****

**CENTRO EDUCATIVO TECNICO LABORAL KINAL**

5TO. PERITO EN INFORMATICA

Peace at home

Alumno

**Kenet Efraín Cuyuch Joj**

Carné 2024-177

Guatemala, 15 de Julio de 2025.

|  |  |
| --- | --- |
| **Nombre Completo** | **Puesto** |
| Kenet Efraín Cuyuch Joj | Scrum master |
| Ricardo Alejandro Hernández Hernandez | Programador |
| Jose Enrique cuc cutz | Programador |
| Angel Gabriel Ernesto Grijalva Castro | Programador |
| Angel Gahel Padilla Pacheco | Programador |
| Osman Emmanuell Veliz Guzmán | Programador |
| Carlos Alejandro Patal Choc | Programador |
| Carlos Enrique López Quino | Programador |
| Benjamin Eli Argueta Caal | Programador |
| Jeferson André Cano López | Programador |
| Franklin Stev Paiz Chávez | Programador |
|  |  |

**Historias de Usuario**

**Tupla : Reserva**

HUCP-001 Realizar Nueva Reserva

Como: Un invitado

Quiero: Seleccionar una propiedad, elegir mis fechas de inicio y fin, y confirmar una reserva

Para: Asegurar mi estancia en la propiedad deseada.

HUCP-002 Gestionar Mis Reservas

Como: Un invitado o un anfitrión

Quiero: Ver el estado de mis reservas (confirmado, cancelado, completado) y tener la opción de modificar o cancelar una reserva si es necesario

Para: Tener control y visibilidad sobre mis estancias o la ocupación de mi propiedad.

HUCP-003 Consultar Disponibilidad y Precios de Propiedades

Como: Un invitado

Quiero: Buscar propiedades y ver su precio por día y disponibilidad en un rango de fechas

Para: Encontrar la propiedad que mejor se adapte a mis necesidades y presupuesto para mis fechas de viaje.

**Tupla: Empleados**

**HUFP-004 Lista de los empleados**

**Como:** Administrador de recursos humanos

**Quiero:** Poder ver una lista de empleados activos

**Para:** Poder gestionar de manera eficiente los recursos y asegurarme de que solo los empleados vigentes están siendo considerados en los procesos internos.

**HUFP-005: Registro de nuevos empleados**

**Como:** Responsable de contratación

**Quiero:** Poder agregar nuevos empleados con su información (nombre, apellido, correo, teléfono, cargo, fecha de contratación)

**Para:** Mantener la base de datos actualizada y asegurarme de que se realice un registro adecuado y completo de todos los empleados nuevos.

**HUFP-006: Actualización de la información del empleado**

**Como:** Gerente de departamento

**Quiero:** Poder actualizar la información de los empleados (cargo, teléfono, correo)

**Para**: Asegurarme de que la base de datos siempre refleje la información más actual y optimizar la comunicación dentro del equipo.

**Tupla: Pago**

HURH-007 Registro de pago:

1. Como un registro de pagos

2. Quiero que registre los datos del monto

3. Para poder confirmar que los datos del monto estén registrados

HURH-008 Validación de monto máximo

1. Como método de pago

2. Quiero que el monto sea en efectivo o en tarjeta

3. Para verificar si el monto de pago es valido

HURH-009 Generación de recibo de pago

1. Como registro de pago recibido

2. Quiero que me den la información del monto de pago

3. Para que registre el recibo y se almacene en el sistema

**TUPLA:Servicios**

HUJC-010

Como: cliente

Quiero: ver los servicios con su nombre, descripción, horario disponible y costo extra,

Para: poder elegir el servicio que mejor se adapte a mis necesidades y presupuesto.

HUJC-011

Como: administrador

Quiero: poder activar, desactivar y modificar los datos de un servicio (nombre, descripción, horario, costo extra y tipo de servicio),

Para: mantener la información actualizada y ofrecer servicios vigentes a los clientes.

HUJC-012

Como: usuario del sistema (empleado),

Quiero: consultar el tipo de servicio y su horario disponible,

Para: poder informar correctamente a los clientes sobre la disponibilidad y características de cada servicio.

