

I.E.S Fuengirola N°1

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROYECTO INTEGRADO

Software de gestión de complejos turísticos de alta ocupación

Departamento de Informática

Manual Técnico

Autor: Antonio Socías Verdugo

ÍNDICE DE CONTENIDOS

1 - Sobre éste proyecto	2
1.1 - Control de versiones	2
1.2 - Licencia de uso	2
2 - Análisis del problema	2
2.1 - Introducción al problema	2
2.2 - Antecedentes	2
2.3 - Objetivos	2
2.4 - Requisitos	2
2.4.1 - Funcionales	2
2.4.2 - No funcionales	2
2.5 - Recursos	2
2.5.1 - Software	2
2.5.2 - Hardware	2
3 - Diseño de la solución software	2
3.1 - Modelados	2
3.1.1 - Casos de uso	2
3.1.2 - [Interacción]	2
3.1.3 - [Estado]	2
3.1.4 - [Actividad]	2
3.2 - Prototipado gráfico	2
3.2.1 - [Escritorio]	2
3.2.2 - [Tablets / Smartphones]	2
3.3 - Base de datos	2
3.3.1 - Diseño Conceptual (ER)	2
3.3.2 - Diseño lógico (tablas normalizadas)	2
4 - Implementación	2
4.1 - Codificación	2
4.1.1 - [Usabilidad]	2
4.1.2 - Backend	2
4.1.3 - Frontend	2
4.2 - [Pruebas]	2
5 - Documentación	2
5.1 - Empaquetado / Distribución	2
5.2 - Instalación	2
5.3 - Manual de Usuario / Referencia	2
6 - Conclusiones	2
7 - Bibliografía	2

1 - Sobre éste proyecto

Este Trabajo de Fin de Grado tiene como objetivo desarrollar un sistema de gestión integral pensado específicamente para complejos vacacionales. A diferencia de muchos programas disponibles en el mercado, que suelen estar dirigidos a hoteles o apartamentos turísticos tradicionales, este proyecto se enfoca en una realidad menos atendida: los complejos donde los apartamentos pertenecen a distintos propietarios y no se alquilan directamente como si fueran habitaciones de hotel.

La idea surge a raíz de mi experiencia laboral, donde he podido ver de primera mano las limitaciones que tienen los gestores de este tipo de complejos para llevar el control de apartamentos, propietarios, empleados y accesos. No existe un sistema que lo integre todo de forma eficiente y adaptada a sus necesidades específicas, por lo que vi una oportunidad de aportar algo útil.

El sistema que propongo permitirá gestionar la información clave de manera centralizada y sencilla: desde saber qué apartamentos están disponibles, quiénes son sus propietarios, hasta llevar un control del personal, sus horarios, salarios y accesos. Además, como parte más innovadora del proyecto, he querido incluir un sistema de control de acceso mediante pulseras NFC. Esto permitirá que los huéspedes puedan entrar a zonas restringidas como piscinas o gimnasios sin necesidad de llaves o tarjetas, de forma cómoda y segura.

Esta memoria tiene como objetivo explicar el desarrollo del sistema, desde la idea inicial hasta su implementación final, y mostrar cómo se ha buscado dar respuesta a una necesidad real con una solución práctica y adaptada al entorno en el que se va a aplicar.

1.1 - Control de versiones

El desarrollo del sistema se ha gestionado siguiendo una estructura de control de versiones basada en Git, utilizando GitHub como plataforma principal para el almacenamiento del código fuente, seguimiento de errores, documentación y organización de tareas.

A lo largo del proceso se han establecido distintas versiones alfa y beta del software, cada una con funcionalidades incrementales y mejoras tras cada iteración:

- v0.1 (Alfa) – Estructura inicial del backend con Spring Boot y conexión a la base de datos PostgreSQL. Diseño preliminar del modelo de datos.
- v0.2 (Alfa) – Implementación del sistema de autenticación básica para empleados y administradores.
- v0.3 (Beta) – Desarrollo de funcionalidades CRUD para entidades clave: propietarios, apartamentos, reservas, empleados y pulseras NFC.
- v0.4 (Beta) – Diseño e implementación del sistema de control de fichajes de empleados y control de accesos mediante pulseras.

