --- | --- | --- |
| Si no hay técnicos disponibles: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema marca la solicitud como pendiente de asignación. |
| Paso 2 | Sistema | El sistema notifica al personal de recepción o eventos para seguimiento manual. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que el personal de mantenimiento y recepción tiene los permisos adecuados para crear y gestionar solicitudes de mantenimiento.  Se asume que el sistema de gestión de mantenimiento está correctamente integrado con el resto del sistema. |

**Gestionar Notificaciones CU6**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Gestionar Notificaciones CU6** |
| Descripción | Este caso de uso permite al administrador del sistema y otros actores relevantes crear, gestionar y enviar notificaciones a los clientes y al personal del hotel. |
| Actores Principales | Administrador del Sistema, Personal de Recepción, Encargado de Eventos, Personal de Mantenimiento, Cliente |
| Actores Secundarios | Personal de mantenimiento. |
| Precondiciones | El sistema debe estar operativo y accesible para los actores. |
| Requerimientos. especiales  Po | El sistema debe permitir la creación y gestión de notificaciones.  El sistema debe validar los datos ingresados.  El sistema debe enviar las notificaciones a los destinatarios seleccionados.  El sistema debe registrar y actualizar el estado de las notificaciones.  El sistema debe mostrar confirmaciones y mensajes de error según corresponda. |
| Postcondiciones | Las notificaciones son enviadas y registradas en el sistema, y los actores relevantes reciben las notificaciones. |
| Requerimientos no funcionales | El sistema debe permitir la creación y gestión de notificaciones.  El sistema debe validar los datos ingresados.  El sistema debe enviar las notificaciones a los destinatarios seleccionados.  El sistema debe registrar y actualizar el estado de las notificaciones.  El sistema debe mostrar confirmaciones y mensajes de error según corresponda. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Administrador del Sistema, Personal de Recepción, Encargado de Eventos, Personal de Mantenimiento, Cliente | El actor selecciona la opción para gestionar notificaciones. |
| Paso 2 | Sistema | El sistema muestra una lista de notificaciones existentes y una opción para crear nuevas. |
| Paso 3 | Administrador del Sistema, Personal de Recepción, Encargado de Eventos, Personal de Mantenimiento, Cliente | El actor selecciona la opción para crear una nueva notificación |
| Paso 4 | Sistema | El sistema muestra un formulario para ingresar los detalles de la notificación (mensaje, destinatarios, tipo de notificación, etc.). |
| Paso 5 | Administrador del Sistema, Personal de Recepción, Encargado de Eventos, Personal de Mantenimiento, Cliente | El actor ingresa los detalles y envía el formulario. |
| Paso 6 | Sistema | El sistema valida los datos y registra la notificación. |
| Paso 7 | Sistema | El sistema envía la notificación a los destinatarios y actualiza el estado de la notificación en el sistema. |
| Paso 8 | Sistema | El sistema muestra una confirmación del envío exitoso. |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si los datos ingresados no son válidos: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de error indicando los campos incorrectos. |
| Paso 2 | Administrador del Sistema, Personal de Recepción, Encargado de Eventos, Personal de Mantenimiento, Cliente | El actor corrige los datos y reenvía el formulario. |

|  |  |  |
| --- | --- | --- |
| Si hay un error en el envío de la notificación: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de error indicando el problema. |
| Paso 2 | Administrador del Sistema, Personal de Recepción, Encargado de Eventos, Personal de Mantenimiento, Cliente | El actor puede intentar reenviar la notificación. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que el administrador del sistema y el personal relevante tienen los permisos adecuados para gestionar notificaciones.  Se asume que los destinatarios de las notificaciones tienen acceso a dispositivos donde puedan recibirlas. |

**Gestionar el Calendario de Eventos CU7**

|  |  |
| --- | --- |
| **Número y nombre del caso de uso** | **Gestionar el Calendario de Eventos CU7** |
| Descripción | Este caso de uso permite al personal del hotel gestionar el calendario de eventos, incluyendo la creación, modificación y cancelación de eventos, así como la coordinación de recursos. |
| Actores Principales | Administrador del Sistema, Encargado de Eventos, Personal de Recepción, Cliente |
| Actores Secundarios | Personal de mantenimiento. |
| Precondiciones | El sistema debe estar operativo y accesible para los actores. |
| Requerimientos. especiales  Po | El sistema debe permitir la creación, modificación y eliminación de eventos en el calendario.  El sistema debe validar los datos ingresados.  El sistema debe notificar a los actores relevantes sobre los cambios en el calendario.  El sistema debe mostrar el calendario actualizado con todos los eventos programados. |
| Postcondiciones | El evento es registrado, actualizado o eliminado en el calendario del sistema, y las notificaciones correspondientes son enviadas a los actores relevantes. |
| Requerimientos no funcionales | El sistema debe procesar las actualizaciones del calendario en menos de 3 segundos.  La interfaz debe ser intuitiva y accesible desde dispositivos móviles y de escritorio.  La información debe ser manejada con altos estándares de seguridad y privacidad. |

Flujo de eventos (Escenario normal)

|  |  |  |
| --- | --- | --- |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Administrador del Sistema, Encargado de Eventos, Personal de Recepción, Cliente | El actor selecciona la opción para gestionar el calendario de eventos. |
| Paso 2 | Sistema | El sistema muestra el calendario actual con los eventos programados. |
| Paso 3 | Administrador del Sistema, Encargado de Eventos, Personal de Recepción, Cliente | El actor selecciona la opción para crear, modificar o eliminar un evento. |
| Paso 4 | Sistema | El sistema muestra un formulario para ingresar o modificar los detalles del evento (nombre, fecha, hora, ubicación, recursos necesarios, etc.). |
| Paso 5 | Administrador del Sistema, Encargado de Eventos, Personal de Recepción, Cliente | El actor ingresa o actualiza los detalles del evento y envía el formulario. |
| Paso 6 | Sistema | El sistema valida los datos y actualiza el calendario con los cambios |
| Paso 7 | Sistema | El sistema notifica a los actores relevantes sobre los cambios realizados. |