**Tupla: Soporte Técnico**

**HUAG-013:**

Como huésped o anfitrión de Airbnb,

quiero poder contactar al soporte técnico rápidamente a través de un chat en la plataforma,

para poder resolver problemas técnicos como errores de reserva, pagos fallidos o problemas con mi cuenta de forma eficiente.

**HUAG-014:**

Como técnico de soporte,

quiero ver una lista de tickets pendientes y asignarme uno,

para poder empezar a resolverlo de forma organizada.

**HUAG-015:**

Como administrador de soporte,

quiero poder hacer seguimiento al estado de cada ticket (pendiente, en proceso, resuelto),

para asegurarme de que los problemas se están resolviendo eficientemente.

**Tupla: Reseñas**

HUJC-016: Registo de nueva reseña

Como: Huesped

Quiero: Registar una reseña sobre los servicios del cuarto de baño

Para: Que lleven un registro de las mejoras que necesita o faltan en el baño

**HUJC-017**

Como: Empresario

Quiero: Registrar una reseña sobre la propiedad que me fue otorgada en Peace At Home

Para: Que los nuevos huéspedes sepan de las comodidades y ventajas de esta propiedad en especifico

**HUJC-018**

Como: Arrendador

Quiero: Registrar una reseña sobre los huéspedes que ocupan mi propiedad de Peace At Home

Para: Llevar un registro de cosas que pueden mejorarse en la propiedad para otorgarles una mejor comodidad

**Tupla: Propiedades**

HUCL-019: Registro de Propiedades

Como: Agente Inmobiliario

Quiero: Poder visualizar todas las propiedades registradas al nombre de algún usuario

Para: Poder llevar un registro de sus facturas y remodelaciones

HUCL-020: Registro de Nueva Propiedad

Como: Arrendador

Quiero: Registrar una nueva propiedad en Peace at Home

Para: Brindar una experiencia cómoda y agradable a los huéspedes que renten la propiedad

HUCL-021:

Como: Administrador

Quiero: Poder editar los estados y descripciones de las propiedades que se encuentren en remodelaciones o con atracciones nuevas

Para: Que los nuevos usuarios sepan de las nuevas características que poseen las propiedades

**Tupla: Reembolso**

HUOV-022

Como: empleado

Quiero: Poder enviar una solicitud de reembolso cargando recibos y completando un formulario,

para que la administración pueda revisar y aprobar el reembolso de los gastos.

Para: Para ofrecer un rembolso cada que lo necesite un cliente.

HUOV-023

Como: Administrador financiero

Quiero: Acceder a todas las solicitudes de rembolso recibidas y poder aprobarlas o rechazarlas

o hasta incluso pedir más información sobre la solicitud de rembolso.

Para: Tener un registro de cada acción tomada con fecha y el usuario responsable.

HUOV-024

Como: Cliente

Quiero: Recibir notificaciones cada vez que cambie el estado de mi solicitud de reembolso,

para mantenerme informado sobre el avance y resultado de mi trámite.

Para: Tener un envio automático de notificaciones por correo electrónico y dentro del sistema.

**Tupla: Comodidades**

HUBA-025: Visualización Detallada de Comodidades Disponibles

Como: Comodidades

Quiero: poder ver una descripción completa y detallada de todas las comodidades disponibles en la propiedad

Para: ofrecer información precisa y enriquecedora que mejore significativamente la experiencia de búsqueda del usuario, permitiéndole tomar decisiones informadas.

HUBA-026: Identificación Clara de Comodidades No Disponibles

Como: Comodidades

Quiero: que el sistema muestre de forma clara las comodidades con las que la propiedad no cuenta,

Para: evitar cualquier expectativa errónea o inconveniente al momento de mi estancia, asegurando una total transparencia.

HUBA-027: Administración Integral de Comodidades de la Propiedad

Como: Comodidades

Quiero: tener la capacidad de añadir nuevas comodidades, eliminar las existentes o actualizar la información de estas,

Para: mantener la lista de servicios siempre al día y así garantizar que los usuarios accedan a la información más precisa y relevante, mejorando constantemente la calidad de nuestro servicio.

