A photograph of a modern building facade featuring a grid of balconies with perforated metal railings. The balconies are arranged in a staggered pattern, creating a complex geometric visual. The building has large glass windows and a light-colored facade.

Grand Hotel Manager

MANUAL DEL PROGRAMADOR

Guía para programadores

Junio de 2024
Version No. 1.0

Tabla de contenidos



1.- Introducción.....	2
1.1.- Información destacada.....	2
1.2.- Objetivos.....	2
1.3.- Tecnologías utilizadas.....	3
Objetivos del Manual.....	3
2.- Requerimientos.....	4
3.- Estructura del proyecto.....	5
Organización de Directorios.....	5
4.- Instalación y configuración.....	8
4.- Base de datos.....	10
Modelo Entidad-Relación.....	10
Estructura de Tablas.....	10
Índices y Claves Primarias.....	13
Relaciones entre Tablas.....	14
5.- Servidor.....	15
Instalación de Apache, PHP, phpMyAdmin y MySQL.....	15
Paso 1: Actualizar los paquetes del sistema.....	15
Paso 2: Instalar Apache.....	15
Paso 3: Instalar PHP y extensiones necesarias.....	15
Paso 4: Instalar MySQL (MariaDB).....	15
Paso 5: Configurar MySQL.....	16
Paso 6: Instalar phpMyAdmin.....	16
Configuración de Apache para Aceptar Conexiones Externas.....	16
Paso 1: Editar el archivo de configuración de Apache.....	16
Paso 2: Cambiar la dirección de escucha de Listen 80 a Listen 0.0.0.0:80..	16
Paso 3: Reiniciar Apache para aplicar los cambios.....	17
Estructura del Servidor.....	17
6.- Desarrollo Android.....	19



1.- Introducción

El presente documento describe los aspectos técnicos informáticos de la aplicación. El documento familiariza al personal técnico especializado encargado de las actividades de mantenimiento, revisión, solución de problemas, instalación y configuración del sistema.

1.1.- Información destacada

El manual técnico hace referencia a la información necesaria con el fin de orientar al personal en la concepción, planteamiento análisis programación e instalación del sistema. Es de notar que la redacción propia del manual técnico está orientada a personal con conocimientos en sistemas y tecnologías de información, conocimientos de programación avanzada sobre aplicaciones móviles, administración de bases de datos, responsables del mantenimiento e instalación del sistema en los servidores.

1.2.- Objetivos

Instruir el uso adecuado de la aplicación, para el acceso oportuno y adecuado en la instalación del mismo, mostrando los pasos a seguir en el proceso de instalación, así como la descripción de los archivos relevantes del sistema los cuales nos orienten en la configuración y soporte del mismo.

1.3.- Tecnologías utilizadas

- **Backend:**
 - **Sistema Operativo:** Linux
 - **Servidor Web:** Apache
 - **Lenguaje de Programación:** PHP
 - **Administración de Base de Datos:** phpMyAdmin
 - **Base de Datos:** MySQL (MariaDB)
- **Frontend:**
 - **Lenguaje de Programación:** Java
 - **Bibliotecas Utilizadas:**
 - `com.squareup.okhttp3:okhttp:4.9.0`
 - `org.json:json:20210307`
 - `com.journeyapps:zxing-android-embedded:4.2.0`

Objetivos del Manual

Este manual tiene como objetivo proporcionar a los desarrolladores la información necesaria para:

- Comprender la estructura y arquitectura del sistema.
- Configurar el entorno de desarrollo.

2.- Requerimientos

El sistema puede ser instalado en cualquier sistema operativo o dispositivo móvil que cumpla con los siguientes requerimientos:

- Servidor Web Apache 2.0 o Superior
- PHP 7.0 o Superior
- MySql 5.0 o Superior
- Navegador Web
- Android 7.0