- v1.0 (Versión estable) – Integración de todas las funcionalidades previstas, pruebas generales del sistema y corrección de errores. Interfaz gráfica final adaptada para entorno kiosko y escritorio (React Native + Tauri).

El control de versiones permitirá mantener una trazabilidad clara sobre los avances del proyecto, así como la posibilidad de revertir cambios en caso de errores críticos.

1.2 - Licencia de uso

Este proyecto se encuentra actualmente licenciado bajo la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0). Esta licencia permite copiar, distribuir y modificar el contenido, siempre que se reconozca la autoría original, no se utilice con fines comerciales, y las obras derivadas se publiquen bajo la misma licencia.

Sin embargo, dado que he concebido esta aplicación como una solución real y potencialmente aplicable en el mercado, contemplo la posibilidad de su evolución futura hacia una versión comercial. En ese caso, el modelo de licencia sería revisado y adaptado, pudiendo adoptar una licencia propietaria o una licencia permisiva como MIT o Apache 2.0, que permita el uso comercial y una mayor flexibilidad de distribución.

2 - Análisis del problema

2.1 - Introducción al problema

En muchos complejos vacacionales, especialmente aquellos gestionados por comunidades de propietarios donde cada apartamento pertenece a una persona distinta, no existen herramientas de gestión eficientes que contemplen todas las particularidades de este modelo. La mayoría del software del mercado está orientado a hoteles o a agencias para que consigan y gestionen sus reservas, lo que genera múltiples ineficiencias en la operativa diaria de estos complejos descentralizados.

La gestión se realiza habitualmente con una combinación de hojas de cálculo, correos electrónicos y sistemas independientes, generando duplicidades, errores y una alta carga de trabajo manual. Además, no se incluye la gestión moderna de accesos ni el control eficiente del personal. Lo cual conlleva la contratación de software de terceros y dependencia de sus mantenimientos, aumentando el coste final de gestión el complejo. Por tanto, se detecta la necesidad de un sistema que centralice toda la gestión de apartamentos, propietarios, empleados y accesos en una sola herramienta adaptada a este entorno concreto.

2.2 – Antecedentes

Durante la fase de investigación previa, se estudiaron varias soluciones comerciales como Siteminder, RoomRaccoon o AvaiBook. Aunque estos sistemas cuentan con amplias funcionalidades, están pensados para hoteles tradicionales con una sola entidad gestora.

Ninguno de ellos contempla la realidad de una propiedad compartida ni la individualización de la información por propietario. Tampoco incluyen de forma integrada un sistema moderno de control de acceso como pulseras NFC, ni herramientas para el fichaje de empleados o control de tareas internas. Esto evidencia una carencia en el mercado y justifica la creación de una solución específica para este tipo de complejos vacacionales.

2.3 – Objetivos

Objetivo general:

Desarrollar un sistema de gestión integral y multiplataforma para complejos vacacionales con propiedad compartida, que centralice y simplifique todas las tareas de administración, acceso y gestión de personal.

Objetivos específicos:

- Gestionar apartamentos con información de estado, incidencias y disponibilidad.
- Asociar apartamentos a sus respectivos propietarios con visibilidad individualizada.
- Controlar horarios, salarios y eventos de fichaje de empleados.
- Implementar un sistema seguro de acceso mediante pulseras NFC para zonas restringidas.
- Garantizar una interfaz clara, moderna y fácil de usar, adaptable a terminales táctiles y ordenadores.

2.4 - Requisitos

2.4.1 – Funcionales

- El sistema permitirá CRUD completo de apartamentos, propietarios, reservas, empleados y pulseras.
- Registro y modificación de turnos y eventos de fichaje de los empleados.
- Asignación y validación de pulseras NFC.
- Gestión de accesos físicos a zonas restringidas.
- Visualización de datos en tiempo real y generación de informes.

2.4.2 - No funcionales

- Interfaz usable por usuarios no técnicos.
- Compatibilidad multiplataforma (Windows, Linux).
- Funcionamiento offline/local (sin necesidad de servidor externo).
- Seguridad en el almacenamiento y transmisión de datos.
- Tiempo de respuesta adecuado para consultas y registros.

