

**Universidad de San Carlos de Guatemala.
Centro Universitario de Occidente.
División de Ciencias de la Ingeniería.
Laboratorio Teoría de Sistemas 1.
Ing. Pedro Domingo**



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

“SISTEMA HOTELERÍA”

**Estudiante: Diego José Maldonado Monterroso.
Carné: 201931811
Enlace a repositorios:**

- <https://github.com/DiegoMaldonado19/hoteleria-api-rest>
- <https://github.com/DiegoMaldonado19/hoteleria-frontend>

Quetzaltenango, Guatemala.

21 de noviembre de 2023.

Presentación del Proyecto y Pensamiento Sistémico:

En este análisis del sistema de recepción de un hotel, se ha adoptado un enfoque sistémico para comprender y evaluar su funcionamiento. El sistema de recepción es un componente crítico de la operación hotelera, y su eficiencia y eficacia son fundamentales para garantizar una experiencia positiva para los huéspedes. A través de este enfoque, se exploraron los roles, interacciones y subsistemas que conforman este sistema complejo.

Principales Objetivos:

El objetivo principal del sistema de recepción de un hotel es proporcionar un servicio eficiente y eficaz a los huéspedes. Este objetivo se desglosa en los siguientes subobjetivos:

- Facilitar el proceso de registro (check-in) y salida (check-out) de los huéspedes de manera rápida y sin complicaciones.
- Ofrecer información precisa y detallada sobre el hotel y sus servicios.
- Satisfacer las necesidades y solicitudes de los huéspedes, como reservas de actividades, transporte y otras consultas.
- Gestionar las reservas de habitaciones de manera eficiente y mantener actualizada la información sobre los huéspedes.
- Fomentar una comunicación efectiva entre los empleados del hotel para brindar un servicio coordinado y cohesivo con el uso de las tecnologías que se implementarán.

Tecnologías a Utilizar:

El sistema de recepción de un hotel hace uso de varias tecnologías para lograr sus objetivos:

- **Sistema de Gestión de Reservas:** Este sistema es fundamental para administrar las reservas de habitaciones. Utiliza software especializado que permite verificar la disponibilidad de habitaciones, asignar habitaciones a los huéspedes y mantener un registro de las reservas.
- **Sistema de Información del Huésped:** Este sistema almacena datos personales de los huéspedes, detalles de sus reservas, preferencias y otra información relevante. Utiliza bases de datos y software de gestión de clientes para mantener registros actualizados.
- **Sistema de Comunicación Interna:** La comunicación efectiva entre los empleados del hotel es esencial. Para ello, se utilizan tecnologías de comunicación interna, como teléfonos, sistemas de mensajería y correo electrónico, que permiten una interacción rápida y fluida entre los miembros del personal.
- **Automatización:** La automatización desempeña un papel importante en la eficiencia del sistema. Los quioscos de autoservicio para el check-in y el check-out, así como la integración con sistemas de pagos electrónicos, pueden acelerar el proceso y reducir los errores humanos.
- **Software de Reservas en Línea:** Para los agentes de reservas y los huéspedes que realizan reservas en línea, se utilizan sistemas de reservas en línea que permiten una reserva conveniente a través de sitios web o aplicaciones móviles.

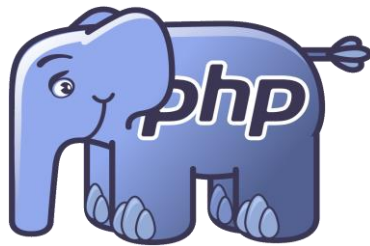
En resumen, el sistema de recepción de un hotel es un ejemplo de cómo un enfoque sistémico puede ayudar a comprender y mejorar la operación de un sistema complejo. Al analizar las partes, interacciones y tecnologías involucradas, es posible identificar oportunidades para aumentar la eficiencia y la satisfacción de los huéspedes, lo que a su vez contribuye al éxito general del hotel.

1. **HTML5, SCSS y TypeScript:** El frontend del sistema se realizará con estas tres tecnologías base, ya que son el core de nuestras vistas, dándole una estructura, un estilo y una funcionalidad para poder renderizar los datos obtenidos desde nuestra API Rest.
2. **PHP 8.2:** PHP 8.2 será el lenguaje de programación del lado del servidor encargado de la lógica de la plataforma hotelera. Esto incluye la generación de contenido dinámico, el procesamiento de formularios y la interacción con la base de datos PostgreSQL, que almacena y gestiona información crucial como reservas, transacciones y datos de usuarios. Estas tecnologías aseguran el funcionamiento eficiente y seguro de la plataforma.
3. **PostgreSQL:** será el motor de base de datos elegido para almacenar toda la información generada en el sistema. Este motor tiene muy buenos antecedentes, ya que ofrece muchas funcionalidades base que podemos utilizar para poder interactuar con nuestros datos desde nuestra API Rest.
4. **Node 18.18.2:** será el gestor de paquetes que nos ayudará a manejar todos los paquetes necesarios en nuestro frontend, para poder renderizar de manera correcta toda la información.

5. Laravel 10.10: Es el framework elegido para poder crear una API Rest que tenga comunicación con nuestra base de datos en PostgreSQL, este framework usa PHP como base, y pues nos otorga muchas funcionalidades como el ORM que posee, para poder realizar consultas sin tener Querys SQL directamente. Así como el fácil mapeo de la base datos con sus modelos.
6. Angular 16.2.12: Es el framework elegido para crear el Frontend de esta aplicación con la que interactuará nuestro usuario final. Es un framework bastante poderoso y que nos otorga bastante escalabilidad en nuestra propuesta.

Al combinar estas tecnologías, se creará una plataforma sólida y escalable para la gestión hotelera. HTML5, SCSS, TypeScript y Angular garantizarán una experiencia de usuario atractiva, mientras que PHP 8.2, PostgreSQL y Laravel permitirán gestionar eficientemente los datos y la lógica del sistema. El uso de Laravel y Angular brindará aplicaciones robustas para la gestión interna. Esto proporcionará a los empleados y huéspedes una herramienta valiosa para una experiencia hotelera excepcional.

HTML



Reservaciones:

El sistema de hotelería también se encarga de la gestión de reservaciones, lo cual es esencial para el funcionamiento eficiente de cualquier hotel. Aquí se detallan algunos aspectos clave relacionados con las reservaciones:

- **Reservaciones en Línea:** El sistema permite a los huéspedes realizar reservas de habitaciones de manera conveniente a través de sitios web o aplicaciones móviles. Esto brinda a los clientes la flexibilidad de elegir sus fechas de estancia y tipos de habitación desde la comodidad de sus dispositivos.
- **Días de Reservación:** Los huéspedes pueden seleccionar las fechas exactas en las que desean alojarse, y el sistema verifica automáticamente la disponibilidad de habitaciones para esas fechas.
- **Habitaciones Disponibles:** El sistema proporciona información en tiempo real sobre las habitaciones disponibles, lo que ayuda a los huéspedes a tomar decisiones informadas.
- **Tipos de Habitaciones:** Los huéspedes pueden elegir entre una variedad de tipos de habitaciones, como estándar, suite, habitaciones con vistas panorámicas, etc.
- **Clasificación de Habitación:** Las habitaciones pueden estar clasificadas según su nivel de lujo, comodidades y precio, lo que permite a los huéspedes seleccionar una opción que se adapte a sus preferencias y presupuesto.

Control de Personal:

La gestión de recursos humanos es fundamental para el funcionamiento eficaz de un hotel. El sistema incluye un módulo de control de personal que abarca los siguientes aspectos:

- **Horarios de Entradas/Salidas:** Registra los horarios de entrada y salida del personal, lo que facilita la gestión de turnos y la planificación de recursos.
- **Tipo de Empleado:** Clasifica a los empleados según su función y responsabilidades, lo que ayuda en la asignación de tareas y responsabilidades específicas.
- **Itinerario:** Este módulo permite a los empleados consultar su itinerario de trabajo, lo que les brinda información sobre sus próximos turnos y responsabilidades.
- **CRUD (Crear, Leer, Actualizar y Eliminar) para Administración:** Los administradores tienen acceso a las herramientas necesarias para gestionar el personal de manera eficiente. Pueden agregar nuevos empleados, actualizar información existente, leer detalles de empleados y eliminar registros según sea necesario.