Flujo Alternos

|  |  |  |
| --- | --- | --- |
| Si los datos ingresados no son válidos: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de error indicando los campos incorrectos. |
| Paso 2 | Administrador del Sistema, Encargado de Eventos, Personal de Recepción, Cliente | El actor corrige los datos y reenvía el formulario. |

|  |  |  |
| --- | --- | --- |
| Si hay un conflicto de horarios: (Paso 6 del flujo normal) | | |
| **Paso** | **Actor** | **Descripción** |
| Paso 1 | Sistema | El sistema muestra un mensaje de conflicto de horarios. |
| Paso 2 | Administrador del Sistema, Encargado de Eventos, Personal de Recepción, Cliente | El actor puede elegir otro horario o revisar los eventos existentes. |

Puntos Abiertos

|  |  |
| --- | --- |
| **Punto** | **Descripción** |
| Punto | Se asume que el administrador del sistema y el personal relevante tienen los permisos adecuados para gestionar los Eventos. |

**Diagrama**

Diagrama, Esquemático

Descripción generada automáticamente

Ilustración 5 Diagrama de Casos de Uso

## 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.>>

* EncargadoDeEventos:
  + Atributos:
    - Nombre :String
    - Eventos : String
    - Fecha: date
    - TipoDeEvento :String
  + Operaciones:
    - PlanificarEvento() :void
    - CordinarServicios() :void
* Eventos:
  + Atributos:
    - NombreDelEvento :String
    - FechaDelEvento : Date
    - EspacioReservado :String
    - ListaDeServicios :String
  + Operaciones:
    - AgregarServicios() :void
    - ConfirmarEventos() :void
* Factura:
  + Atributos:
    - Cliente :Cliente
    - Items : ItemFactura
    - Total :Float
  + Operaciones:
    - AgregaItem(item: itemFactura) :void
    - GenerarFactura() :void
* CLIENTE:
  + Atributos:
    - Nombre :String
    - Numero de reserva : Int
    - MetodoDePago :String
    - FechaDeEntrada :Date
    - FechaDeSalida :Date
  + Operaciones:
    - RealizarPago() :void
    - ReservarHabitacion() :void
* Habitacion:
  + Atributos:
    - NumeroDeHabitacion :Int
    - TipoDeHabitacion : String
    - Estado :Bolean
  + Operaciones:
    - Reservar() :void
    - Liberar() :void
* Reserva:
  + Atributos:
    - NombreDeReserva :String
    - Cliente : int
    - Habitacion :String
    - Fecha de entrada :Date
    - Fecha de salida :Date
  + Operaciones:
    - CalcularTotal() :void
    - ConfirmarReserva() :void
* PersonalDeRecepcion:
  + Atributos:
    - Nombre :String
    - Habitacion : Date
    - TipoDeServicio :String
  + Operaciones:
    - RealizarCheckIn() :void
    - RealizarChecOut() :void
    - GenerarFactura() :Voi

# Diseño de la arquitectura de módulos del sistema

**Módulo de Gestión de Reservas:**

Descripción:

Este módulo se encarga de la gestión completa de las reservas de habitaciones y salas de eventos, incluyendo la consulta de disponibilidad, creación, modificación y cancelación de reservas.

Submódulos:

Consulta de disponibilidad: Permite verificar la disponibilidad de habitaciones y salas.

Creación de Reservas: Facilita la creación de nuevas reservas.

Modificación de Reservas: Permite modificar las reservas existentes.

Cancelación de Reservas: Gestiona la cancelación de reservas.

**Módulo de Gestión de Eventos:**

Descripción:

Gestiona la planificación, administración y seguimiento de eventos, incluyendo la asignación de recursos y gestión de invitados.

Submódulos:

Planificación de Eventos: Facilita la creación y planificación de eventos.

Administración de Recursos: Gestiona los recursos necesarios para eventos.

Seguimiento de Eventos: Permite el seguimiento del estado de los eventos.

Gestión de Invitados: Administra la lista de invitados y envía invitaciones.

**Módulo de Gestión de Clientes:**

Descripción:

Gestiona los datos de los clientes, incluyendo el registro, actualización de información y consulta de historial.

Submódulos:

Registro de Clientes: Permite el registro de nuevos clientes.

Actualización de Información: Facilita la actualización de datos del cliente.

Consulta de Historial: Proporciona acceso al historial de reservas y transacciones.

**Módulo de Facturación y Pagos:**

Descripción:

Gestiona la facturación y los pagos realizados por los clientes, incluyendo la generación de facturas, procesamiento de pagos y emisión de recibos.

Submódulos:

Generación de Facturas: Crea facturas detalladas para reservas y eventos.

Procesamiento de Pagos: Gestiona los pagos de los clientes.

Emisión de Recibos: Proporciona recibos electrónicos tras la confirmación de pago.

Informes Financieros: Genera informes financieros.

**Módulo de Gestión de Servicios Adicionales:**

Descripción:

Gestiona los servicios adicionales ofrecidos por el hotel, como spa, transporte y excursiones.

Submódulos:

Reserva de Servicios: Permite reservar servicios adicionales.

Gestión de Disponibilidad: Administra la disponibilidad de los servicios.

Facturación de Servicios: Gestiona la facturación de servicios adicionales.

**Módulo de Integración con Plataformas Externas:**

Descripción:

Facilita la integración con plataformas externas, como agencias de viaje, sistemas de pago y redes sociales.

Submódulos:

API de Integración: Proporciona interfaces para la integración con sistemas externos.

Sincronización de Datos: Gestiona la sincronización de datos con plataformas externas.

Gestión de autenticaciones: Administra las autenticaciones con servicios externos.

**Módulo de Gestión de Usuarios y Roles:**

Administra los usuarios del sistema y sus roles, asegurando la correcta asignación de permisos.

Submódulos:

Registro de Usuarios: Permite el registro de nuevos usuarios.

Asignación de Roles: Administra la asignación de roles y permisos.

Autenticación y Seguridad: Gestiona la autenticación y políticas de seguridad.

**Módulo de Reportes y Análisis:**

Descripción:

Genera reportes y análisis de datos para la toma de decisiones.

Submódulos:

Generación de Reportes: Crea reportes detallados sobre diversas operaciones.

Análisis de Datos: Realiza análisis de datos para identificar tendencias y patrones.

Dashboards: Proporciona paneles de control interactivos.

**Módulo de Gestión del Calendario:**

Descripción:

Administra los calendarios de disponibilidad para habitaciones, salas de eventos y recursos.