2.5 – Recursos

2.5.1 – Software

- Frontend: React + Tauri
- Backend: Python (Flask)
- Base de datos: PostgreSQL
- Control de versiones: GitHub
- IDE: Visual Studio Code
- Prototipado: Figma.

2.5.2 - Hardware

- Ordenador personal (mínimo i5, 8 GB RAM).
- Dispositivo Android (tablet) para pruebas de interfaz táctil.
- Lector NFC USB (para simular escaneo de pulseras).
Para pruebas de desarrollo se usará un dispositivo android con lector NFC
- Conexión a internet para instalación y pruebas iniciales.

3 - Diseño de la solución software

3.1 – Modelados

3.1.1 - Casos de uso

Se han definido varios casos de uso principales para los distintos perfiles del sistema: administradores y empleados. Estos casos de uso permiten entender cómo interactúan los usuarios con la aplicación y qué funcionalidades están disponibles según sus roles.

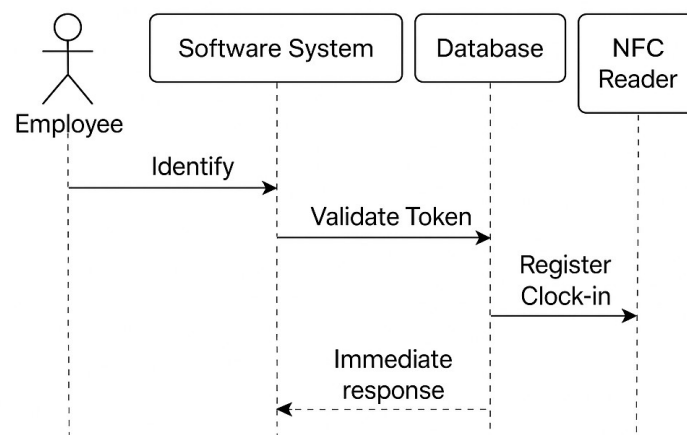
Los casos de uso principales son:

- Gestión de apartamentos: Alta, baja y modificación de información de apartamentos, asignación de propietarios, y consulta de estado y reservas asociadas.
- Gestión de propietarios: Registro de nuevos propietarios, modificación de datos y consulta de su historial de actividad.
- Gestión de empleados: Registro de empleados, control de horarios y acceso a funcionalidades administrativas según el rol.
- Control de accesos: Lectura de pulseras NFC para permitir o denegar el acceso a zonas restringidas.
- Gestión de reservas: Creación y modificación de reservas, asignación a apartamentos disponibles y control de fechas.
- Registro de fichajes: Los empleados pueden fichar su entrada y salida del trabajo mediante el sistema.
- Autenticación y gestión de sesiones: Inicio de sesión seguro y validación mediante tokens, diferenciando permisos según el rol del usuario.

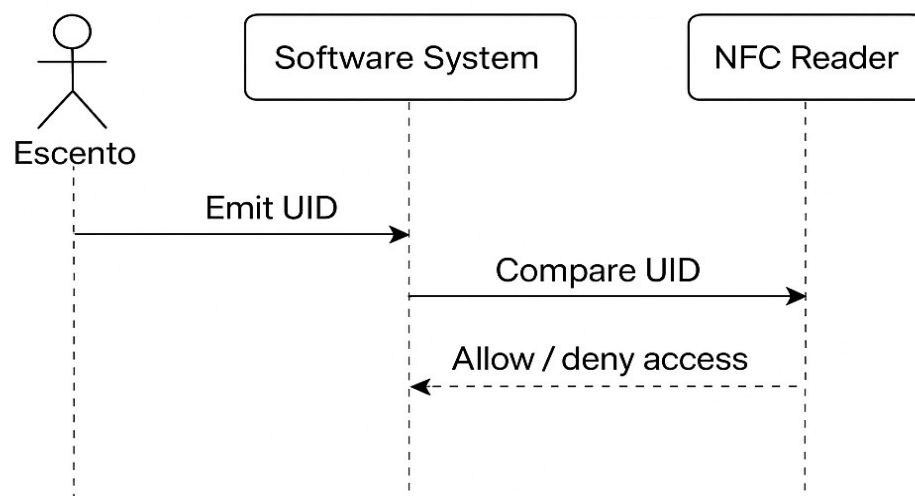
3.1.2 - [Interacción]

Los diagramas de interacción reflejan la comunicación entre los distintos módulos del sistema y los actores implicados, representándose principalmente mediante diagramas de secuencia. A continuación se describen dos de los procesos más relevantes:

- Fichaje de empleados: el empleado se identifica a través de su usuario y contraseña o mediante una pulsera vinculada. El sistema valida el token de autenticación o el UID recibido, registra el evento (entrada o salida) en la base de datos y proporciona una respuesta inmediata indicando el estado de la operación.