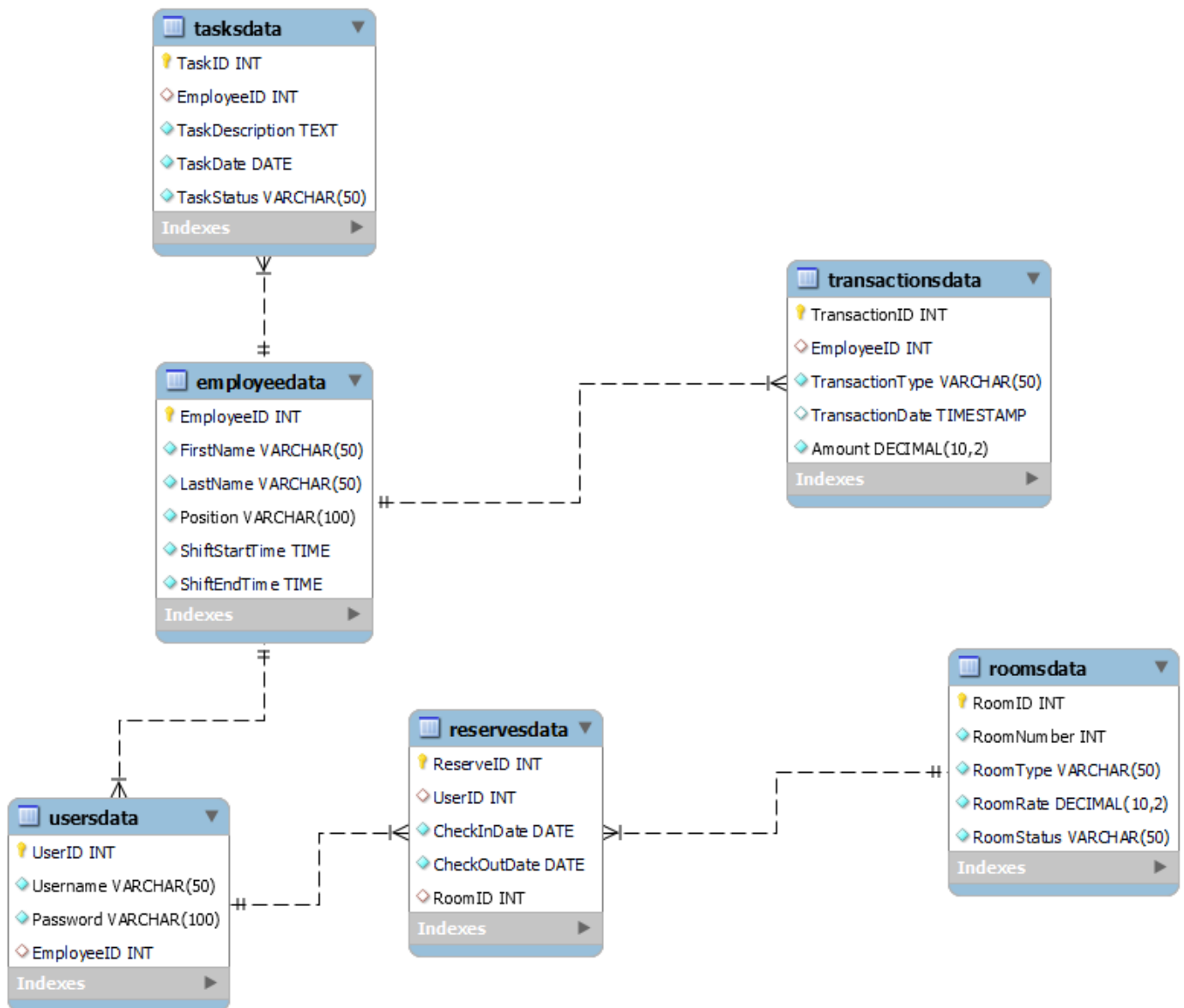
Admin:

El sistema de hotelería también cuenta con una funcionalidad de administración que incluye:

- **Reportes para Administradores:** Los administradores pueden acceder a una variedad de informes que les proporcionan información crucial para la toma de decisiones. Estos informes pueden incluir datos sobre ocupación, ingresos, reservaciones y más.
- **Reportes por Fechas:** Los administradores pueden generar informes específicos según las fechas de interés, lo que les permite evaluar el rendimiento del hotel en períodos específicos.
- **Gráficos de Ganancias:** Los gráficos de ganancias brindan una representación visual de los ingresos del hotel a lo largo del tiempo, lo que facilita la identificación de tendencias y patrones.
- **Afluencia de Personas en Gráficas:** La afluencia de personas se representa en gráficos para ayudar a los administradores a comprender las fluctuaciones en la ocupación del hotel a lo largo del año.

En resumen, el sistema de hotelería adopta un enfoque completo para gestionar todos los aspectos críticos de la operación de un hotel. Desde la gestión de reservaciones y el control de personal hasta la administración de informes y análisis, el sistema se esfuerza por garantizar la eficiencia y la satisfacción de los huéspedes, al tiempo que brinda herramientas valiosas para la toma de decisiones estratégicas por parte de los administradores.

Diagrama Relacional Base de Datos:



Casos de Uso:

Caso de uso principal

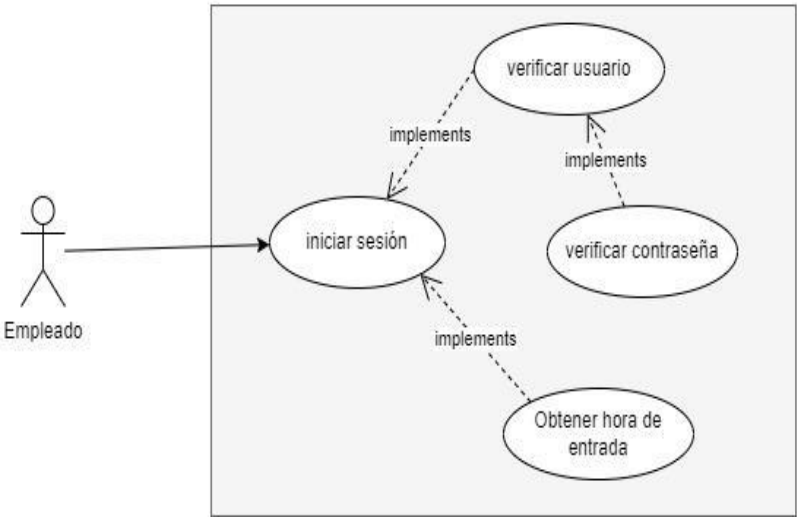
Caso de uso	Iniciar sesión
Descripción	El empleado ingresa sus credenciales para poder ingresar a la aplicación
Entidades	Empleado

Casos de uso incluidos

Caso de uso	Verificar usuario
Descripción	El sistema debe verificar que el usuario ingresado exista en la base de datos.
Entidades	

Caso de uso	Verificar contraseña
Descripción	Después de haber confirmado la existencia del usuario se debe verificar que la contraseña ingresada sea la misma.
Entidades	

Caso de uso	Obtener hora de entrada
Descripción	El sistema debe almacenar la hora a la que el usuario ingresó por primera vez al sistema durante el día
Entidades	

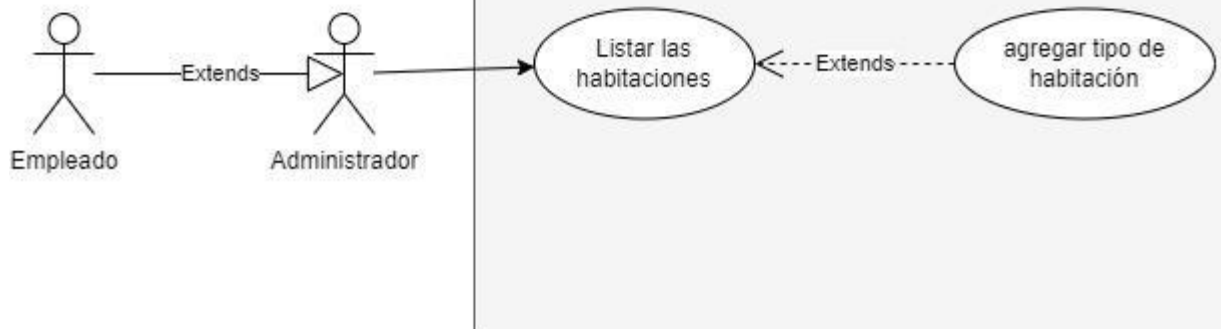


Caso de uso principal

Caso de uso	Listar las habitaciones
Descripción	El administrador puede listar todas las habitaciones agregadas al sistema
Entidades	Administrador

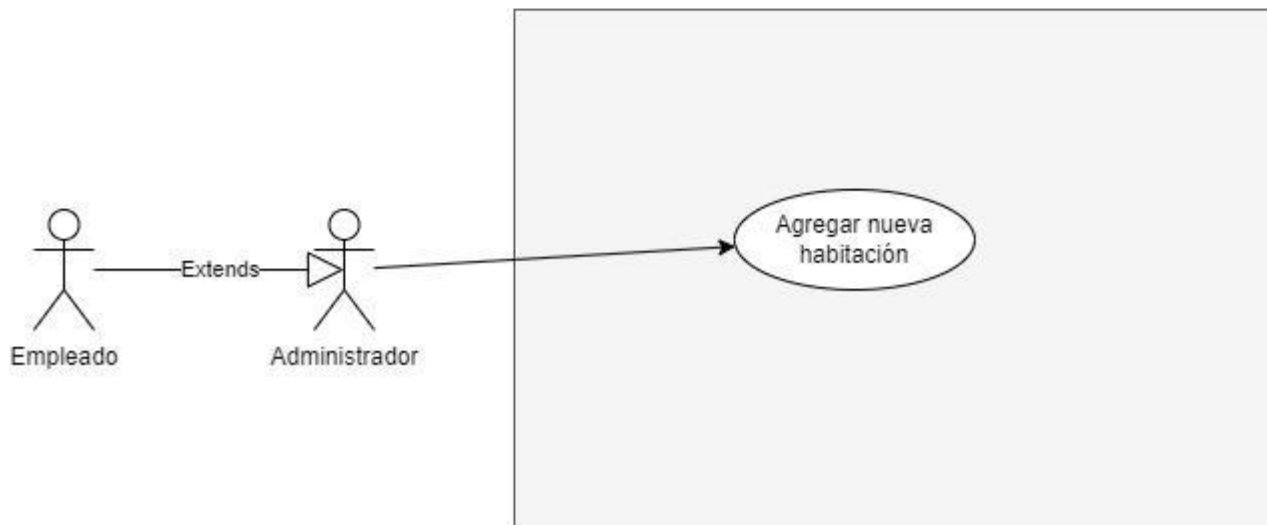
Caso de uso extendido

Caso de uso	Agregar tipo de habitación
Descripción	El administrador puede agregar un filtro en base al tipo de habitación que desea obtener
Entidades	Administrador



Caso de uso principal

Caso de uso	Agregar nueva habitación
Descripción	El administrador agrega habitaciones al sistema
Entidades	Administrador