Submódulos:

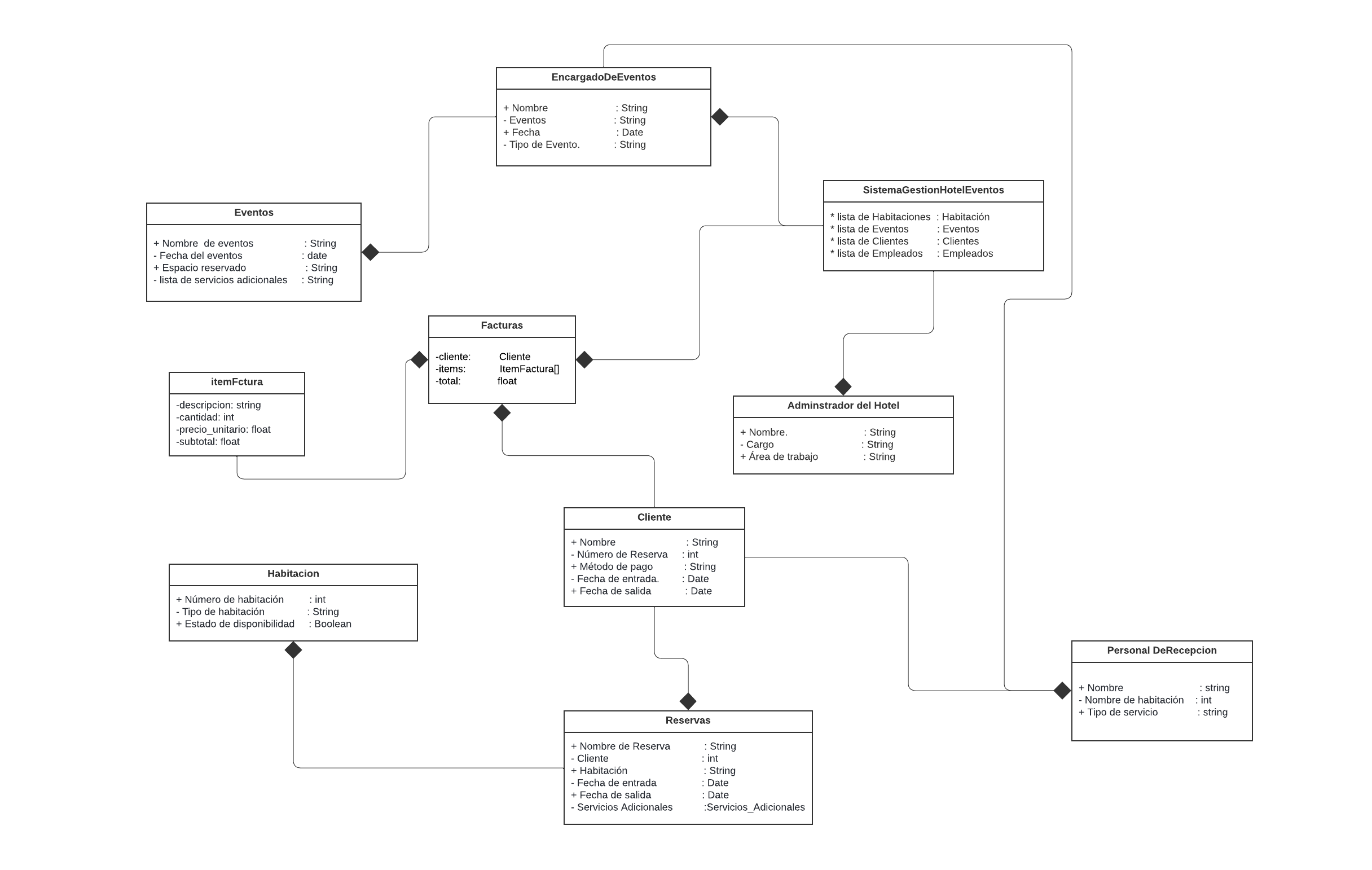
Calendario de Habitaciones: Gestiona la disponibilidad de habitaciones.

Calendario de Eventos: Administra la disponibilidad de salas de eventos.

Calendario de Recursos: Gestiona la disponibilidad de recursos adicionales.

<<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)