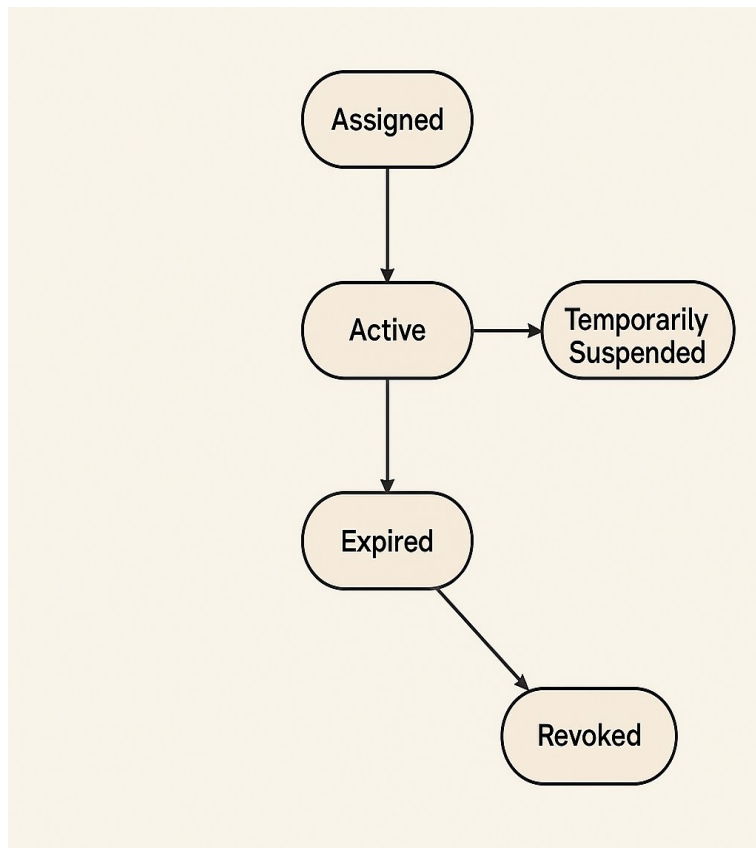
- Control de acceso mediante pulsera: el lector NFC captura el UID de la pulsera al aproximarla. El sistema verifica su validez y estado en la base de datos, comprueba los permisos asociados a la zona solicitada y, en función del resultado, concede o deniega el acceso. La acción se registra en la tabla de accesos como válida o rechazada.



3.1.3 - [Estado]

Los diagramas de estado son especialmente relevantes en el módulo de control de accesos, ya que permiten representar las distintas situaciones en las que puede encontrarse una pulsera a lo largo de su ciclo de vida.

Una pulsera puede estar en los siguientes estados:



- **Asignada:** la pulsera ha sido vinculada a un usuario (cliente o empleado), pero aún no está activa.
- **Activa:** la pulsera está en uso y permite el acceso a las zonas autorizadas.
- **Suspendida temporalmente:** la pulsera ha sido desactivada de forma provisional, por ejemplo, por mantenimiento o por incidencias detectadas.
- **Caducada:** ha expirado su periodo de validez (por ejemplo, al finalizar una reserva vinculada).
- **Revocada:** la pulsera ha sido invalidada permanentemente y no podrá volver a utilizarse.

Este sistema de estados permite evitar usos indebidos, mejorar la trazabilidad de los accesos y automatizar permisos en función de factores como el estado de la reserva o el estado del apartamento asociado al cliente.