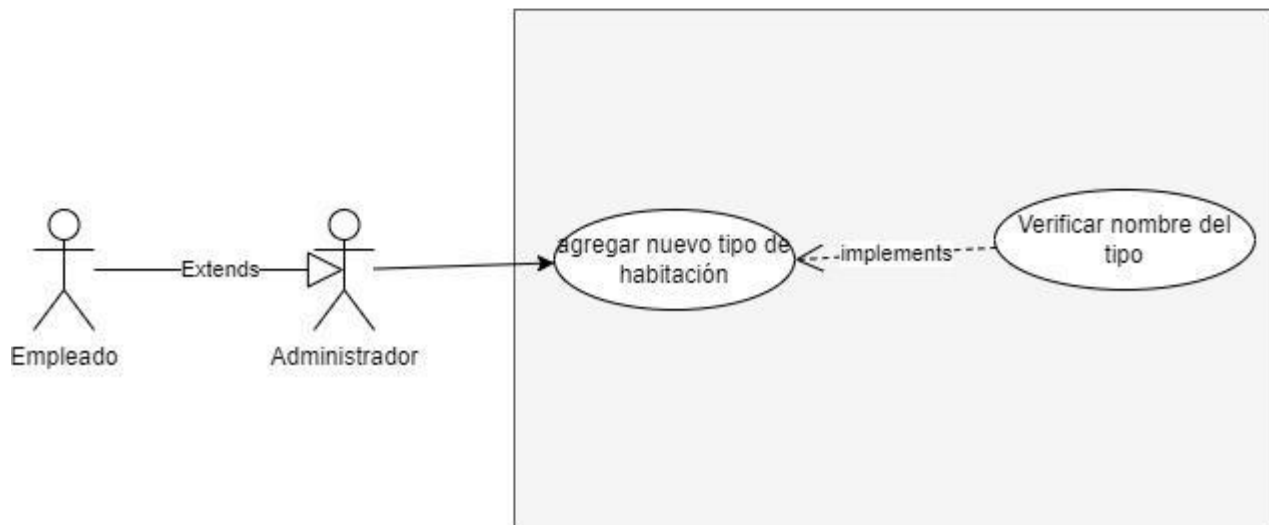


Caso de uso principal

Caso de uso	Agregar nuevo tipo de habitación
Descripción	El administrador puede agregar nuevos tipos de habitación al sistema
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Verificar nombre del tipo
Descripción	El sistema debe confirmar que el nombre del nuevo tipo de habitación sea único y no se encuentre ya ingresado al sistema
Entidades	



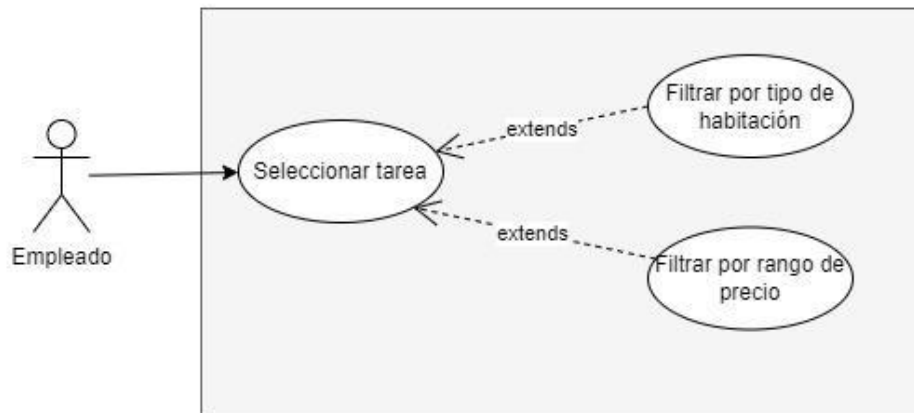
Caso de uso principal

Caso de uso	Listar Habitaciones
Descripción	El usuario es capaz de ver todas las habitaciones existentes en el sistema
Entidades	Usuario

Casos de uso extendidos

Caso de uso	filtrar por tipo de habitación
Descripción	El usuario es capaz de filtrar las habitaciones por tipo de habitación
Entidades	Usuario

Caso de uso	filtrar por rango de precio
Descripción	El usuario es capaz de filtrar las habitaciones por rango de precio
Entidades	Usuario



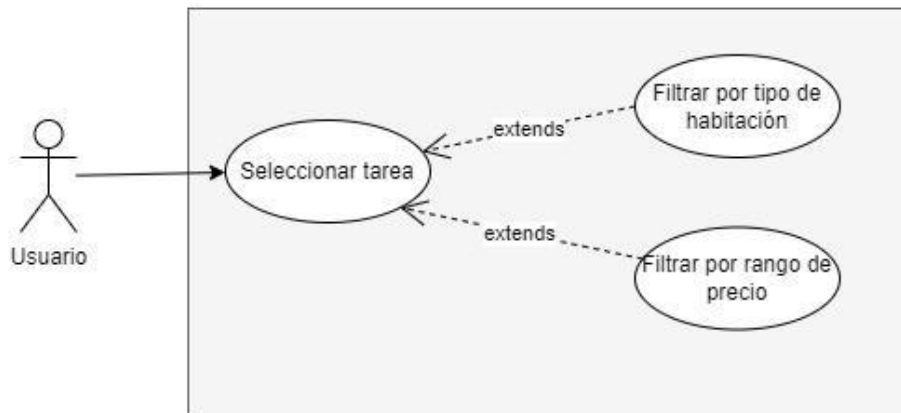
Caso de uso principal

Caso de uso	Ver información de reservación
Descripción	El usuario es capaz de ver la información sobre la reservación hecha en el sistema en base a algún código
Entidades	Usuario

Casos de uso extendidos

Caso de uso	filtrar por tipo de habitación
Descripción	El usuario es capaz de filtrar las habitaciones por tipo de habitación
Entidades	

Caso de uso	filtrar por rango de precio
Descripción	El usuario es capaz de filtrar las habitaciones por rango de precio
Entidades	



Caso de uso principal

Caso de uso	Finalizar tarea
Descripción	Una vez completada la tarea el empleado marca su tarea asignada como FINALIZADA
Entidades	Empleado

Caso de uso implementado

Caso de uso	Cambiar estado de tarea
Descripción	El sistema cambia el estado de la tarea de ASIGNADA a FINALIZADA.
Entidades	



Caso de uso principal

Caso de uso	Seleccionar tarea
Descripción	Los empleados listan las tareas pendientes por realizar
Entidades	Empleado

Caso de uso implementado

Caso de uso	Cambiar estado de tarea
Descripción	El sistema cambia el estado de la tarea de 'Pendiente' a 'ASIGNADA' también agrega el código del empleado que realizará la tarea
Entidades	

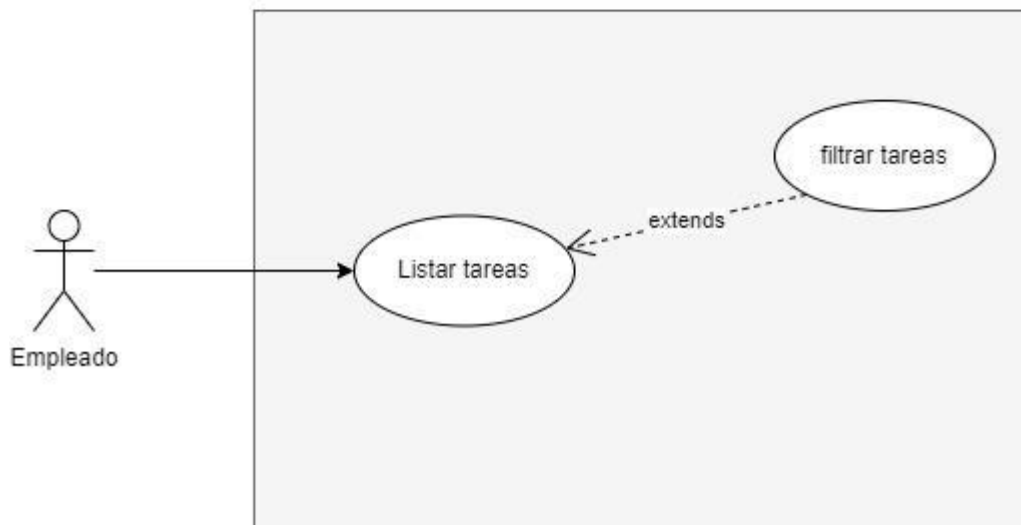


Caso de uso principal

Caso de uso	Listar tareas
Descripción	Los empleados listan las tareas que el administrador ha asignado durante el día
Entidades	empleado

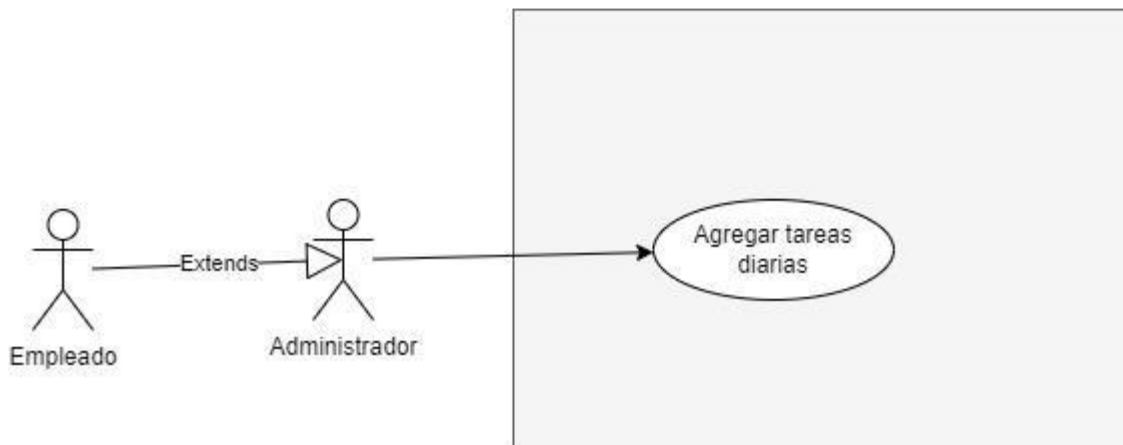
Caso de uso extendido

Caso de uso	Filtrar tareas
Descripción	El empleado es capaz de filtrar la tarea, puede ser por: PENDIENTE o ASIGNADA
Entidades	empleado



Caso de uso principal

Caso de uso	Agregar tareas diarias
Descripción	El administrador puede agregar tareas que serán realizadas por los demás empleados
Entidades	Administrador