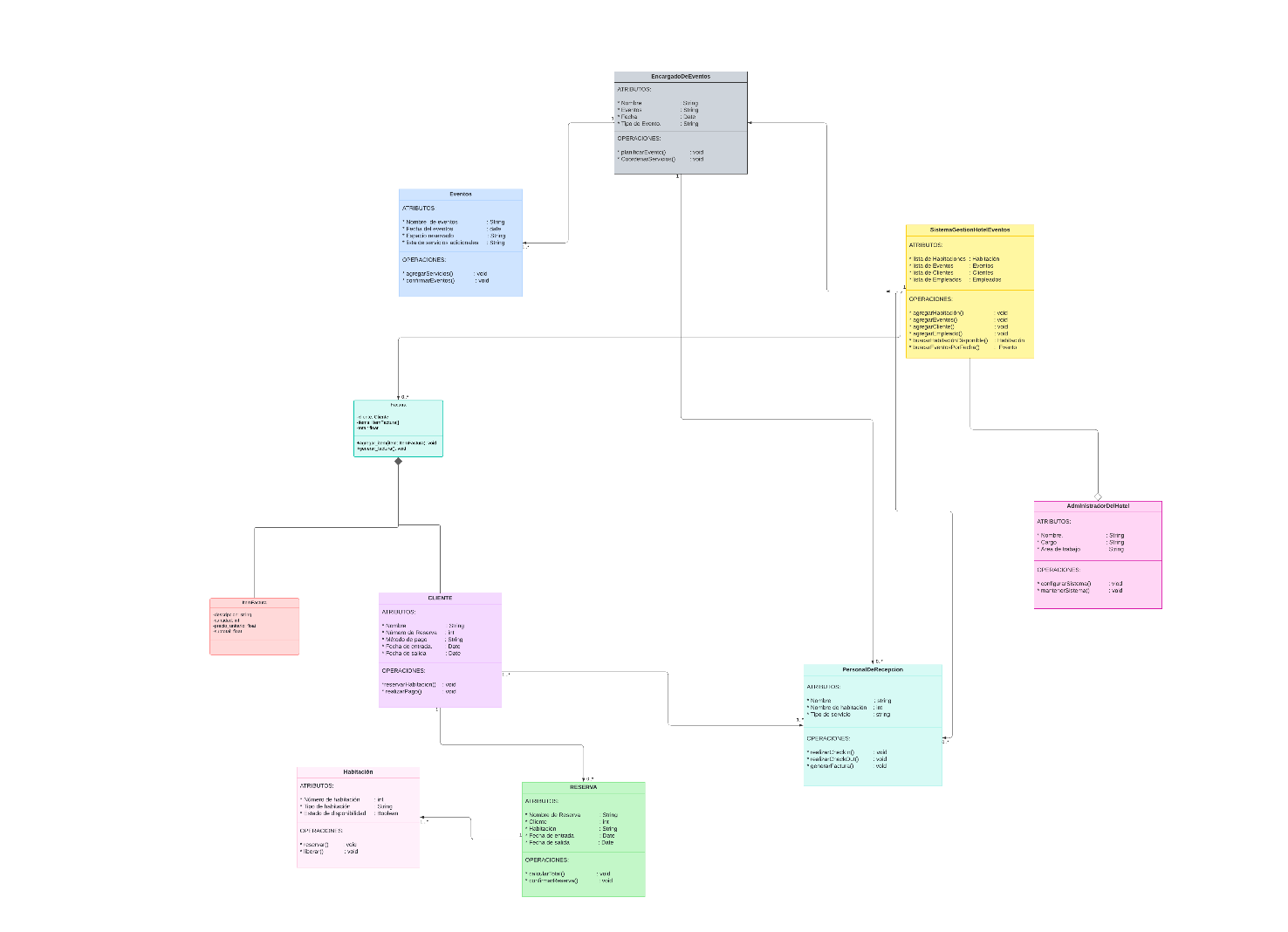


Ilustración 6 Diagrama de Clases

EncargadoDeEventos:

El encargado de eventos tiene atributos para identificarlos y el tipo de eventos que maneja, en sus operaciones esta planificarEvento y coordinarServicios, estos hacen funcional la clase para lo que es, encargarse de los eventos

Eventos:

La clase eventos tiene un nombre, su fecha, su espacio y sus servicios adicionales comoa tributos, y en las operaciones que tiene son agregar servicios, es decir pueden pedir servicios despues de que se hayan incrito y el de confirmar evento

SistemaGestionHotelEventos:

Esta clase controlara los eventos , sus atributos son la lista de habitaciones, la lista de eventos, la alista de clientes y la lista de empleados que se utilizaran para el evento, entre sus operaciones esta el de agregar habitaciones, agregar eventos, agregar clientes, buscar habitaciones disponibles y un identificador

AdministradorDelHotel:

Sus atributos tiene nombre y cargo para identificarlos ademas de su area de trabajjo, el puede configurar el sistema y hacer mantenimiento del mismo

PersonalDeRecepcion:

Tiene nombre, nombre de habitaciones encargadas, tipo de servicio y puede realizar un checkin, checkout y generar una factura, es el unico capacitado para poder generar una factura

Facturas:

Las facturas tienen el nombre del cliente, los items osease lo que compro y el total de la cuenta, la factura tambien puede agregar items e imprimirse

ItemFactura:

Es la factura impresa, solo tiene atributos y es la descripcion, cantidad y el precio unitario asi como el subtotal

CLIENTE:

El cliente tiene sus atributos para identificarlos como lo es su nombre, numero de reserva el metodo de pago y la fehca de entrada como su fecha de salida, el cleinte solo puede realizar su pago y reservar una habitacion

Habitacion:

Tiene de atributos identificables su numero de habitacion, su tipo de habitacion y el estado de disponibilidad que es un booleano, puede reservar y liberar

En el siguiente link se incluye el modelo entidad Relacion para sus detalles:

<https://lucid.app/lucidchart/bea5f4c3-4425-4da5-88f7-8575a3a45488/edit?viewport_loc=40%2C300%2C6492%2C3328%2CHWEp-vi-RSFO&invitationId=inv_a5a03939-476b-432d-b75c-8c10a82e9c02>

## Diccionario de datos.

## Diccionario de datos.

**Tabla 2 Diccionario de datos**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - 1 | EncargadoDeEventos: | | | | | |
| Descripción | | EncargadoDeEventos | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| 1 | id | Identificador único del encargado de eventos | | int | No | PK | - | si |
| 2 | nombre | Nombre del encargado | | Char( 255) | No | - | - | No |
| 3 | tipoEvent | Tipo de evento que maneja | | Char( 255) | No | - | - | No |
|  |  |  | |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - 2 | Eventos | | | | | |
| Descripción | |  | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| 1 | id | Identificador único del evento | | int | No | PK | - | si |
| 2 | nombre | Nombre del evento | | Char( 255) | No | - | - | No |
| 3 | fecha | Fecha del evento | | Date | No | - | - | No |
| 4 | espacio | Espacio donde se realizará | | Char( 255) | No | - | - | No |
| 5 | serviciosAdicionales | La inclusión de servicios adicionales | | JSON | Si | - | - | No |
| 6 | encargadoid | ID del encargado de eventos | | int | No | FK | - | No |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - 3 | SistemaGestionHotelEventos | | | | | |
| Descripción | |  | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| 1 | id | Identificador único | | int | No | PK | - | si |
| 2 | ListaHabitaciones | Lista de habitaciones (JSON) | | JSON | Si | - | - | No |
| 3 | listaEventos | Lista de eventos (JSON) | | JSON | Si | - | - | No |
| 4 | listaClientes | Lista de clientes (JSON) | | JSON | Si | - | - | No |
| 5 | listaEmpleados | Lista de empleados (JSON) | | JSON | Si | - | - | No |
|  |  |  | |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - 4 | AdministradorDelHotel | | | | | |
| Descripción | |  | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| 1 | id | Identificador único | | int | No | PK | - | si |
| 2 | nombre | Nombre del administrador | | Char (255) | No | - | - | No |
| 3 | cargo | Cargo del administrador | | Char (255) | No | - | - | No |
| 4 | areaTrabajo | Área de trabajo | | Char (255) | No | - | - | No |
|  |  |  | |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - 5 | PersonalDeRecepcion | | | | | |
| Descripción | |  | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| 1 | id | Identificador único | | int | No | PK | - | si |
| 2 | nombre | Nombre del personal | | Char (255) | No | - | - | No |
| 3 | HabitacionesEncargadas | Lista de habitaciones encargadas (JSON) | | JSON | Si | - | - | No |
| 4 | tipoServicio | Tipo de servicio | | Char (255) | No | - | - | No |
|  |  |  | |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Número y nombre de**  **la entidad** | | TBL - 6 | AdministradorDelHotel | | | | | |
| Descripción | |  | | | | | | |
| Nro. | Nombre de la Columna | Descripción | | Tipo de datos | Permitir Valores NULL | Llave | Valor Predeter minado | Auto numér ico |
| 1 | id | Identificador único | | int | No | PK | - | si |
| 2 | nombre | Nombre del administrador | | Char (255) | No | - | - | No |
| 3 | cargo | Cargo del administrador | | Char (255) | No | - | - | No |
| 4 | areaTrabajo | Área de trabajo | | Char (255) | No | - | - | No |
|  |  |  | |  |  |  |  |  |

# 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.

Nuestro diseño de software para un sistema de reserva de habitaciones y eventos del “Hotel Chachapa” consta de 8 pantallas de navegación para el cliente .