**Tupla: Locación**

HUAP-028 - Actualizar información de la locación

Cómo: Locación

Quiero: Editar mis datos como dirección, descripción, y contacto

Para: Mantener mi información actualizada y brindar datos correctos a los huéspedes

HUAP-029 - Ver historial de reservas

Cómo: Locación

Quiero: Consultar las reservas que se han realizado en mi espacio

Para: Llevar un control de los huéspedes que me han visitado

HUJM-030 - Establecer horario de check-in y del check-out

Cómo: Locación

Quiero: Definir un horario disponible para la llegada y salida de los huéspedes

Para: Asegurar que las estancias se organicen de forma eficiente y sin conflictos

**Tupla: Usuario**

HUKC-031 **Registro de Nuevo Usuario**

Como: Un nuevo usuario (anfitrión o invitado) Quiero: Registrarme en el sistema proporcionando mi nombre, correo, teléfono, fecha de nacimiento y tipo de usuario (anfitrión o invitado) Para: Poder utilizar los servicios de la plataforma, ya sea para ofrecer propiedades o para reservarlas.

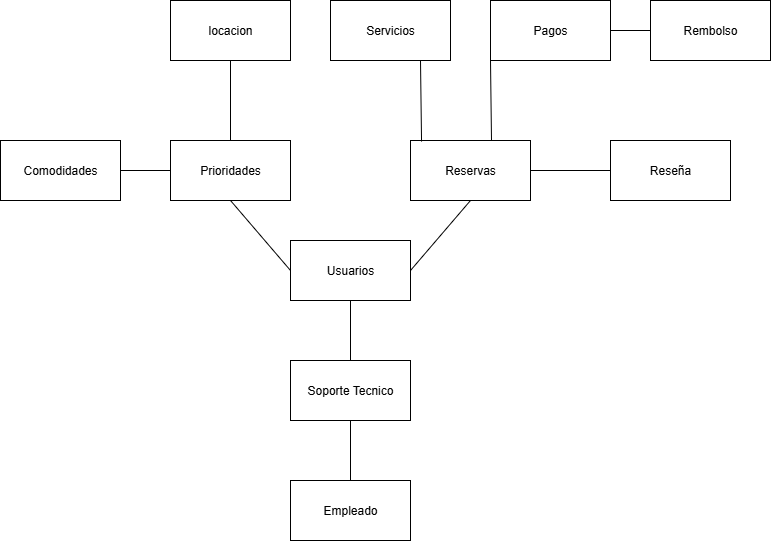
HUKC-032 **Edición de Perfil de Usuario**