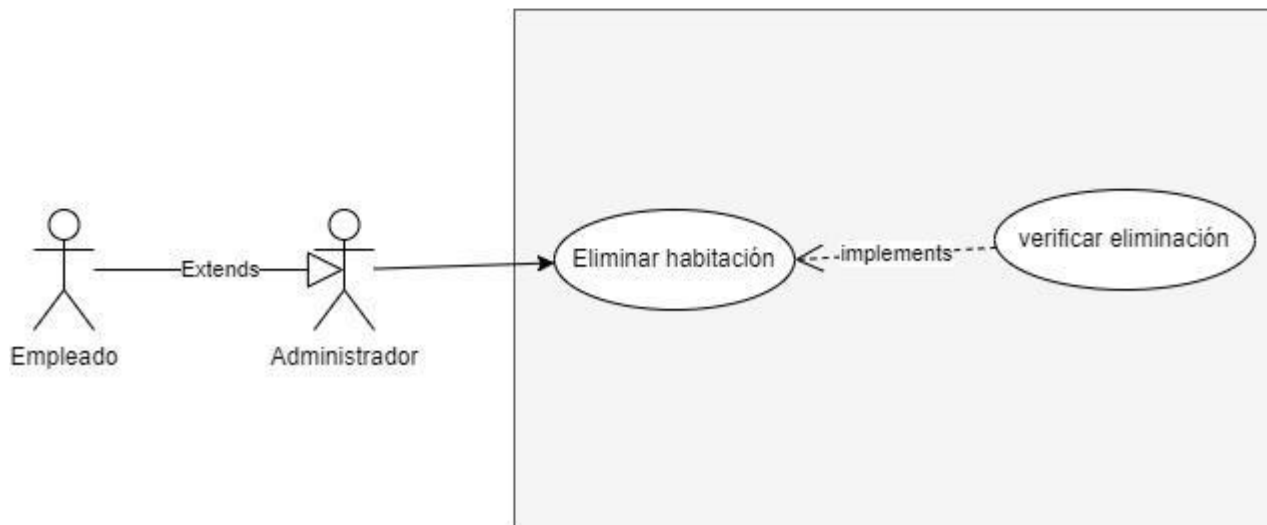


Caso de uso principal

Caso de uso	Eliminar habitación
Descripción	El administrador puede eliminar una habitación registrada anteriormente
Entidades	Administrador

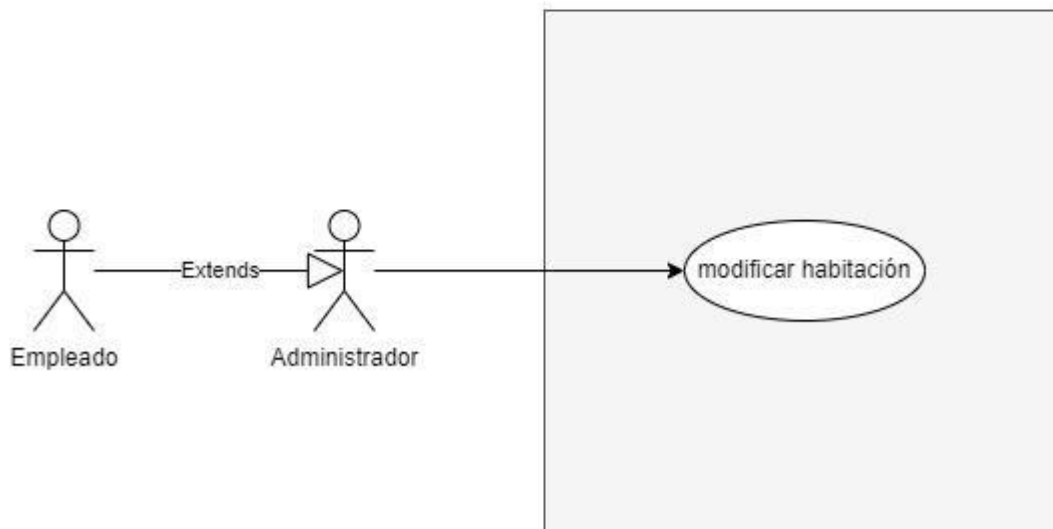
Casos de uso incluidos

Caso de uso	Verificar eliminación
Descripción	El sistema debe confirmar que la habitación a eliminar del sistema no se encuentre ocupada con alguna reserva vigente.
Entidades	



Caso de uso principal

Caso de uso	modificar habitación
Descripción	El administrador puede modificar la información sobre alguna habitación ingresada anteriormente
Entidades	Administrador

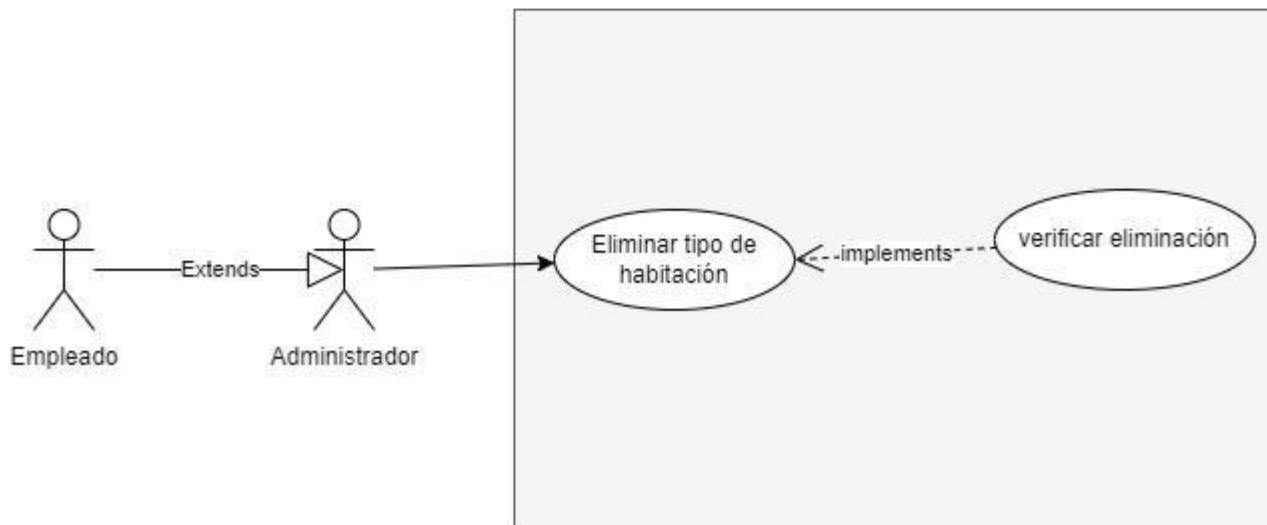


Caso de uso principal

Caso de uso	Eliminar tipo de habitación
Descripción	El administrador puede Eliminar algún tipo de habitación que ya se encuentre registrado en el sistema.
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Verificar eliminación
Descripción	El sistema debe verificar que la eliminación sea posible. Para esto debe corroborar que este tipo de habitación no se encuentre asignado a una habitación registrada.
Entidades	

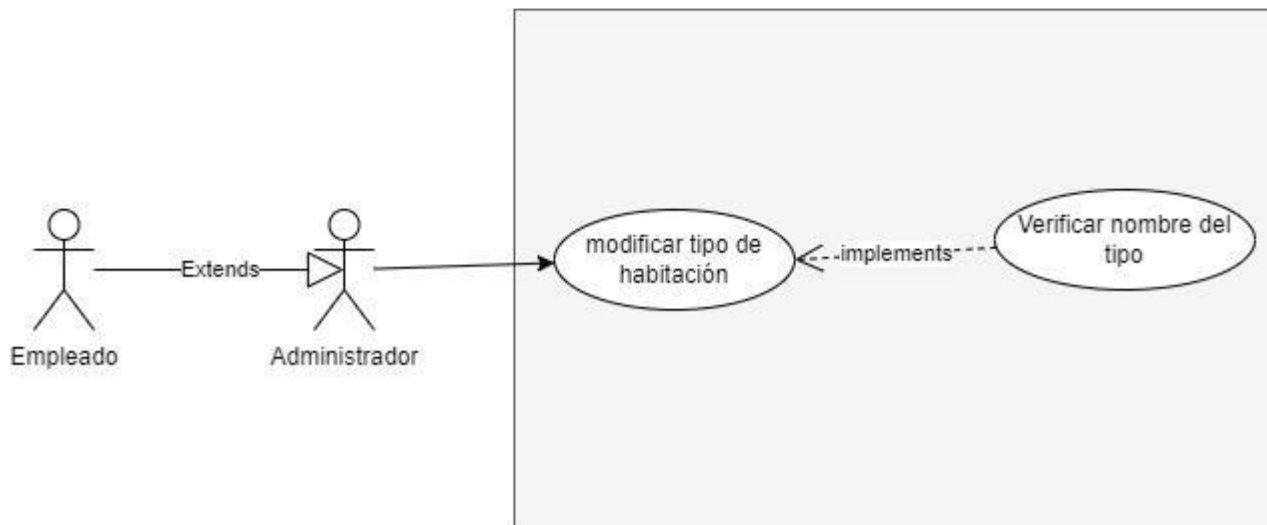


Caso de uso principal

Caso de uso	Modificar tipo de habitación
Descripción	El administrador puede modificar algún tipo de habitación que ya se encuentre registrado en el sistema.
Entidades	Administrador

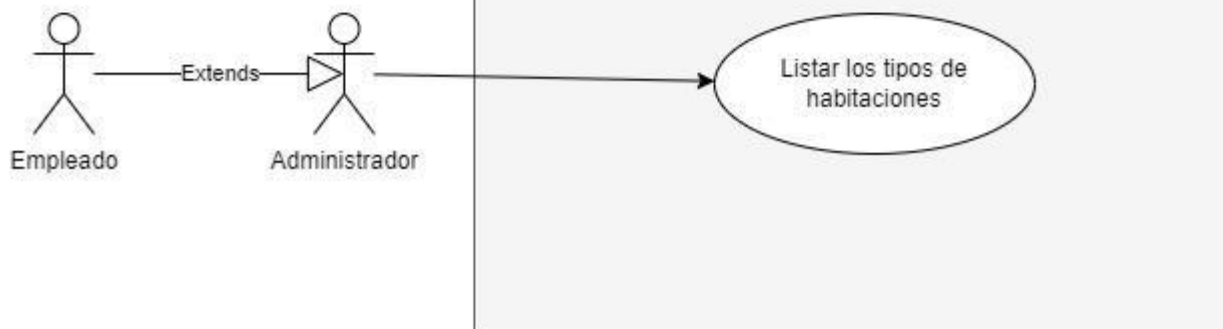
Casos de uso incluidos

Caso de uso	Verificar nombre del tipo
Descripción	El sistema debe confirmar que el nombre del nuevo tipo de habitación sea único y no se encuentre ya ingresado al sistema
Entidades	