Pantalla 1 Home:

En esta se muestra la pantalla inicial del sitio web llamada home en la cual se pueden redirigir a todas las demás, empezando con 2 ventanas principales ya sea habitaciones o eventos en las cuales tiene informacion para cada tipo de reserva, tambien contamos con pantallas secundarias ya sea contacto para que nos puedas contactar, conocenos para ver información más detallada del hotel, una para iniciar sesión y una que te redirige a la pantalla principal home.



Ilustración 7

Pantalla 2 Reserva de Cuarto:

En este espacio esta donde se puede reservar un cuarto conteniendo 3 tipos de habitaciones asi como información de cada habitación ya sea sencilla, double y home, también contiene preguntas frecuentes.

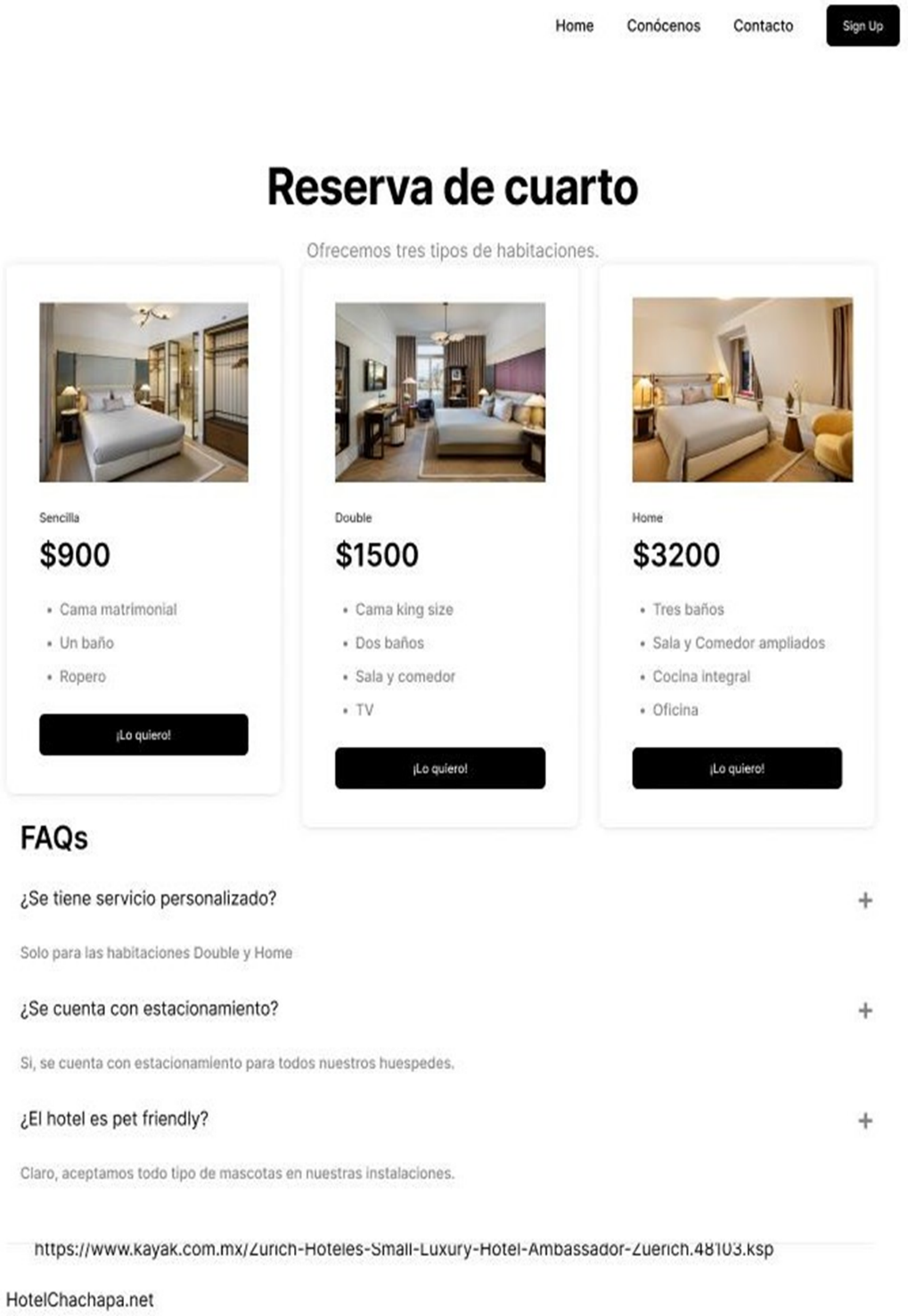


Ilustración 8

Pantalla 3 Reserva de eventos:

En esta pantalla tenemos igual que en la reserva de cuartos 3 “para diferentes tipos de reserva, de acuerdo con las necesidades del cliente e igual, las preguntas frecuentes.