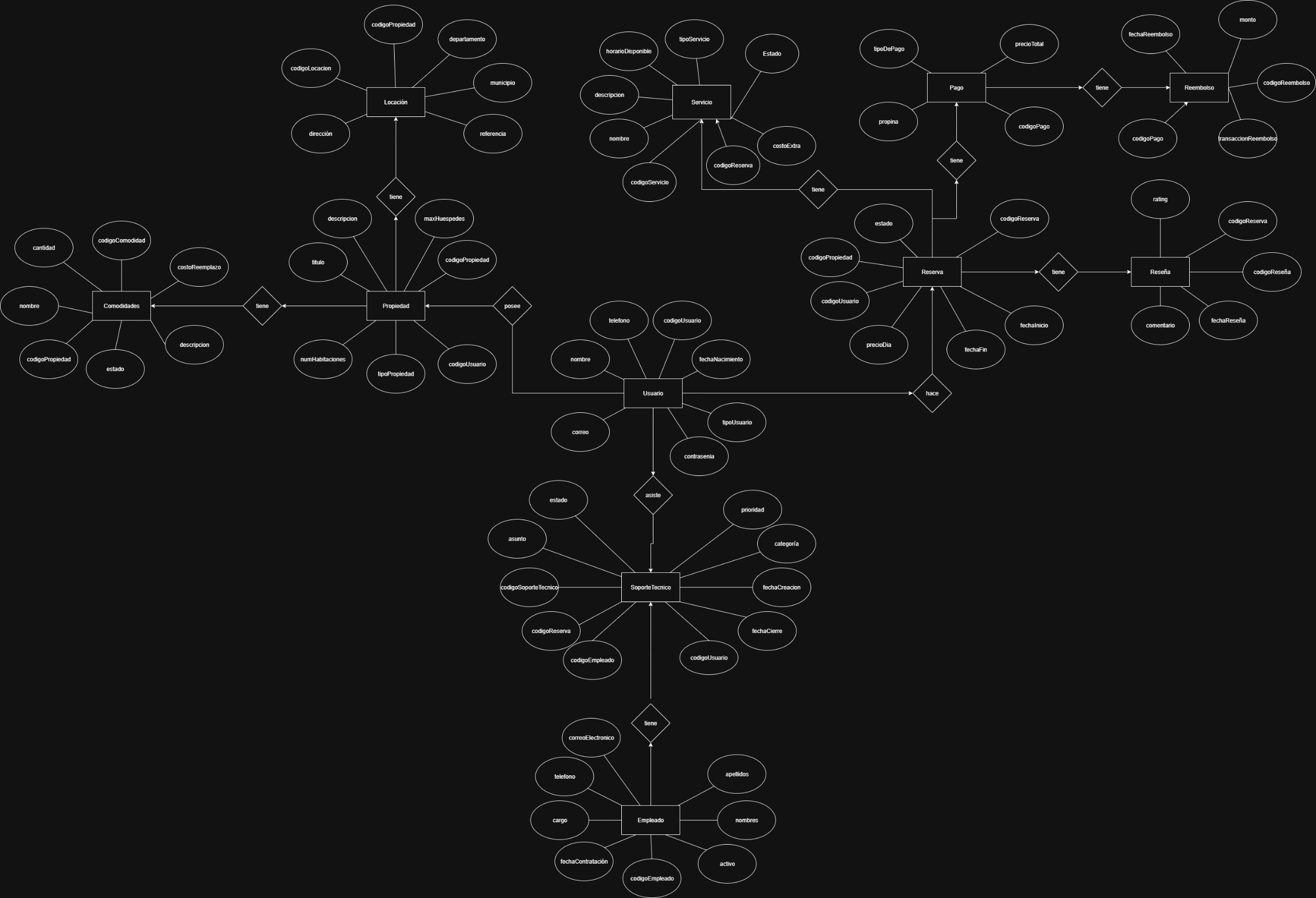
Como: Un usuario registrado Quiero: Poder editar mi nombre, correo, teléfono, fecha de nacimiento, tipo de usuario y subir imágenes a mi perfil Para: Mantener mi información actualizada y personalizada en la plataforma.