Caso de uso principal

Caso de uso	Listar los tipos de habitaciones
Descripción	El administrador puede ver el listado de todos los tipos de habitaciones ingresadas anteriormente al sistema
Entidades	Administrador

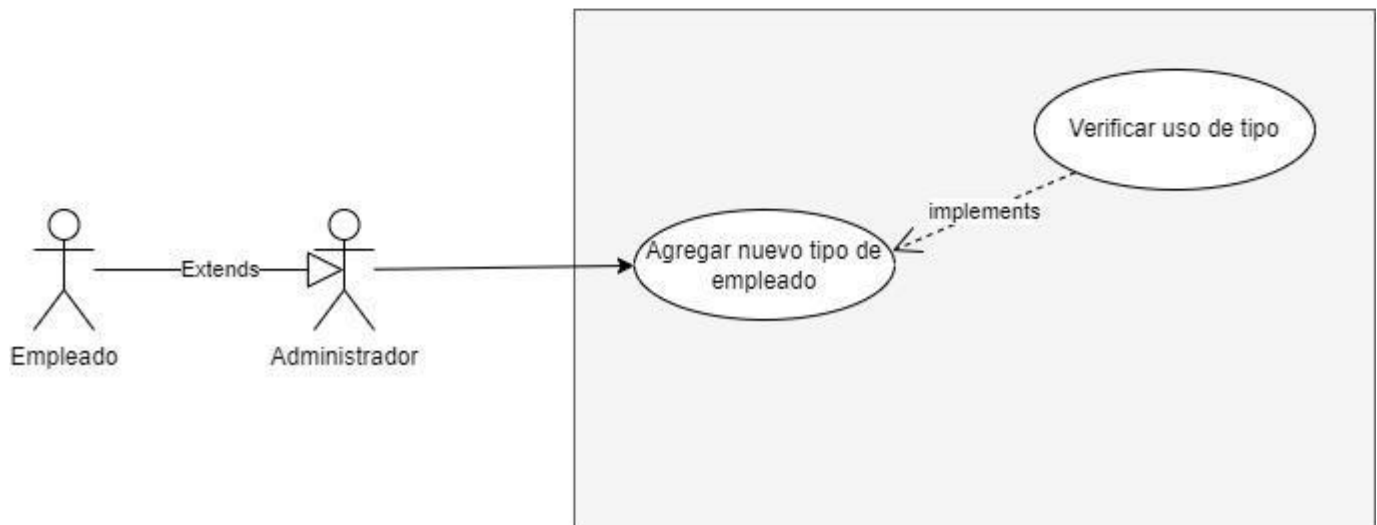


Caso de uso principal

Caso de uso	Eliminar tipo de empleado
Descripción	El administrador puede directamente eliminar un tipo de empleado en caso de que este ya no se necesite en la aplicación
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Verificar uso de tipo
Descripción	El sistema debe confirmar que ningún empleado registrado y activo se encuentre usando ese tipo de empleado actualmente
Entidades	

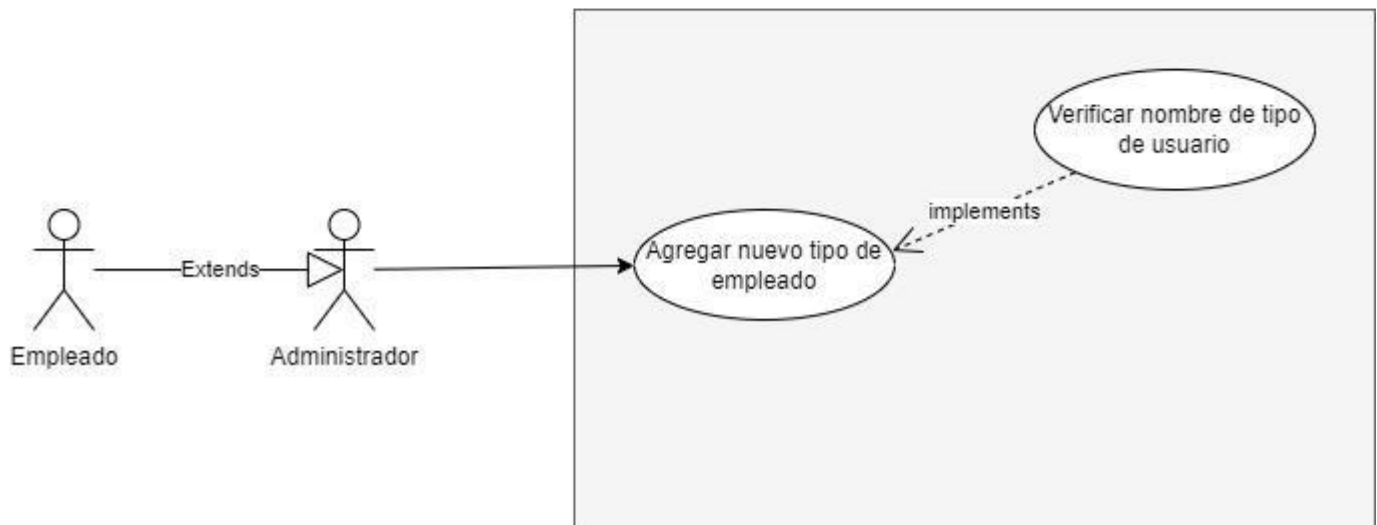


Caso de uso principal

Caso de uso	Actualizar tipo de empleado
Descripción	El administrador modifica el nombre de un tipo de empleado registrado anteriormente en caso de que haya existido un error al momento de registrarlo
Entidades	Administrador

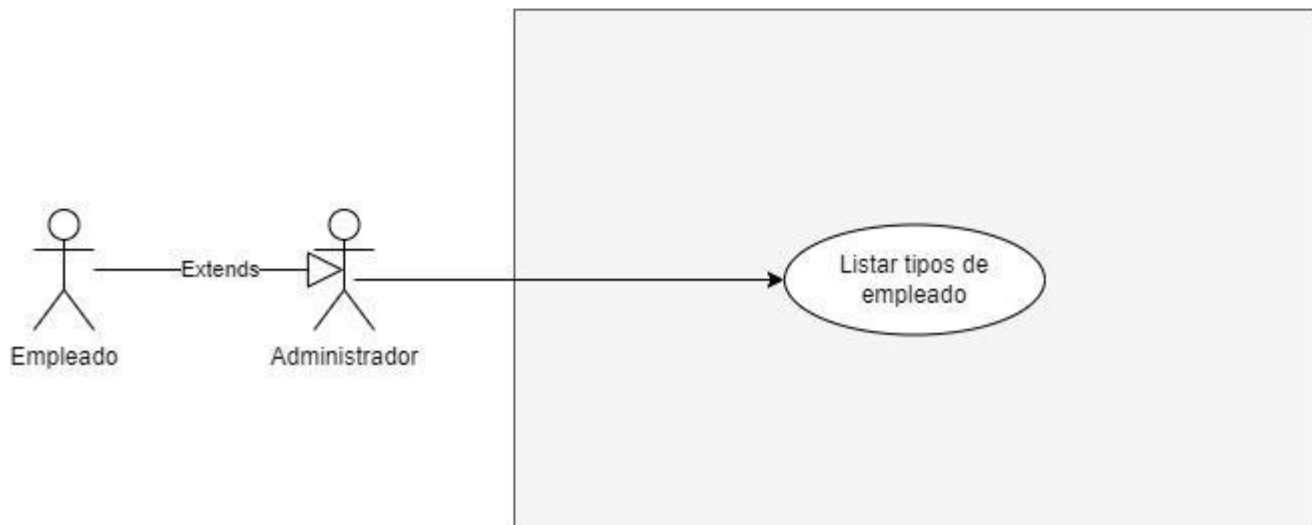
Casos de uso incluidos

Caso de uso	Verificar nombre de tipo de usuario
Descripción	El sistema debe verificar que la actualización de nombre se pueda realizar ya que este es un campo único
Entidades	



Caso de uso principal

Caso de uso	Listar tipos de empleado
Descripción	El administrador puede obtener una lista de todos los tipos de empleado registrados en el sistema
Entidades	Administrador

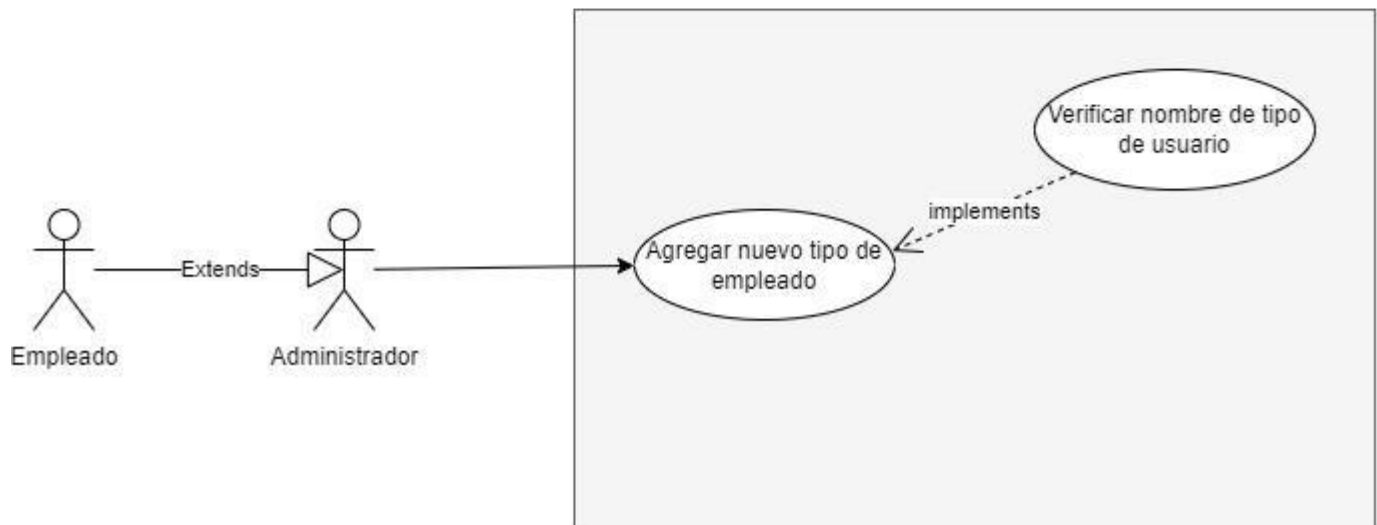


Caso de uso principal

Caso de uso	Agregar nuevo tipo de empleado
Descripción	El administrador agrega un nuevo tipo de empleado al sistema.
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Verificar nombre de tipo de usuario
Descripción	El sistema debe confirmar que el nombre del nuevo tipo sea único para evitar ambigüedades.
Entidades	