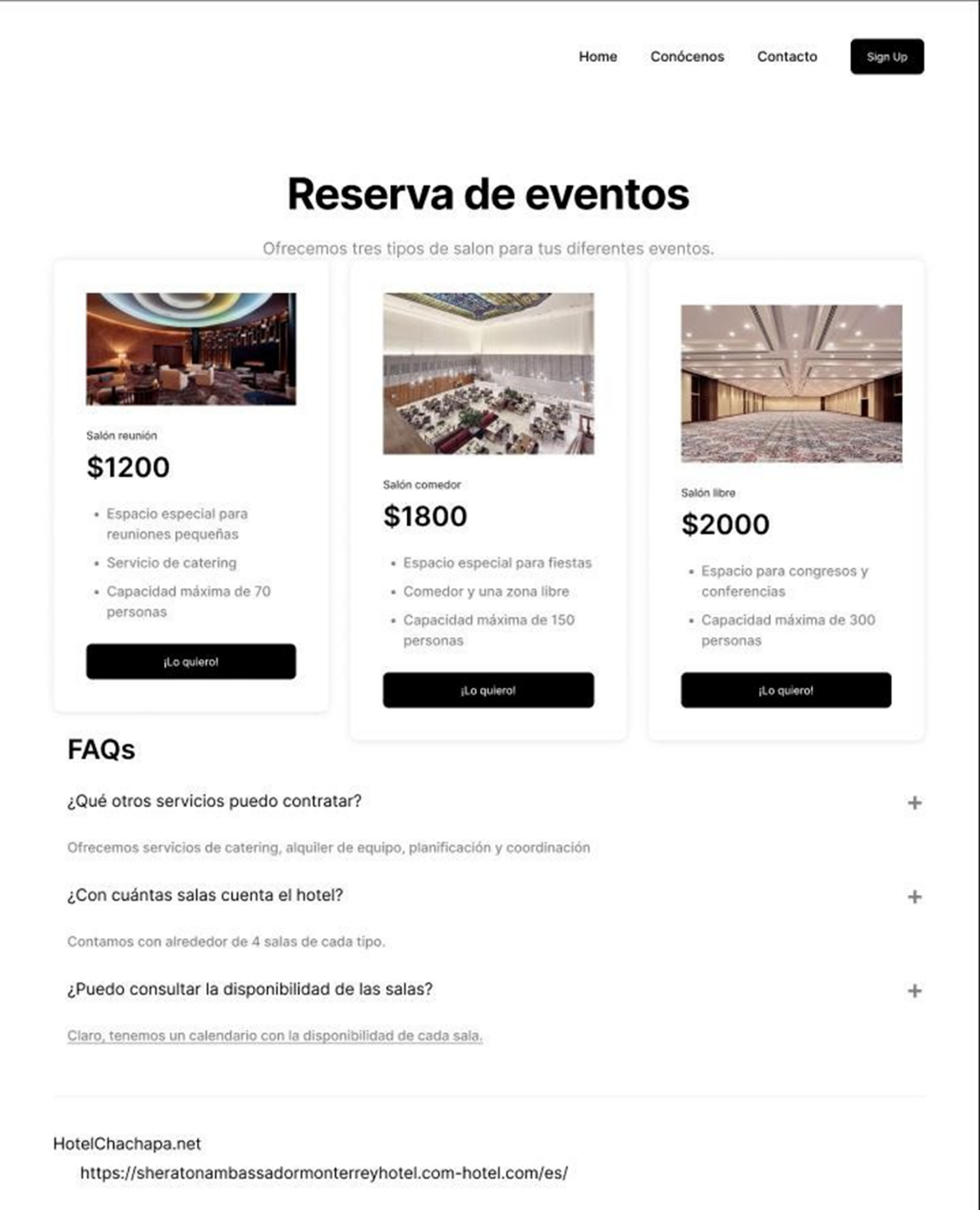


Ilustración 9

## Diccionario de datos.

Pantalla 4 Conócenos/Acerca de nosotros:

En esta esta la información acerca del hotel para que el usuario pueda consultar la misión, visión y pueda conocer acerca de porque es más de un hotel.



Ilustración 10

Pantalla 5 Crear cuenta o Iniciar sesión:

En esta únicamente se le pregunta al usuario información para crear su cuenta o para iniciar su cuenta.