3.- Estructura del proyecto

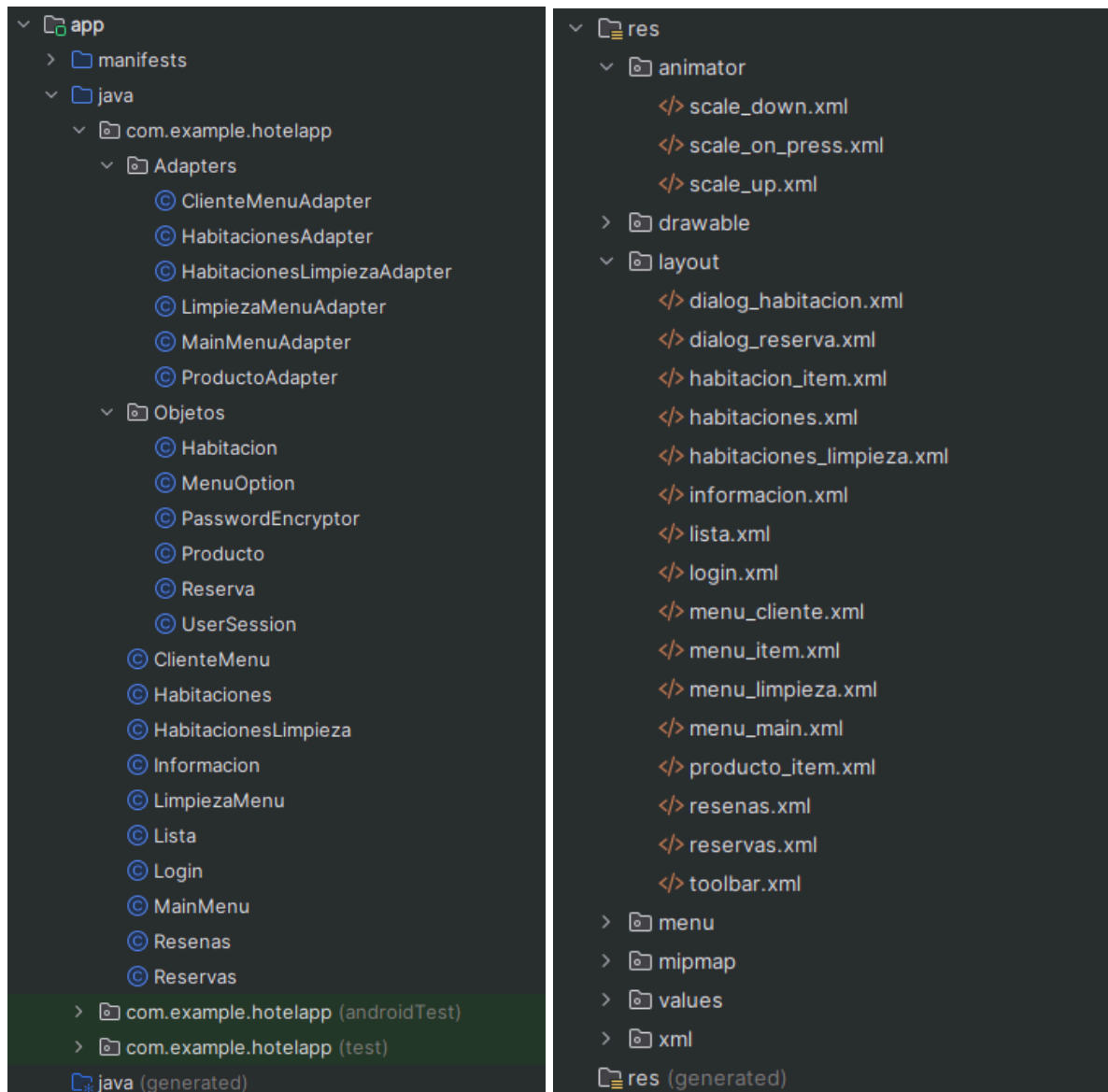
El proyecto de gestión hotelera está organizado en una estructura de directorios clara y lógica para facilitar su desarrollo, mantenimiento y escalabilidad. A continuación, se detalla la estructura de directorios y la función de cada clase y recurso en el proyecto.

Organización de Directorios

1. **app/java/com/example/hotelapp/Adapters:** Este directorio contiene todas las clases de adapters.
 - **Clases de adapters:**
 - ClienteMenuAdapter
 - HabitacionesAdapter
 - HabitacionesLimpiezaAdapter
 - LimpiezaMenuAdapter
 - MainMenuAdapter
 - ProductoAdapter
2. **app/java/com/example/hotelapp/Objetos:** Este directorio contiene todas las clases de objetos.
 - **Objetos:**
 - Habitacion
 - MenuOption
 - PasswordEncryptor
 - Producto
 - Reserva
 - UserSession
3. **app/java/com/example/hotelapp:** Este directorio contiene las clases principales
 - **Clases:**
 - ClienteMenu: Vista para la creación de nuevos clientes.
 - Habitaciones: Vista para la creación de nuevos empleados.