HUKC-003 **Visualización de Perfil de Usuario**

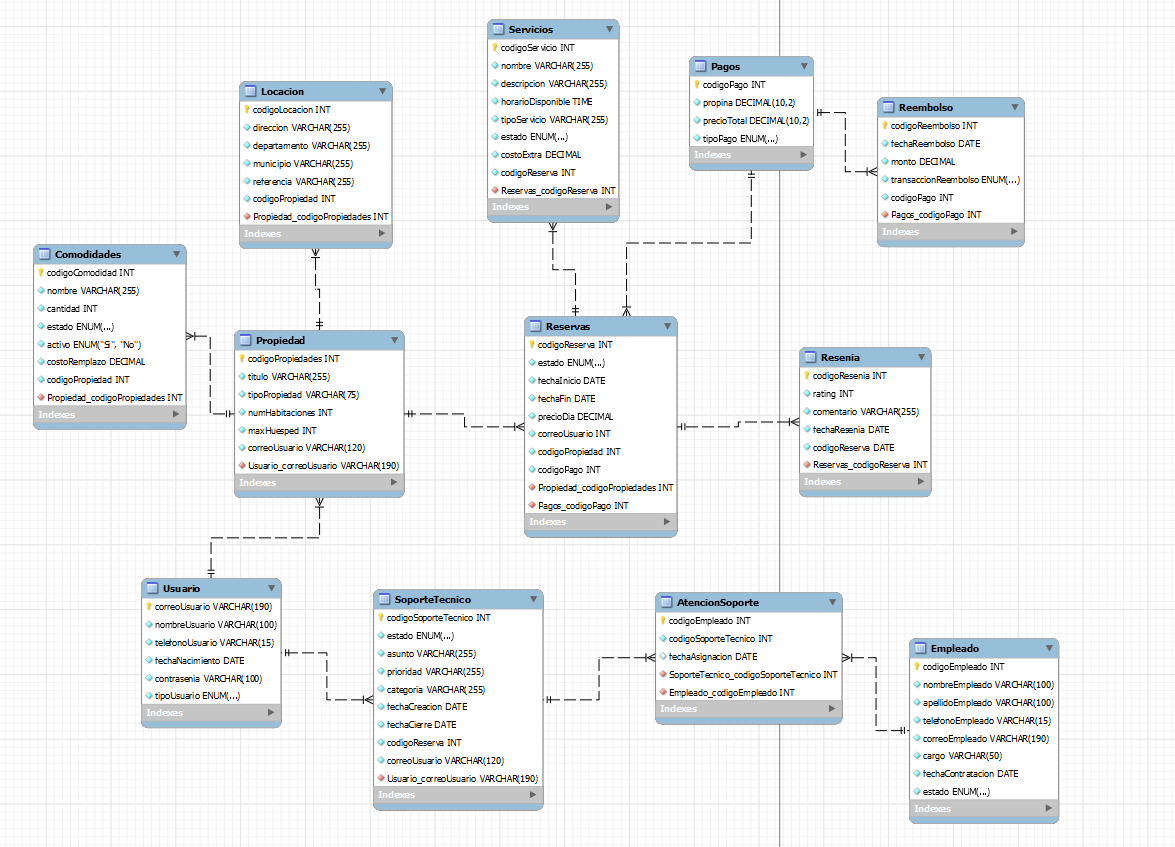
Como: Un usuario (o un empleado del área con permisos) Quiero: Poder ver el perfil de un usuario, incluyendo su nombre, correo, teléfono, fecha de nacimiento, tipo de usuario y sus imágenes Para: Obtener información relevante sobre un usuario específico en la plataforma.

**MODELO DE ENTIDAD**

****

**MODELO ENTIDAD RELACION**

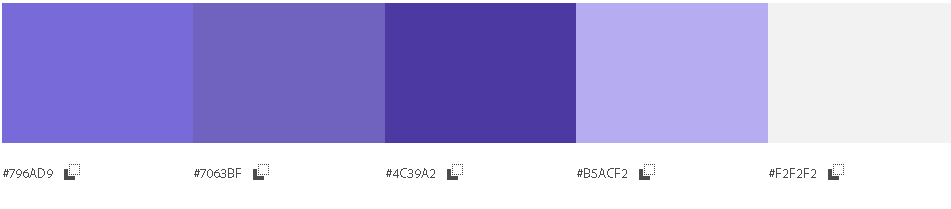
**DIAGRAMA ENTIDAD-RELACIÓN**

****

**LOGO PEACE AT HOME**

****

****

****

**Base De Datos MySQL**

drop database if exists DBPeaceAtHome;

create database if not exists DBPeaceAtHome;

use DBPeaceAtHome;

create table Usuario(

correoUsuario varchar(190) not null,

nombreUsuario varchar(100) not null,

telefonoUsuario varchar(15) not null,

fechaNacimiento date,

contrasenia varchar(100),

tipoUsuario enum("Huesped","Arrendador","Administrador"),

primary key PK\_correoUsuario (correoUsuario)

);

create table Pago(

codigoPago int not null auto\_increment,

propina decimal(10,2),

precioTotal decimal(10,2) not null,

tipoPago enum("Transferencia","PayPal"),

primary key PK\_codigoPago (codigoPago)

);

create table Propiedad(

codigoPropiedad int not null auto\_increment,

titulo varchar(100) not null,

descripcion varchar(255),

tipoPropiedad varchar(75) not null,

numHabitaciones int not null,

maxHuesped int not null,

correoUsuario varchar(120) not null,

primary key PK\_codigoPropiedad (codigoPropiedad),

constraint FK\_Propiedad\_Usuario foreign key (correoUsuario)

references Usuario(correoUsuario)

);

create table Comodidades(

codigoComodidad int not null auto\_increment,

nombre varchar(255) not null,

cantidad int not null,

estado enum("Nuevo","Usado","Fuera de Servicio") not null,

activo enum("Si", "No") not null,

costoRemplazo decimal(5,2),

codigoPropiedad int not null,

primary key PK\_codigoComodidad (codigoComodidad),

constraint FK\_Comodidades\_Propiedad foreign key (codigoPropiedad)

references Propiedad(codigoPropiedad)