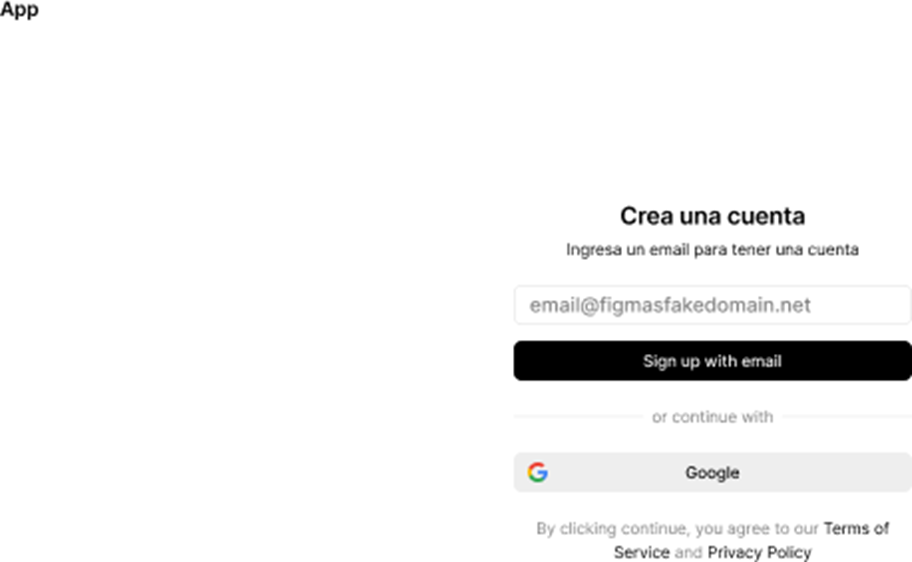


Ilustración 11

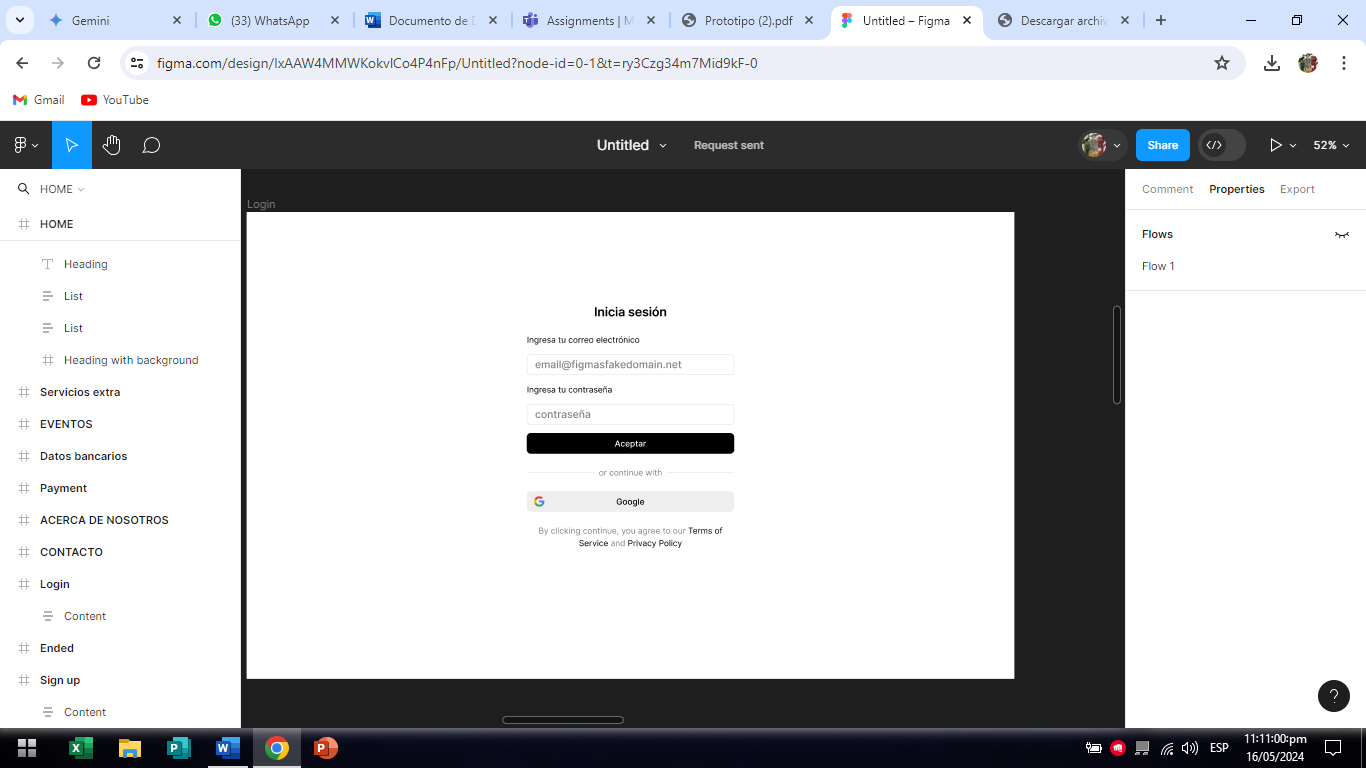


Ilustración 12

Pantalla 6 Contáctanos:

En esta página es como nos puede contactar ya sea para evento o habitación, siendo una herramienta importante.

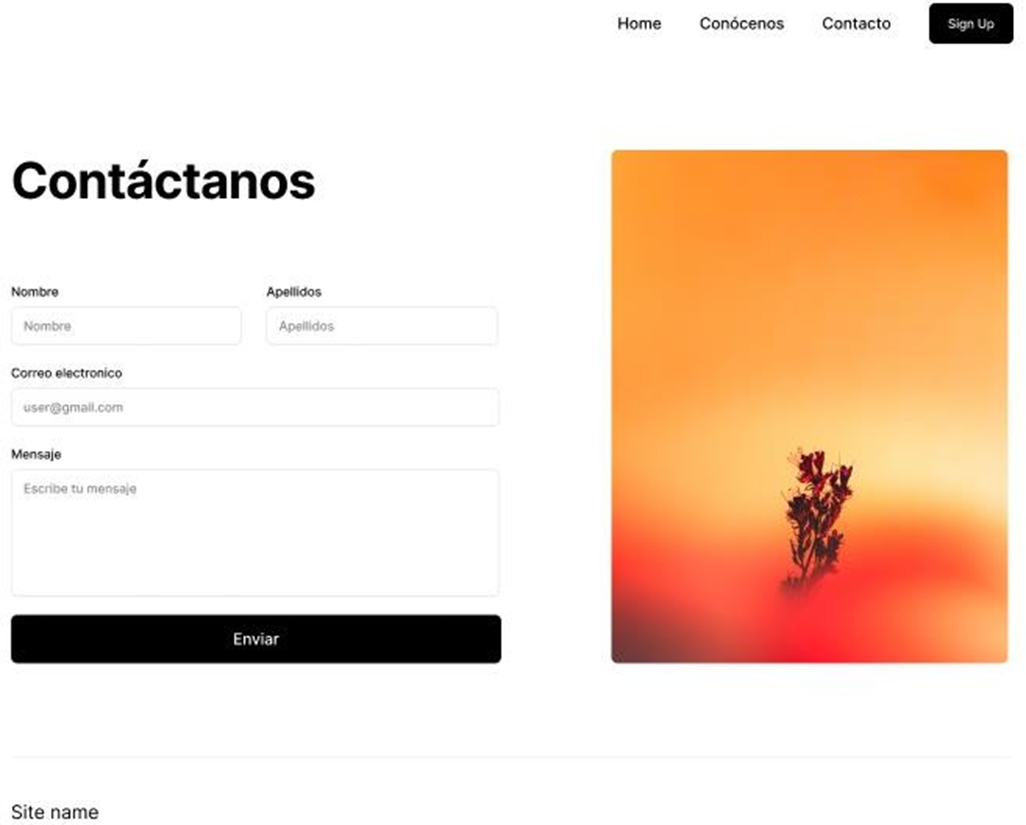


Ilustración 13

Pantalla 7 Servicios extra:

En esta página viene información extra acerca de otros servicios extra que incluye el hotel muy aparte de reserva habitaciones/eventos.