- **HabitacionesLimpieza**: Vista para la creación de nuevas habitaciones.
 - **Informacion**: Vista para la creación de nuevos usuarios.
 - **LimpiezaMenu**: Vista de gestión de habitaciones.
 - **Lista**: Vista de hospedaje de clientes.
 - **Login**: Vista de información detallada de clientes.
 - **MainMenu**: Vista de información detallada de reservas.
 - **Resenas**: Vista de información detallada de empleados.
 - **Reservas**: Vista de información detallada de pedidos.
4. **app/res/layout**: Este directorio contiene las layouts utilizados en la aplicación.
- **Layouts**:
 - **dialog_habitacion**
 - **dialog_reserva**
 - **habitacion_item**
 - **habitaciones**
 - **habitaciones_limpieza**
 - **informacion**
 - **lista**
 - **login**
 - **menu_cliente**
 - **menu_item**
 - **menu_limpieza**
 - **menu_main**
 - **producto_item**
 - **resenas**
 - **reservas**
 - **toolbar**
5. **app/res/drawable**: Este directorio contiene los archivos de imágenes.
6. **app/res/animator**: Este directorio contiene los archivos de animación

Esta organización de directorios y clases asegura una estructura clara y mantenible para el proyecto, facilitando la colaboración entre desarrolladores y permitiendo una fácil navegación y comprensión del código fuente.



4.- Instalación y configuración

Debe descargarse el archivo *apk* subido al repositorio y una vez descargado, lo instala. Una vez hecho esto, la aplicación estaría lista para usarse.

El proyecto también está en el repositorio y su configuración es la siguiente:

Archivo *local.properties*-

```
## This file is automatically generated by Android Studio.
# Do not modify this file -- YOUR CHANGES WILL BE ERASED!
#
# This file should *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
sdk.dir=C:\\Users\\usuario\\AppData\\Local\\Android\\Sdk
```

Se deberá cambiar la ruta del *sdk* correspondiente.

Archivo *build.gradle.kts*-

```
android { this: BaseAppModuleExtension
    namespace = "com.example.hotelapp"
    compileSdk = 34

    defaultConfig { this: ApplicationDefaultConfig
        applicationId = "com.example.hotelapp"
        minSdk = 24
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes { this: NamedDomainObjectContainer<ApplicationBuildType>
        release { this: ApplicationBuildType
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile( name: "proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }

    compileOptions { this: CompileOptions
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}
```

Mantener las versiones correctas.

