

An aerial view of a modern coworking space. The room features several wooden desks with computers and monitors. There are two large L-shaped sofas, one blue and one yellow, with small coffee tables in front of them. The floor is light-colored tile, and there are potted plants scattered throughout the space. The overall atmosphere is bright and professional.

# SISTEMA DE GESTIÓN DE RESERVAS PARA ESPACIOS DE COWORKING

Eliú Manuel Viera Lorenzo

David Liaño Macías

2ºDAM - DAD

*Proyecto creado Por Eliú Manuel Viera Lorenzo y David Liaño Macías.*

*2º DAM Semipresencial*

*2024-2025*

*Desarrollo de Interfaces.*

*Proyecto: Aplicación Sistema de Gestión de Reservas para Espacios de Coworking*

**WORKTOPIA**

## INDICE

Descripción	Pag.1-3
Idea General	Pag.1
Estructura	Pag.1
Módulos	
Principales	Pag.1
Herramientas	Pag.1-2
Interfaces	Pag.2
Base de	
Datos	Pag.2
Desarrollo	Pag.3
Pruebas	Pag.3
Objetivos	Pag.3
Base de datos	Pag.4-5
Esquema	Pag.4
MySQL	Pag.4-5



## DESCRIPCIÓN

### IDEA GENERAL

Un programa para administrar espacios de coworking o salas de reuniones. Con características para reservar espacios por horarios y capacidades, interfaz gráfica para reservas. Facturación automática por uso del espacio. Registro de clientes frecuentes y descuentos personalizados. Estadísticas de uso de espacios para optimizar recursos. Aplicación comercial: Diseñado para negocios de coworking en expansión.

### ESTRUCTURA GENERAL DEL PROYECTO

#### MÓDULOS PRINCIPALES

- Gestión de Usuarios:

La aplicación poseerá un registro y edición de clientes. Cada uno tendrá su autenticación e histórico de reservas.

- Gestión de Espacios:

Registro de salas con datos como capacidad, disponibilidad y tarifas por hora o periodo. Actualización del estado (ocupado, libre, mantenimiento, etc.). Se dispondrá de un total de 3 espacios diferentes para reservar.

- Sistema de Reservas:

Interfaz visual para seleccionar fechas, horarios o periodos, asientos libres, ocupados o seleccionados además de servicios proporcionados en el espacio como máquinas expendedoras o servicios. Validación de disponibilidad en tiempo real. Generación de facturas al completar reservas.

- Facturación

Cálculo automático de precios basado en el periodo de reserva o descuentos por fidelidad. Emisión de comprobantes (PDF o formato simple).

#### HERRAMIENTAS Y TECNOLOGÍAS

- IDE:

Se utilizará IntelliJ IDEA.

- Lenguaje

Java.

- Interfaz Gráfica

JavaFX.

- Base de Datos

MySQL para crear la base de datos y Dia para el diagrama relacional.



- Bibliotecas útiles:

JasperReports o iText para la generación de facturas PDF (*Pendiente de valorar y experimentar*).

Daterangepicker como biblioteca para gestión de calendarios o Google Calendar.

- Github

Se intentará el uso de Github para el desarrollo cooperativo de la aplicación.

- Diseño

Se usará Photoshop / Illustrator para generar partes visuales de interfaz y diseño, además de Homebyme para crear los interiores o alternativamente Blender si fuera necesario.

## DISEÑO DE INTERFACES

- Pantallas Principales
  1. Pantalla de presentación de aplicación.
  2. Listado o menú de los 3 espacios en general.
  3. Diseño en vista plano (aérea) de cada espacio.
  4. Selector de calendario (fechas y hora).
- Diseño de usuarios
  1. Diseño de menú de inicio de sesión de usuario (ID y contraseña).
  2. Menú de propiedades del usuario.
  3. Visualizar reservas del usuario, activas y pasadas.
- Diseño de facturación.
  1. Visualización del coste de la reserva dependiendo del modo de realizarla.
  2. Historial de facturación.
  3. Aplicación de impuestos y/o descuentos.
- Elementos visuales a tener en cuenta.
  1. Diseñar iconos de sillas y muebles.
  2. Colores para selección y estado de ocupación.
  3. Diseño de botones e iconos. (Agregar, reservar, editar...).