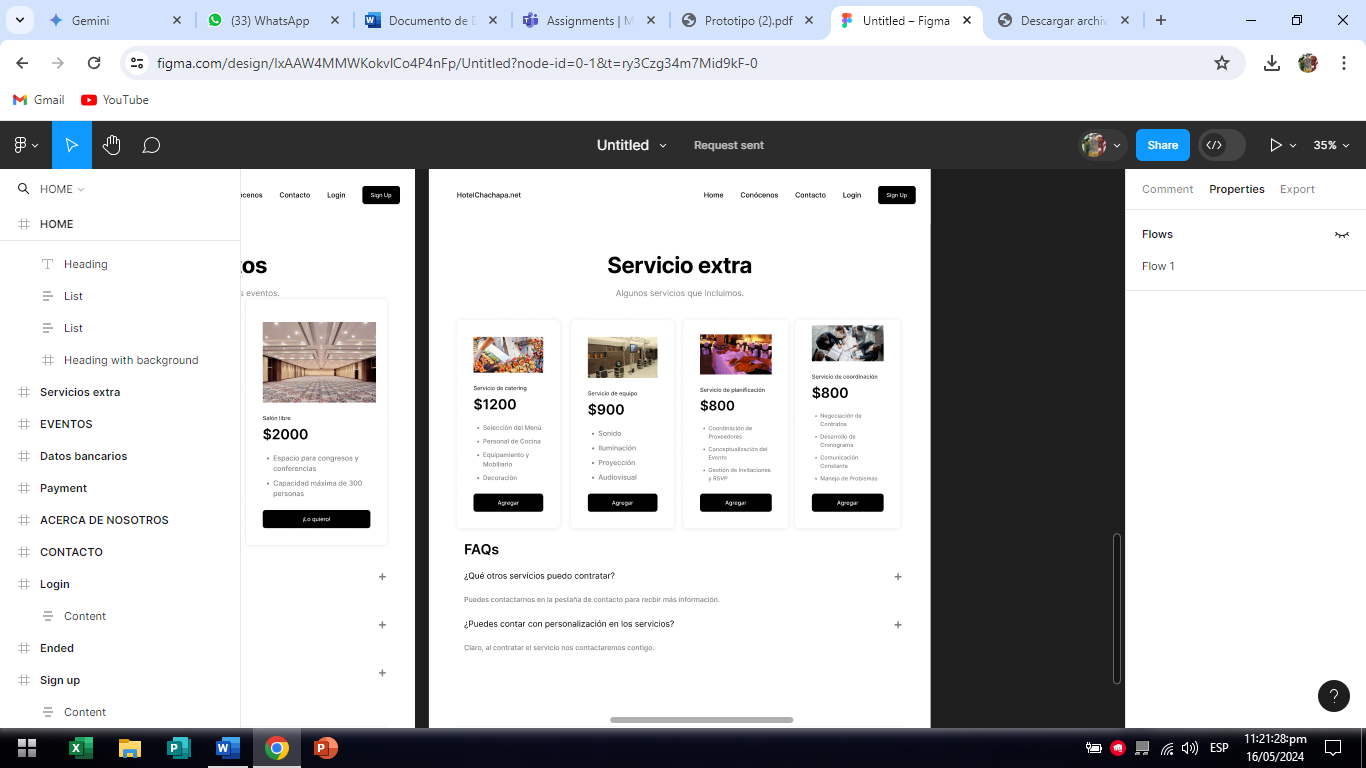


Ilustración 14

Pantalla 8 Datos de pago:

Informacion que tiene que dar el cliente para poder reservar o querer algun servicio que el hotel ofrece en el cual le pide información bancaria protegida hacia al cliente para que no haya robo de información por 3eros.

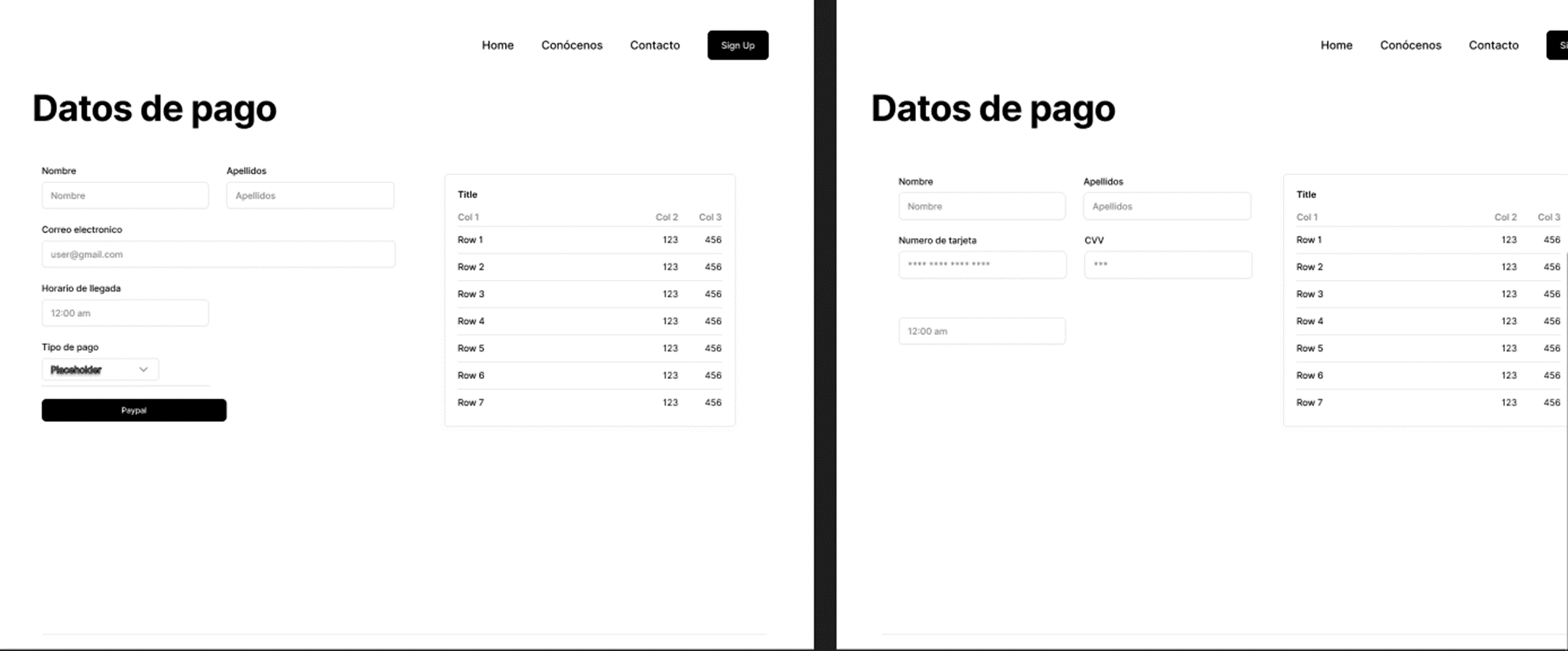


Ilustración 15