Archivo *build.gradle.kts* (librerías)-

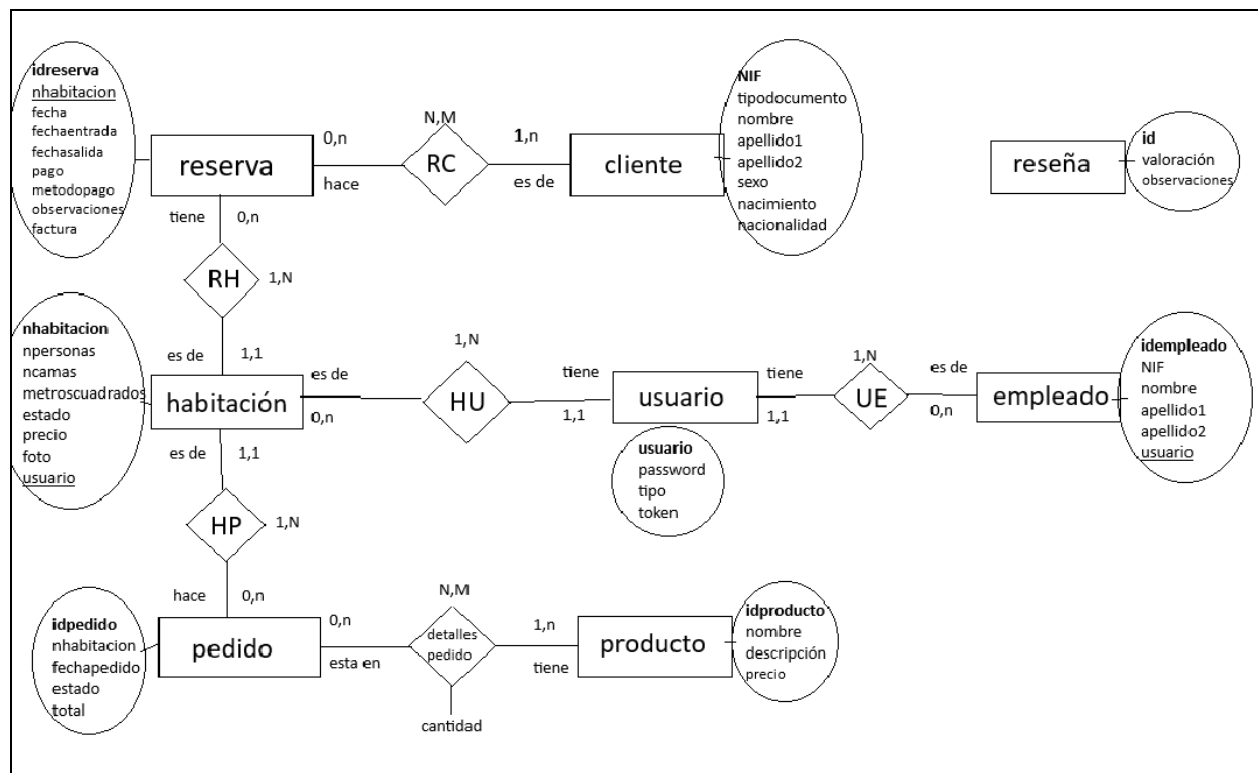
```
dependencies { this: DependencyHandlerScope

    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    implementation("com.squareup.okhttp3:okhttp:4.9.0")
    implementation("org.json:json:20210307")
    implementation("com.journeyapps:zxing-android-embedded:4.2.0")
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)
}
```

4.- Base de datos

La base de datos del proyecto de gestión hotelera está implementada en MariaDB y consta de varias tablas relacionadas que almacenan la información necesaria para gestionar clientes, empleados, habitaciones, pedidos, productos, reseñas, reservas y usuarios. A continuación, se detalla la estructura de la base de datos y las relaciones entre las tablas.

Modelo Entidad-Relación



Estructura de Tablas

1. Tabla **cliente**:

- Almacena la información de los clientes.
- **Campos:**
 - **NIF** (varchar(20), NOT NULL): Identificador único del cliente.
 - **tipodocumento** (varchar(45), DEFAULT NULL): Tipo de documento del cliente.
 - **nombre** (varchar(45), DEFAULT NULL): Nombre del cliente.

- **apellido1** (varchar(45), DEFAULT NULL): Primer apellido del cliente.
 - **apellido2** (varchar(45), DEFAULT NULL): Segundo apellido del cliente.
 - **sexo** (varchar(45), DEFAULT NULL): Sexo del cliente.
 - **nacimiento** (varchar(45), DEFAULT NULL): Fecha de nacimiento del cliente.
 - **nacionalidad** (varchar(45), DEFAULT NULL): Nacionalidad del cliente.
2. **Tabla detallespedido:**
- Almacena los detalles de los pedidos realizados.
 - **Campos:**
 - **iddetalle** (int(11), NOT NULL, AUTO_INCREMENT): Identificador único del detalle.
 - **idpedido** (int(11), DEFAULT NULL): Identificador del pedido.
 - **idproducto** (int(11), DEFAULT NULL): Identificador del producto.
 - **cantidad** (int(11), NOT NULL): Cantidad del producto pedido.
3. **Tabla empleado:**
- Almacena la información de los empleados.
 - **Campos:**
 - **idempleado** (int(11), NOT NULL, AUTO_INCREMENT): Identificador único del empleado.
 - **NIF** (varchar(9), NOT NULL): Identificador fiscal del empleado.
 - **nombre** (varchar(45), DEFAULT NULL): Nombre del empleado.
 - **apellido1** (varchar(45), DEFAULT NULL): Primer apellido del empleado.
 - **apellido2** (varchar(45), DEFAULT NULL): Segundo apellido del empleado.
 - **usuario** (varchar(45), DEFAULT NULL): Usuario asociado al empleado.
4. **Tabla habitacion:**
- Almacena la información de las habitaciones.
 - **Campos:**
 - **nhabitacion** (int(11), NOT NULL): Número de la habitación.