3.1.4 - [Actividad]

Los diagramas de actividad reflejan los flujos de trabajo principales. Entre ellos:

- **Proceso de creación de reserva:** Selección de fechas → Validación de disponibilidad → Confirmación y asignación → Registro del acceso.
- **Asignación de pulsera:** Selección de huésped o empleado → Escaneo de la pulsera → Validación → Registro.

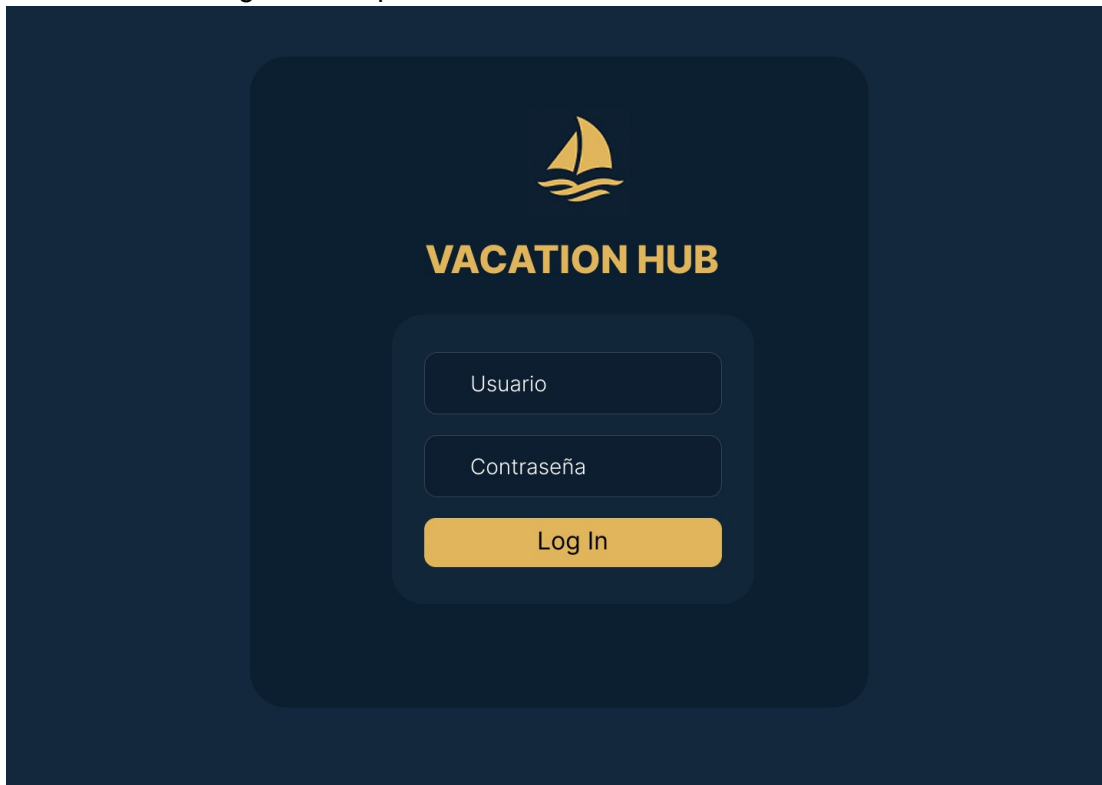
3.2 - Prototipo gráfico

Durante la fase de diseño, se ha desarrollado un prototipado gráfico de alta fidelidad usando **Figma**, tanto para la versión de escritorio como para tablets. Este prototipo ha servido de guía para la implementación y también se incluirá como anexo en la memoria.


3.2.1 - [Escritorio]

El prototipo para la versión de escritorio está orientado a los administradores o empleados (con menos nivel de permisos, por tanto menos) del complejo. Algunas de las pantallas incluidas son:

- Panel de Log-in de empleados



- Panel principal con resumen de estado general.

 VACATION HUB

Panel de control

Reservas

Apartamentos

Pulseras de acceso


Registro Accesos

Panel de control

Bienvenido, Administrador

Antonio Manuel

- Gestión de apartamentos y propietarios.

 VACATION HUB

Panel de control

Reservas

Apartamentos

Pulseras de acceso


Registro Accesos

Apartamentos

Buscar...