);

create table Reserva(

codigoReserva int not null auto\_increment,

estado enum("Disponible","Ocupada","Cancelada") not null,

fechaInicio date not null,

fechaFin date not null,

precioDia decimal(5,2) not null,

correoUsuario varchar(190) not null,

codigoPropiedad int not null,

codigoPago int,

primary key PK\_codigoReserva (codigoReserva),

constraint FK\_Reserva\_Propiedad foreign key (codigoPropiedad)

references Propiedad(codigoPropiedad),

constraint FK\_Reserva\_Usuario foreign key (correoUsuario)

references Usuario(correoUsuario),

constraint FK\_Reserva\_Pago foreign key (codigoPago)

references Pago(codigoPago)

);

create table Resenia(

codigoResenia int not null auto\_increment,

rating int check (rating between 1 and 5),

comentario varchar(255),

fechaResenia date not null,

codigoReserva int not null,

primary key PK\_codigoResenia(codigoResenia),

constraint FK\_Resenia\_Reserva foreign key (codigoReserva)

references Reserva(codigoReserva)

);

create table Servicio(

codigoServicio int not null auto\_increment,

nombre varchar(255) not null,

descripcion varchar(255) not null,

horarioDisponible time not null,

tipoServicio varchar(255) not null,

estado enum("Activo","Inactivo"),

costoExtra decimal(5,2),

codigoReserva int not null,

primary key PK\_codigoServicio (codigoServicio),

constraint FK\_Servicio\_Reserva foreign key (codigoReserva)

references Reserva(codigoReserva)

);

create table Reembolso(

codigoReembolso int not null auto\_increment,

fechaReembolso date not null,

monto decimal(10,2) not null,

transaccionReembolso enum("Transferencia","PayPal"),

codigoPago int not null,

primary key PK\_codigoReembolso (codigoReembolso),

constraint FK\_Reembolso\_Pago foreign key (codigoPago)

references Pago(codigoPago)

);

create table Locacion(

codigoLocacion int not null auto\_increment,

direccion varchar(255) not null,

departamento varchar(255) not null,

municipio varchar(255) not null,

referencia varchar(255) not null,

codigoPropiedad int not null,

primary key PK\_codigoLocacion (codigoLocacion),

constraint FK\_Locacion\_Propiedad foreign key (codigoPropiedad)

references Propiedad(codigoPropiedad)

);

create table SoporteTecnico(

codigoSoporteTecnico int not null auto\_increment,

estado enum("Abierto","Procesado","Cerrado") not null,

asunto varchar(255) not null,

prioridad varchar(255) not null,

categoria varchar(255) not null,

fechaCreacion date not null,

fechaCierre date,

codigoReserva int,

correoUsuario varchar(120) not null,

primary key PK\_codigoSoporteTecnico (codigoSoporteTecnico),

constraint FK\_SoporteTecnico\_Usuario foreign key (correoUsuario)

references Usuario(correoUsuario),

constraint FK\_SoporteTecnico\_Reserva foreign key (codigoReserva)

references Reserva(codigoReserva)

);

create table Empleado(

codigoEmpleado int not null auto\_increment,

nombreEmpleado varchar(100) not null,

apellidoEmpleado varchar(100) not null,

telefonoEmpleado varchar(15) not null,

correoEmpleado varchar(190) not null,

cargo varchar(50) not null,

fechaContratacion date not null,

estado enum("Activo","Inactivo") not null,

primary key PK\_codigoEmpleado (codigoEmpleado)

);

create table AtencionSoporte (

codigoEmpleado int not null,

codigoSoporteTecnico int not null,

fechaAsignacion date not null,

primary key (codigoEmpleado, codigoSoporteTecnico),

constraint FK\_Atencion\_Empleado foreign key (codigoEmpleado)

references Empleado(codigoEmpleado),

constraint FK\_Atencion\_Soporte foreign key (codigoSoporteTecnico)

references SoporteTecnico(codigoSoporteTecnico)

);