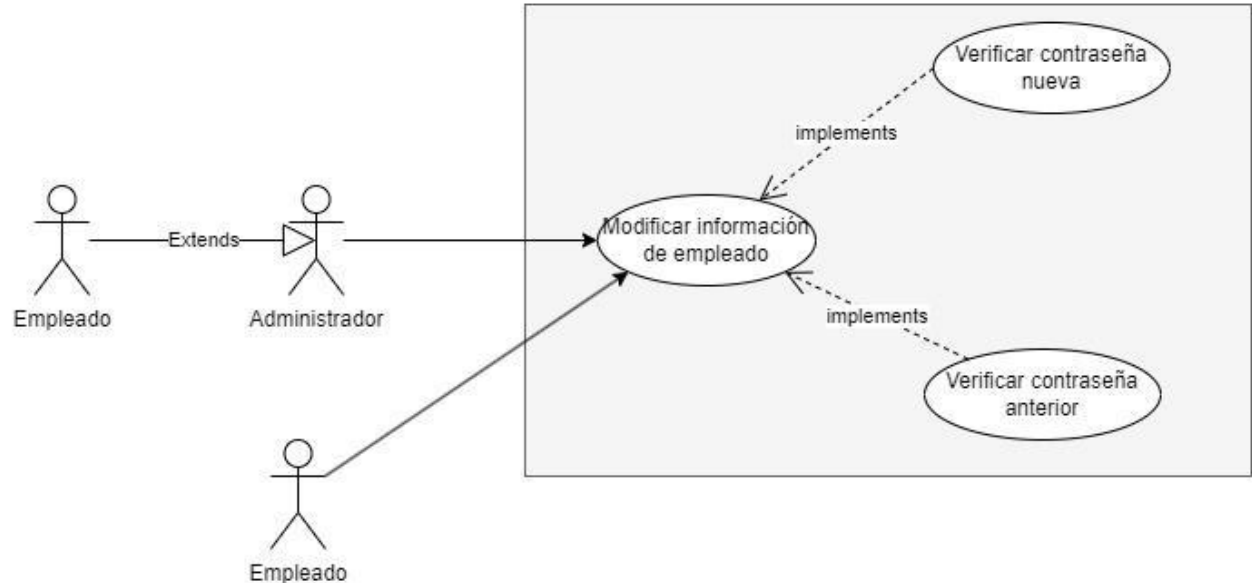
Caso de uso principal

Caso de uso	Modificar información de empleado
Descripción	El administrador puede modificar cosas como el nombre del empleado o la contraseña. Si se desea modificar la contraseña el empleado debe ingresar la contraseña anterior seguido de la nueva contraseña
Entidades	Administrador, Empleado

Casos de uso incluidos

Caso de uso	Verificar contraseña nueva
Descripción	El sistema debe confirmar que la información ingresada en los campos "contraseña" y "confirmar contraseña" coincidan.
Entidades	

Caso de uso	Verificar contraseña anterior
Descripción	El sistema debe confirmar que la contraseña anterior sea la correcta para poder hacer el cambio a la nueva.
Entidades	

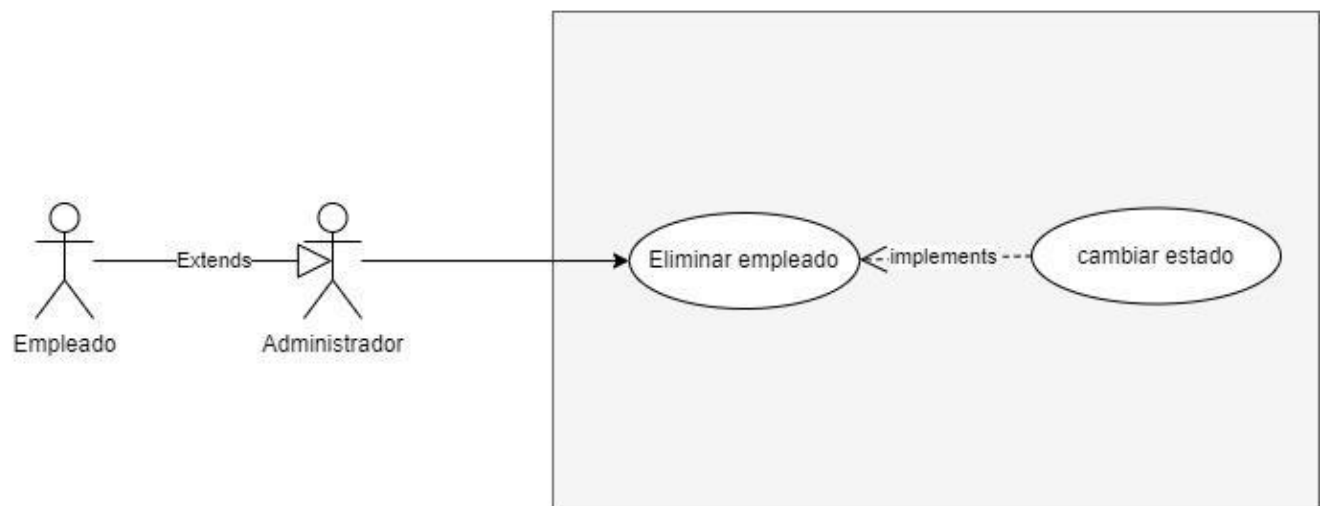


Caso de uso principal

Caso de uso	Eliminar empleado
Descripción	El administrador elimina a un empleado.
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Cambiar estado
Descripción	El sistema cambia el estado del emplado a "inactivo"
Entidades	



Caso de uso principal

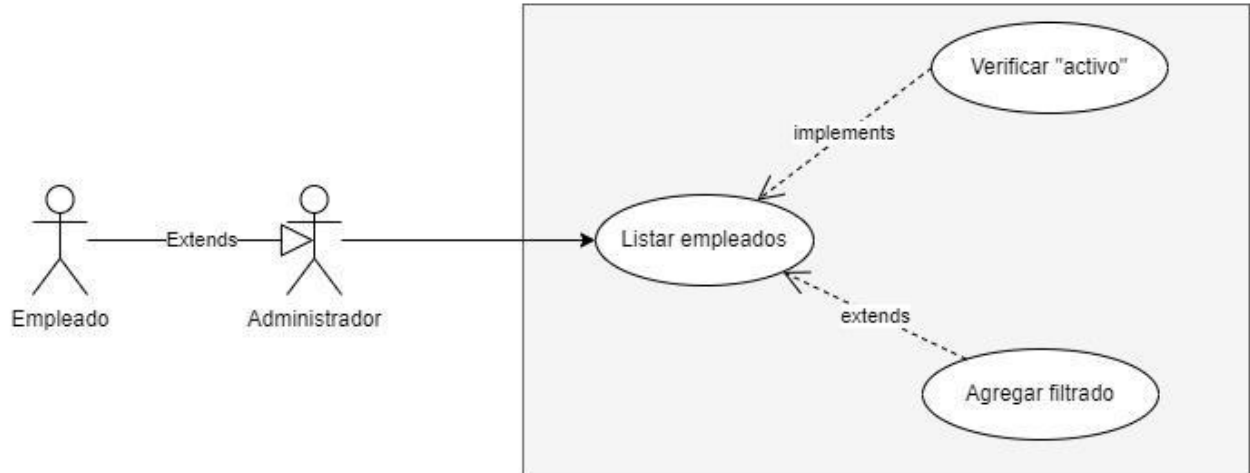
Caso de uso	Listar empleados
Descripción	El administrador lista todos los usuarios registrados en el sistema
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Verificar "activo"
Descripción	El sistema obtiene todos los empleados registrados en el sistema que se encuentren "activos"
Entidades	

Casos de uso incluidos

Caso de uso	Agregar filtro
Descripción	El administrador puede filtrar los usuarios en base al tipo de empleado.
Entidades	



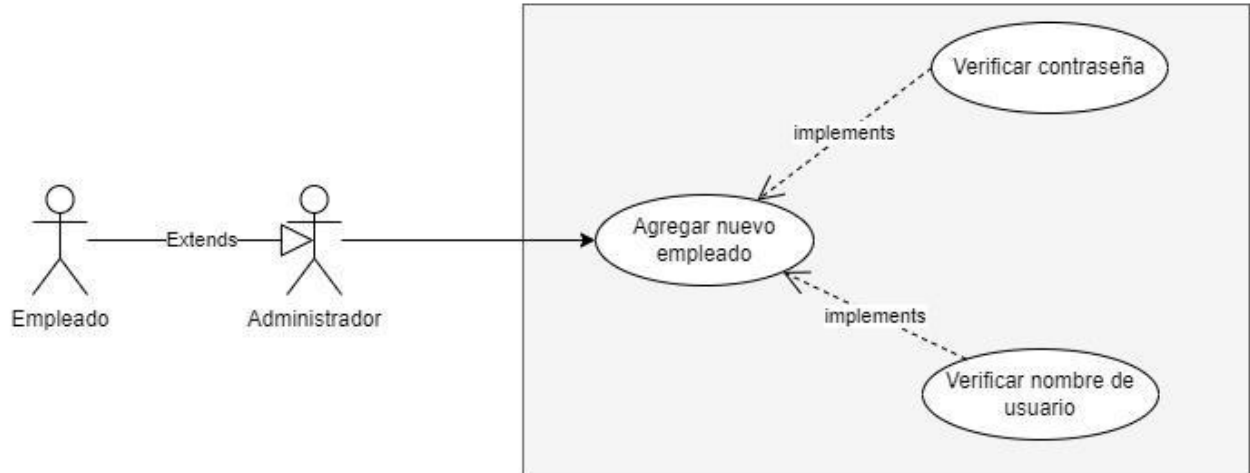
Caso de uso principal

Caso de uso	Agregar nuevo empleado
Descripción	El administrador registra un nuevo empleado al sistema ingresando toda la información necesaria como: nombre del empleado, nombre de usuario, contraseña, tipo de empleado
Entidades	Administrador