## BASE DE DATOS

Será una base de datos sencilla donde tan solo necesitamos los datos para clientes, asientos, el lugar del asiento (espacio), los datos de la reserva y factura, por ejemplo:

- Tablas
  1. Clientes: **id\_cliente**, nombre, email, telefono, contrasenia, frecuente (booleano), descuento (porcentaje).
  2. Espacios: **id\_espacio**, nombre.
  3. Asientos: **id\_asiento**, id\_espacio\*, estado (libre, ocupado, no disponible) tarifa\_hora.
  4. Reservas: **id\_reserva**, id\_cliente\*, fecha\_hora\_inicio, fecha\_hora\_fin.
  5. Reservas\_Asentos (*Escoge*): **id\_reserva\***, **id\_asiento\***.



6. Facturas: **id factura**, id\_reserva\*, precio\_total, descuento\_aplicado.

### DESARROLLO

- Crear base de datos: Se hará un esquema de entidad/relación para desarrollar la base de datos. Una vez concretada se comenzará a realizar la base de datos en MySQL.
- De idea a boceto: Se crearán maquetas de las interfaces ya sea a mano, illustrator o similar.
- Interfaz: Para una planificación más ágil se realizará primero la parte de interfaz y diseño. Las interfaces o mapas de los planos de los espacios se desarrollarán en homebyme, blender, illustrator y Photoshop. Una vez realizadas se irán implementando en la interfaz con JavaFX.
- Programación: Una vez diseñado la aplicación, se tendrá más claro los puntos a programar y terminar con el programa.
- Enlace DDBB: Se realizará el enlace con la base de datos.
- Pruebas finales.
- Presentación.

### PRUEBAS

Las pruebas se llevarán a cabo con Junit. Todas las funciones se llevarán a cabo de manera unitaria a medida que se van desarrollando.

Una vez desarrollado el programa sin estar vinculado a la base de datos se llevarán a cabo las pruebas de enlace y funcionamiento general. Probando introducción de usuarios, reservas, ocupaciones, etc.

Por último, si todo está correcto se volverá a dar un análisis completo para ver si necesita algún tipo de optimización, ya sea para mejorar la interfaz o el mismo código.

### OBJETIVOS

Los objetivos iniciales es crear una aplicación funcional donde una empresa que posea unos locales o espacios acondicionados a una oficina de coworking pueda usarla para gestionar reservas.

La aplicación debe poder gestionar a los usuarios/clientes para que puedan realizar sus reservas cómodamente, de manera intuitiva y fácil. Pudiendo ver en todo momento sus reservas actuales, antiguas y facturación.

Debe ser una aplicación visualmente atractiva, minimalista y con colores suaves. Iconos altamente identificativos con sus acciones y localizables.