Añadir Apartamento

Número	Propietario	Cliente	Check-out	Estado	Acciones
2-105	Pablo casado	05/05/2024	15/05/2024	Confirmado	Ver ▼
4-501	Rajoy	05/05/2024	15/05/2024	Pendiente	Ver ▼
2-605	Zapatero	-	-	Mantenimiento	Ver ▼

 VACATION HUB

[Panel de control](#)
[Reservas](#)
[Apartamentos](#)
[Pulseras de acceso](#)
[Registro Accesos](#)

Apartamentos

Añadir apartamento

Número

Propietario

Dueño

Fulanito

▼

Estado

Activo

▼

Capacidad

▼


Gestionado por

▼

Cancelar

Guardar

- Asignación de pulseras y control de accesos.

 VACATION HUB

[Panel de control](#)
[Reservas](#)
[Apartamentos](#)
[Pulseras de acceso](#)
[Registro Accesos](#)

Access Bands

Asignar Band

ID de Pulsera	Asignado a	Tipo	Estado	Acciones
123456	John Doe	Cliente	Activo	Asignar
234567	Jaime García	Empleado	Bloqueado	Eliminar
234567	-	-	Activa	Asignar

The screenshot shows a web interface for 'VACATION HUB'. On the left is a dark blue sidebar with a logo and a menu: 'Panel de control', 'Reservas', 'Apartamentos', 'Pulseras de acceso' (highlighted in orange), and 'Registro Accesos'. The main content area is light gray and contains a white form titled 'ASIGNAR NUEVA PULSERA'. The form has the following fields: 'ID de Pulsera' (text input), 'Tipo' (dropdown menu with 'Cliente' selected), 'Usuario' (dropdown menu with 'Buscar por nombre' selected), and 'Estado' (dropdown menu with 'Activo' selected). At the bottom right of the form are two buttons: 'Cancelar' (white) and 'Guardar' (orange).

- Visualización del log de accesos.

The screenshot shows the 'Registro Accesos' section of the VACATION HUB interface. It features a table with access logs. Above the table are three filter dropdowns: 'Usuario', 'Tipo', and 'Fecha', followed by an orange 'Filtrar' button. The table has four columns: 'Nombre', 'Tipo', 'Zona', and 'Fecha'. It contains three rows of data.

Nombre	Tipo	Zona	Fecha
Maria Gómez	Cliente	Piscina	05/05/2024
Laura Castillo	Empleado	Jardin	05/05/2024
Pablo Motos	Empleado	Calderas	15/05/2024

3.2.2 - [Tablets / Smartphones]

3.3 - Base de datos

Se ha definido un modelo entidad-relación que recoge las principales entidades del sistema:

Apartamento, Propietario, Reserva, Empleado, Horario, Pulsera, Acceso, Evento_fichaje, entre otras tablas.