Casos de uso incluidos

Caso de uso	Verificar contraseña
Descripción	El sistema debe confirmar que la información ingresada en los campos "contraseña" y "confirmar contraseña" coincidan.
Entidades	

Caso de uso	Verificar nombre de usuario
Descripción	El sistema debe confirmar que el usuario ingresado se encuentre disponible ya que este debe ser único para evitar ambigüedades.
Entidades	



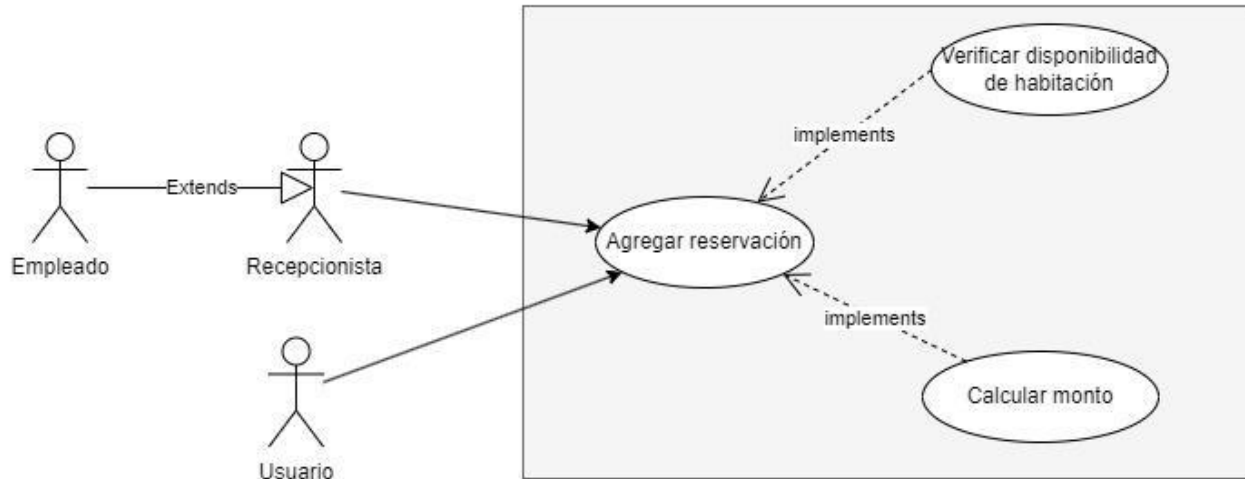
Caso de uso principal

Caso de uso	Agregar reservación
Descripción	La recepcionista o usuario (en caso de reservación en línea) registra una nueva reservación al sistema
Entidades	Recepcionista, usuario

Casos de uso incluidos

Caso de uso	Verificar disponibilidad de habitación
Descripción	El sistema debe verificar la disponibilidad del tipo de habitación que el empleado desee
Entidades	

Caso de uso	Calcular monto
Descripción	El sistema debe calcular el monto total en base a la habitación seleccionada y el número de días que el huésped reservará la habitación.
Entidades	



Diagramas Completos:

https://app.diagrams.net/#G10wYx6_hel6sIrfudJwC5FpRNfa6rypKB

Diagrama de Despliegue:

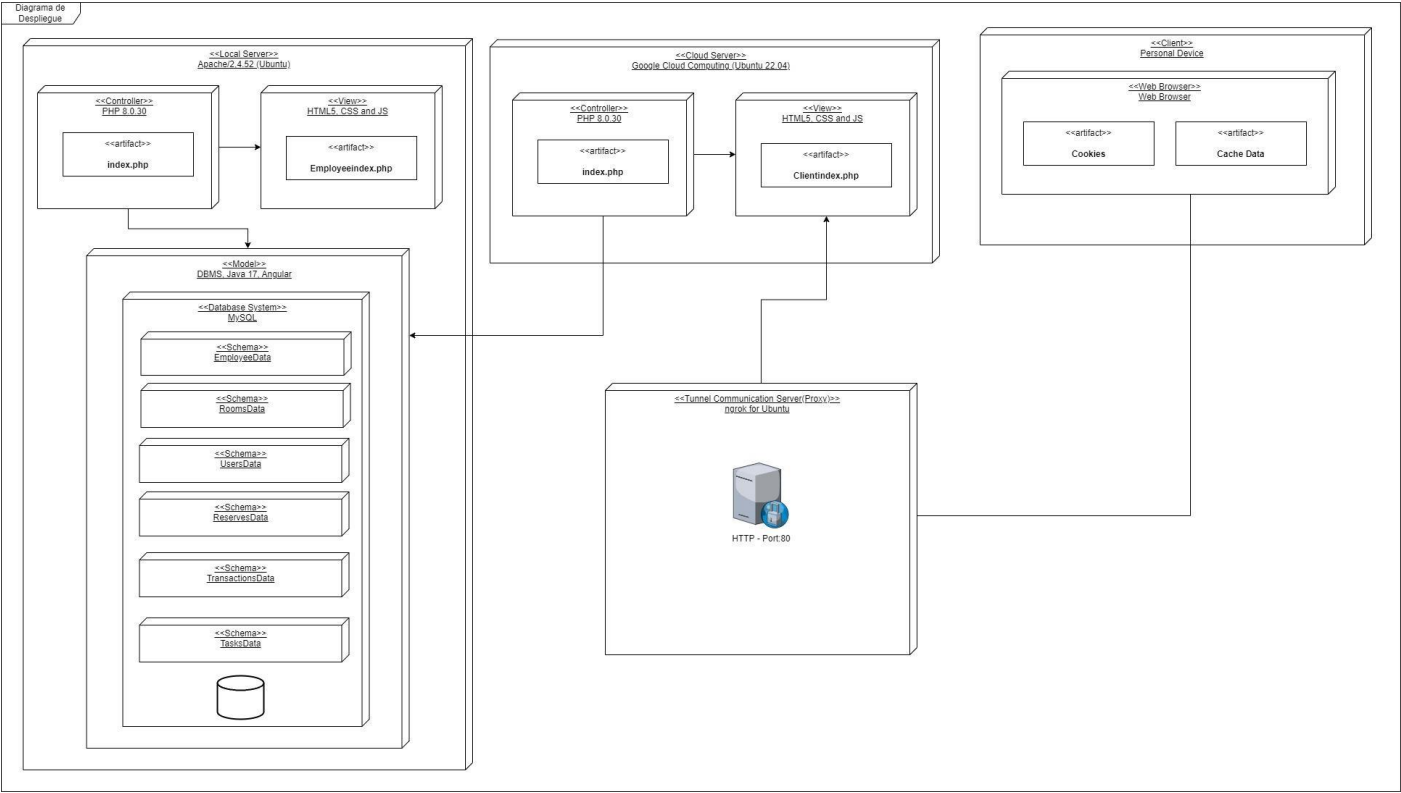


Diagrama:

https://app.diagrams.net/#G1HiHVAmjFF6Pd_VKOhQJPoAv4crxGk-UD

Diagrama de Componentes (UsersWeb):

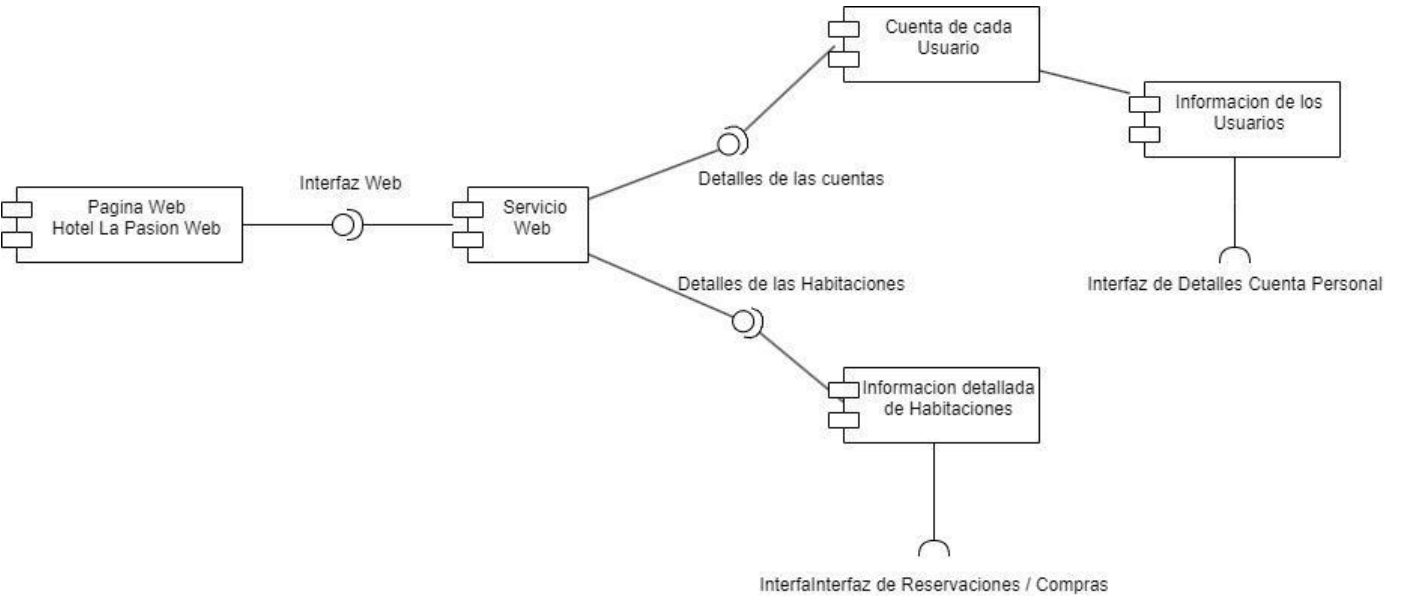


Diagrama de Componentes (AdminWeb):