- **npersonas** (varchar(45), DEFAULT NULL): Capacidad de personas.
- **ncamas** (varchar(45), DEFAULT NULL): Número de camas.
- **metroscuadrados** (varchar(45), DEFAULT NULL): Metros cuadrados de la habitación.
- **estado** (varchar(20), NOT NULL): Estado de la habitación (disponible, ocupada, etc.).
- **precio** (float, NOT NULL): Precio por noche.
- **foto** (longblob, DEFAULT NULL): Foto de la habitación.
- **usuario** (varchar(45), DEFAULT NULL): Usuario responsable de la habitación.

5. Tabla **pedido**:

- Almacena la información de los pedidos realizados por los clientes.
- **Campos**:
 - **idpedido** (int(11), NOT NULL, AUTO_INCREMENT): Identificador único del pedido.
 - **nhabitacion** (int(11), DEFAULT NULL): Número de la habitación que realizó el pedido.
 - **fechapedido** (datetime, NOT NULL): Fecha y hora del pedido.
 - **estado** (varchar(20), NOT NULL): Estado del pedido (procesado, completado, etc.).
 - **total** (float, NOT NULL): Total del pedido en euros.

6. Tabla **producto**:

- Almacena la información de los productos disponibles para pedido.
- **Campos**:
 - **idproducto** (int(11), NOT NULL, AUTO_INCREMENT): Identificador único del producto.
 - **nombre** (varchar(100), NOT NULL): Nombre del producto.
 - **descripcion** (varchar(255), DEFAULT NULL): Descripción del producto.
 - **precio** (float, NOT NULL): Precio del producto.

7. Tabla **resena**:

- Almacena las reseñas realizadas por los clientes.
- **Campos**:

- **id** (int(11), NOT NULL, AUTO_INCREMENT): Identificador único de la reseña.
- **valoracion** (float, NOT NULL): Valoración otorgada.
- **observaciones** (varchar(200), NOT NULL): Observaciones de la reseña.

8. Tabla **reserva**:

- Almacena la información de las reservas realizadas.
- **Campos**:
 - **idreserva** (int(11), NOT NULL, AUTO_INCREMENT): Identificador único de la reserva.
 - **nhabitacion** (int(11), DEFAULT NULL): Número de la habitación reservada.
 - **fecha** (date, NOT NULL): Fecha de la reserva.
 - **fechaentrada** (date, DEFAULT NULL): Fecha de entrada.
 - **fechasalida** (date, DEFAULT NULL): Fecha de salida.
 - **pago** (float, NOT NULL): Pago realizado.
 - **metodopago** (varchar(32), NOT NULL): Método de pago utilizado.
 - **observaciones** (varchar(255), NOT NULL): Observaciones de la reserva.
 - **factura** (mediumblob, DEFAULT NULL): Factura generada para la reserva.

9. Tabla **reservacliente**:

- Relaciona las reservas con los clientes.
- **Campos**:
 - **idreserva** (int(11), NOT NULL): Identificador de la reserva.
 - **nifcliente** (varchar(20), NOT NULL): NIF del cliente.

10. Tabla **usuario**:

- Almacena la información de los usuarios del sistema.
- **Campos**:
 - **usuario** (varchar(45), NOT NULL): Nombre de usuario.
 - **password** (varchar(100), NOT NULL): Contraseña encriptada del usuario.
 - **tipo** (int(11), NOT NULL): Tipo de usuario (administrador, empleado, etc.).
 - **token** (varchar(64), DEFAULT NULL): Token de autenticación.

Índices y Claves Primarias

- Cada tabla tiene su clave primaria (**PRIMARY KEY**) definida para garantizar la unicidad de las filas.
- Se han añadido índices (**KEY**) para optimizar las consultas y mejorar el rendimiento.

Relaciones entre Tablas

- **detallespedido:**
 - **idpedido** referencia a **pedido(idpedido)**.
 - **idproducto** referencia a **producto(idproducto)**.
- **empleado:**
 - **usuario** referencia a **usuario(usuario)**.
- **habitacion:**
 - **usuario** referencia a **usuario(usuario)** con la opción de eliminación en cascada (**ON DELETE SET NULL**).
- **pedido:**
 - **nhabitacion** referencia a **habitacion(nhabitacion)**.
- **reserva:**
 - **nhabitacion** referencia a **habitacion(nhabitacion)**.
- **reservacliente:**
 - **idreserva** referencia a **reserva(idreserva)**.
 - **nifcliente** referencia a **cliente(NIF)**.