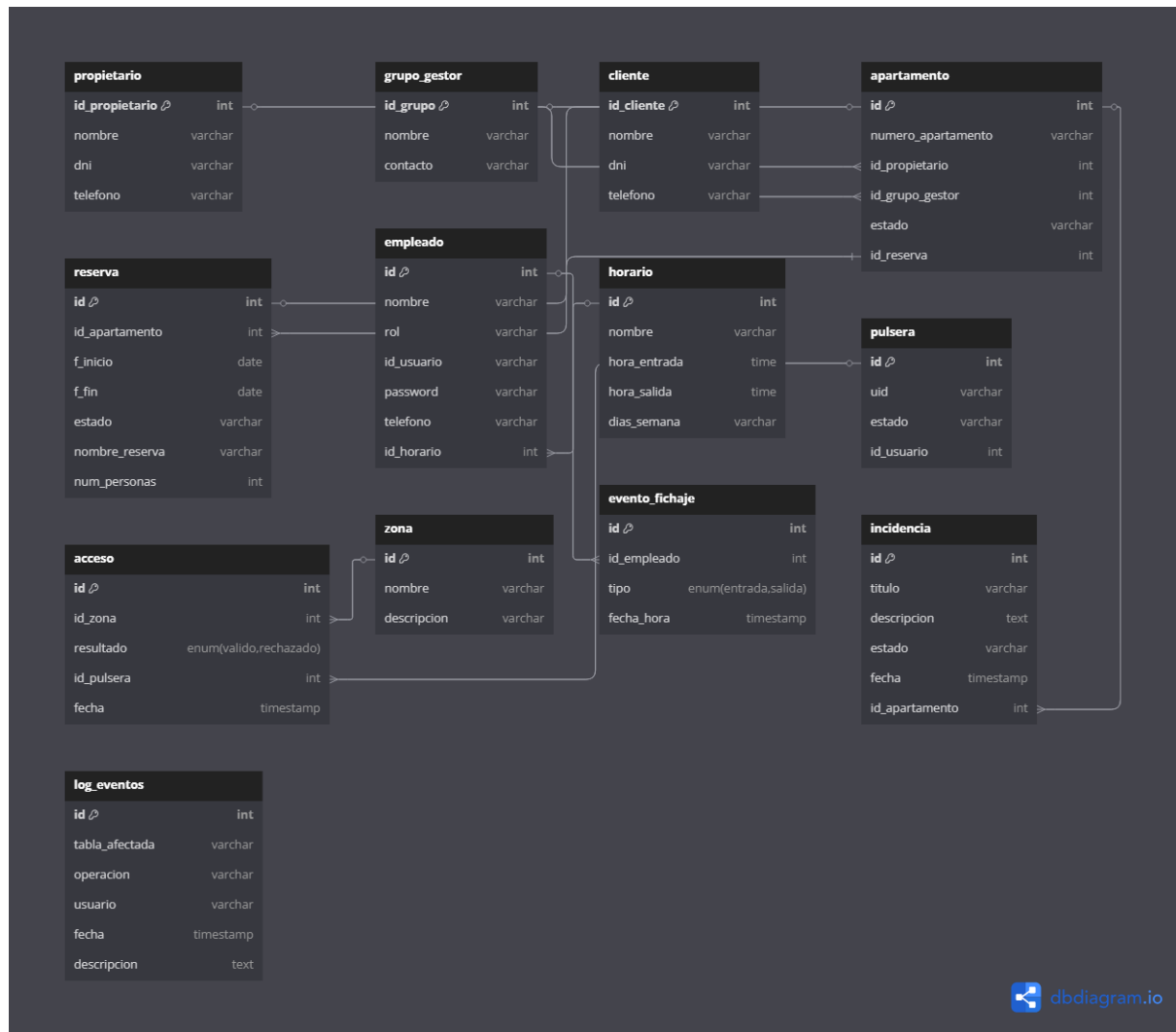
Estas entidades están relacionadas mediante claves foráneas y respetan las reglas de integridad referencial, reflejando fielmente las necesidades del dominio real.

3.3.1 - Diseño Conceptual (ER)

3.3.2 - Diseño lógico (tablas normalizadas)

El modelo lógico ha sido generado a partir del diseño conceptual, manteniendo la tercera forma normal (3FN) para evitar redundancias y asegurar consistencia. Algunas de las tablas principales son:

- apartamento(id, numero_apartamento, id_propietario, id_grupo_gestor, estado, id_reserva)
- propietario(id_propietario, nombre, dni, telefono)
- grupo_gestor(id_grupo, nombre, contacto)
- cliente(id_cliente, nombre, dni, telefono)
- reserva(id, id_apartamento, f_inicio, f_fin, estado, nombre_reserva, num_personas)
- empleado(id, nombre, rol, id_usuario, password, telefono, id_horario)
- horario(id, nombre, hora_entrada, hora_salida, dias_semana)
- pulsera(id, uid, estado, id_usuario)
- acceso(id, id_zona, resultado, id_pulsera, fecha)
- zona(id, nombre, descripcion)
- evento_fichaje(id, id_empleado, tipo, fecha_hora)
- incidencia(id, titulo, descripcion, estado, fecha, id_apartamento)
- log_eventos(id, tabla_afectada, operacion, usuario, fecha, descripcion)



4 - Implementación

4.1 - Codificación

4.1.1 - [Usabilidad]

4.1.2 - Backend

4.1.3 - Frontend

4.2 - [Pruebas]

5 - Documentación

5.1 - Empaquetado / Distribución

5.2 - Instalación

5.3 - Manual de Usuario / Referencia

6 - Conclusiones

7 - Bibliografía