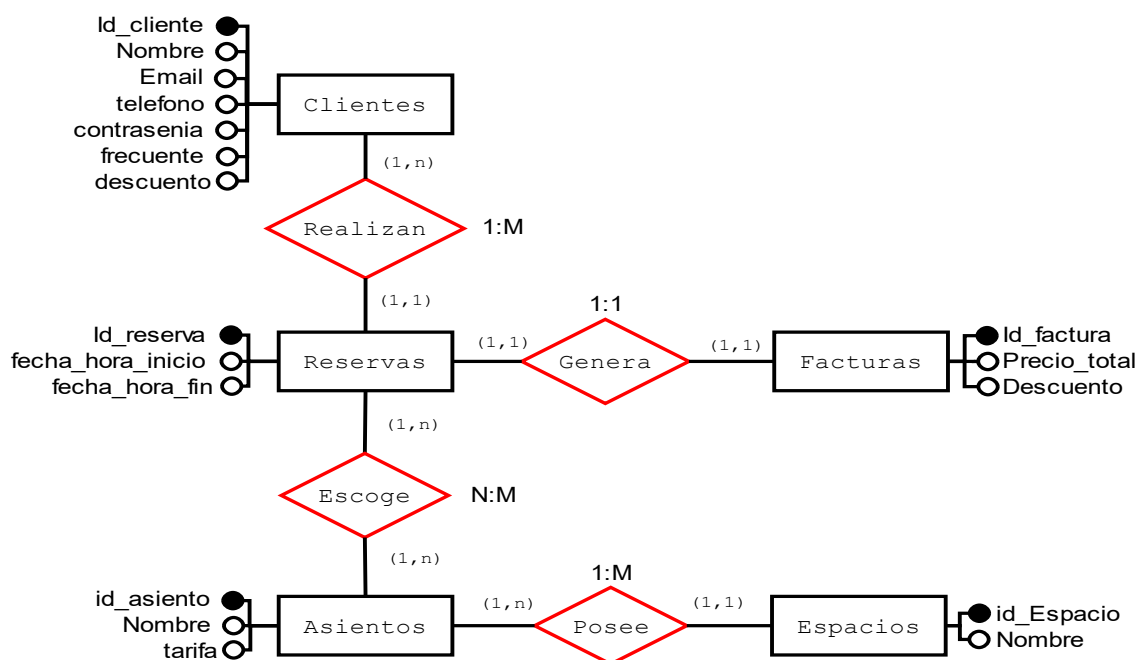
## BASE DE DATOS





Usando el programa Dia para la generación de diagramas y los datos en la descripción, se ha generado el siguiente diagrama:

### DIAGRAMA



Y su MySQL correspondiente

### MYSQL

```
CREATE TABLE Cientes (  
  id_cliente INT AUTO_INCREMENT ,  
  nombre VARCHAR(100) NOT NULL,  
  email VARCHAR(100) NOT NULL UNIQUE,  
  telefono VARCHAR(15),  
  contrasenia VARCHAR(255) NOT NULL,  
  frecuente BOOLEAN DEFAULT FALSE,  
  descuento DECIMAL(5, 2) DEFAULT 0.00  
  CONSTRAINT pk_id_cliente PRIMARY KEY (id_cliente)  
);  
  
CREATE TABLE Espacios (  
  id_espacio INT AUTO_INCREMENT,  
  nombre VARCHAR(100) NOT NULL,  
  CONSTRAINT pk_id_espacio PRIMARY KEY (id_espacio)  
);
```



```
CREATE TABLE Asientos (  
  id_asiento INT AUTO_INCREMENT,  
  estado ENUM('libre', 'ocupado', 'no disponible') NOT NULL,  
  tarifa_hora DECIMAL(10, 2) NOT NULL,  
  id_espacio INT,  
  CONSTRAINT pk_id_asiento PRIMARY KEY (id_asiento),  
  CONSTRAINT fk_id_espacio FOREIGN KEY (id_espacio) REFERENCES  
  Espacios(id_espacio)  
);  
  
CREATE TABLE Reservas (  
  id_reserva INT AUTO_INCREMENT,  
  id_cliente INT,  
  fecha_hora_inicio DATETIME NOT NULL,  
  fecha_hora_fin DATETIME NOT NULL,  
  CONSTRAINT pk_id_reserva PRIMARY KEY (id_reserva),  
  CONSTRAINT fk_id_cliente FOREIGN KEY (id_cliente) REFERENCES  
  Clientes(id_cliente)  
);  
  
CREATE TABLE Reserva_Asientos (  
  id_reserva INT,  
  id_asiento INT,  
  CONSTRAINT pk_reserva_asiento PRIMARY KEY (id_reserva, id_asiento),  
  CONSTRAINT fk_reserva FOREIGN KEY (id_reserva) REFERENCES  
  Reservas(id_reserva),  
  CONSTRAINT fk_asiento FOREIGN KEY (id_asiento) REFERENCES Asientos(id_asiento)  
);  
  
CREATE TABLE Facturas (  
  id_factura INT AUTO_INCREMENT,  
  id_reserva INT UNIQUE,  
  precio_total DECIMAL(10, 2) NOT NULL,  
  tiene_descuento BOOLEAN DEFAULT FALSE,  
  CONSTRAINT pk_id_factura PRIMARY KEY (id_reserva),  
  CONSTRAINT fk_id_reserva FOREIGN KEY (id_reserva) REFERENCES  
  Reservas(id_reserva)  
);
```