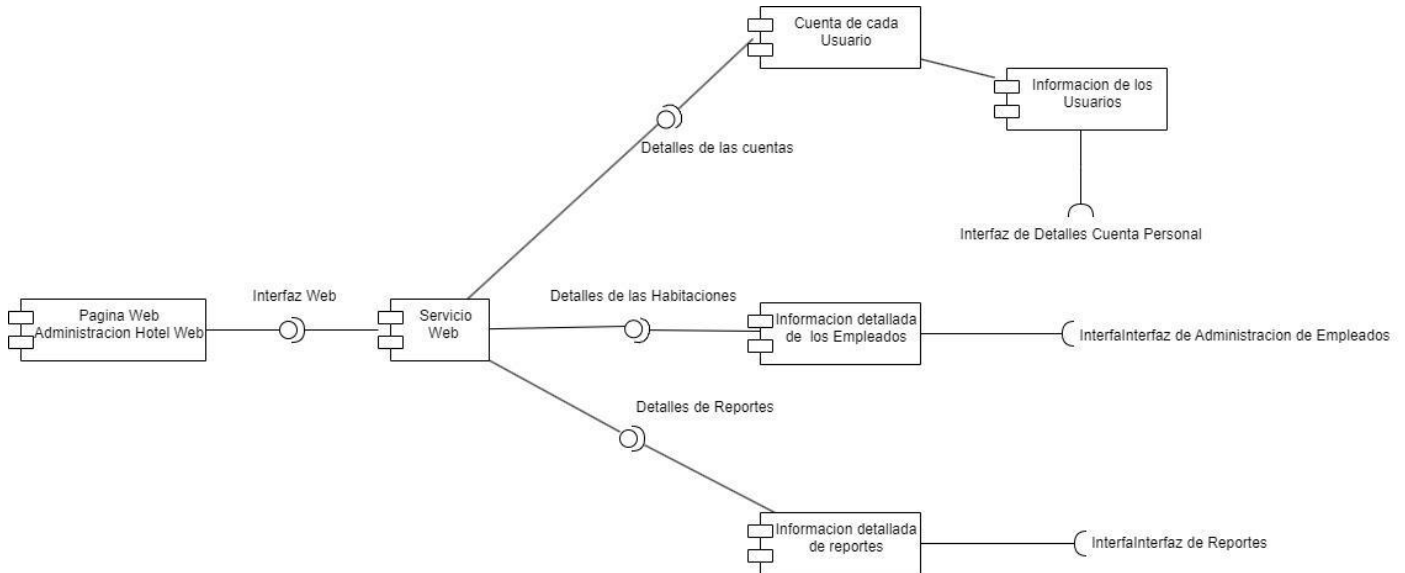
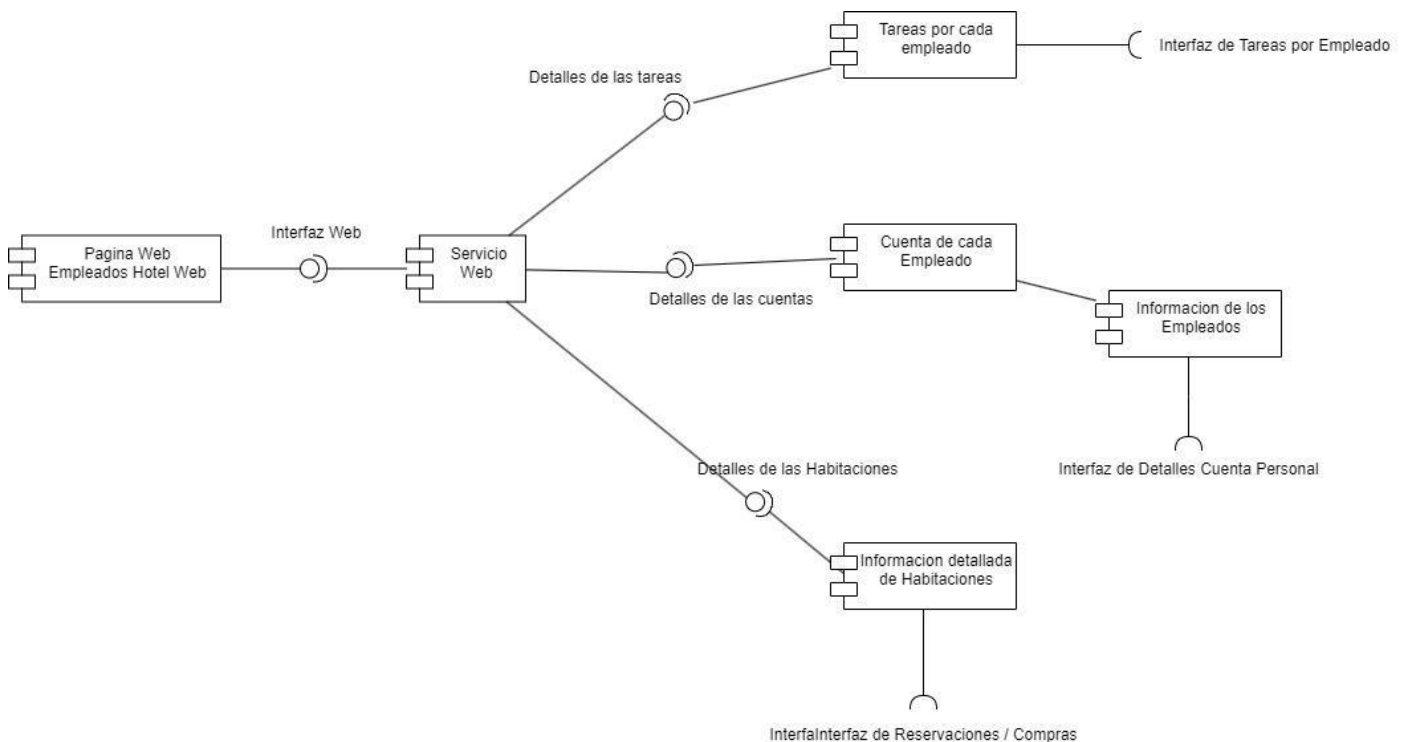


Diagrama de Componentes (EmployeeWeb):



Diagramas:

https://app.diagrams.net/#G1HiHVAmjFF6Pd_VKOhQJPoAv4crxGk-UD#%7B%22pageld%22%3A%22RzWk7jUkcD9xkociXsJE%22%7D

https://app.diagrams.net/#G1HiHVAmjFF6Pd_VKOhQJPoAv4crxGk-UD#%7B%22pageld%22%3A%22_wnwLJzLY11zwsI_h76T%22%7D

https://app.diagrams.net/#G1HiHVAmjFF6Pd_VKOhQJPoAv4crxGk-UD#%7B%22pageld%22%3A%22uOmQKqz_6JZfp2VdKNX9%22%7D

Script Base de Datos:

```
CREATE DATABASE hotel;

/* \c hotel; Para usar base de datos hotel */

CREATE TABLE EmployeeRole(
    id SERIAL,
    name VARCHAR(25) NOT NULL,
    PRIMARY KEY(id)
);

CREATE TABLE UserRole(
    id SERIAL,
    name VARCHAR(25) NOT NULL,
    PRIMARY KEY(id)
);

CREATE TABLE Employee(
    id SERIAL,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    employee_role_id bigint NOT NULL,
    shift_start_time TIME NOT NULL,
    shift_end_time TIME NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(employee_role_id) REFERENCES EmployeeRole(id)
);

CREATE TABLE Users(
    id SERIAL,
    username VARCHAR(50) NOT NULL,
    email VARCHAR(50) NOT NULL,
    password VARCHAR(100) NOT NULL,
    employee_id bigint,
    user_role_id bigint NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(employee_id) REFERENCES Employee(id),
    FOREIGN KEY(user_role_id) REFERENCES UserRole(id)
);
```

```
CREATE TABLE Task(  
    id SERIAL,  
    employee_id bigint NOT NULL,  
    task_description VARCHAR(255) NOT NULL,  
    task_date timestamp NOT NULL,  
    task_completed boolean NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(employee_id) REFERENCES Employee(id)  
);  
  
CREATE TABLE RoomType(  
    id SERIAL,  
    name VARCHAR(100) NOT NULL,  
    price Decimal(10,2) NOT NULL,  
    PRIMARY KEY(id)  
);  
  
CREATE TABLE Room(  
    id SERIAL,  
    rate DECIMAL(3, 2) NOT NULL,  
    available boolean NOT NULL,  
    room_type bigint NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(room_type) REFERENCES RoomType(id)  
);  
  
CREATE TABLE TransactionType(  
    id SERIAL,  
    name VARCHAR(50) NOT NULL,  
    PRIMARY KEY(id)  
);  
  
CREATE TABLE Transaction(  
    id SERIAL,  
    employee_id bigint NOT NULL,  
    transaction_type bigint NOT NULL,  
    transaction_date timestamp NOT NULL,  
    amount Decimal(10, 2) NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(employee_id) REFERENCES Employee(id),  
    FOREIGN KEY(transaction_type) REFERENCES TransactionType(id)  
);
```

```
CREATE TABLE Reservation(  
    id SERIAL,  
    check_in_date timestamp NOT NULL,  
    check_out_date timestamp NOT NULL,  
    created_by_customer boolean NOT NULL,  
    transaction_id bigint NOT NULL,  
    employee_id bigint,  
    user_id bigint NOT NULL,  
    room_id bigint NOT NULL,  
    PRIMARY KEY(id),  
    FOREIGN KEY(employee_id) REFERENCES Employee(id),  
    FOREIGN KEY(user_id) REFERENCES Users(id),  
    FOREIGN KEY(room_id) REFERENCES Room(id),  
    FOREIGN KEY(transaction_id) REFERENCES Transaction(id)  
);
```