5.- Servidor

Este apartado detalla el proceso de instalación y configuración del servidor en Linux, incluyendo la instalación de Apache, PHP, phpMyAdmin y MySQL (MariaDB). También incluye la configuración de Apache para aceptar conexiones desde cualquier IP y la estructura del servidor para manejar las solicitudes de la aplicación.

Instalación de Apache, PHP, phpMyAdmin y MySQL

Paso 1: Actualizar los paquetes del sistema

bash

Copiar código

```
sudo apt update
```

```
sudo apt upgrade -y
```

Paso 2: Instalar Apache

bash

Copiar código

```
sudo apt install apache2 -y
```

Paso 3: Instalar PHP y extensiones necesarias

bash

Copiar código

```
sudo apt install php libapache2-mod-php php-mysql -y
```

Paso 4: Instalar MySQL (MariaDB)

bash

Copiar código

```
sudo apt install mariadb-server mariadb-client -y
```

Paso 5: Configurar MySQL

bash

Copiar código

```
sudo mysql_secure_installation
```

Sigue las instrucciones en pantalla para configurar la seguridad de MySQL (establecer contraseña root, eliminar usuarios anónimos, deshabilitar el acceso root remoto, etc.).

Paso 6: Instalar phpMyAdmin

bash

Copiar código

```
sudo apt install phpmyadmin -y
```

Durante la instalación, selecciona Apache y configura phpMyAdmin con el servidor de bases de datos. Crea una base de datos y un usuario para phpMyAdmin cuando se te solicite.

Configuración de Apache para Aceptar Conexiones Externas

Paso 1: Editar el archivo de configuración de Apache

bash

Copiar código

```
sudo nano /etc/apache2/ports.conf
```

Paso 2: Cambiar la dirección de escucha de `Listen 80` a `Listen 0.0.0.0:80`

text

Copiar código

```
Listen 0.0.0.0:80
```

Paso 3: Reiniciar Apache para aplicar los cambios

bash

Copiar código

```
sudo systemctl restart apache2
```

Estructura del Servidor

La estructura del servidor se organiza en el directorio `/var/www/html/hotel/`, con los siguientes archivos PHP para manejar diferentes operaciones de la aplicación:

- **Actualización**
 - `actualizar-cliente.php`
 - `actualizar-empleado.php`
- **Eliminación**
 - `borrar-cliente.php`
 - `borrar-empleado.php`
 - `borrarpedido.php`
 - `borrar-reserva.php`
 - `borrar-reservas-antiguas.php`
- **Cambiar Estado**
 - `cambiar-estado-hab.php`
 - `completar-pedido.php`
- **Descargas**
 - `descargar-factura.php`
 - `descargar-foto-hab.php`
- **Información**
 - `info-reserva.php`
 - `obtener-info.php`
 - `obtener-pedido.php`
 - `obtener-pedidos.php`
- **Inserciones**
 - `insertar-cliente.php`
 - `insertar-empleado.php`
 - `insertar-habitacion.php`
 - `insertar-pedido.php`

- `insertar-reserva.php`
- `insertar-usuario.php`
- **Autenticación y Sesión**
 - `login.php`
 - `usersession.php`
- **Valoración**
 - `valoracion.php`
- **Archivos Auxiliares**
 - `connection.php` (para conectar con la base de datos)
 - Carpeta `img` (para almacenar imágenes subidas)

Cada archivo PHP se conecta a la base de datos utilizando `connection.php` y valida un token de autenticación (`token_auth`) para mayor seguridad. Además, los archivos PHP devuelven una respuesta en formato JSON, que es utilizada por la aplicación Java para procesar los datos.

6.- Desarrollo Android

Para el desarrollo del aplicativo Android se utilizó el IDE propio de google llamado Android Studio. Para poder realizar la configuración correcta son necesarios varios elementos. Antes que nada, al Android trabajar con un lenguaje base Java pues se verá necesario instalar los JDK pertinentes a nuestro sistema operativo, estas distribuciones están disponibles en:

<https://www.oracle.com/java/technologies/downloads/?er=221886#java8>

Java SE subscribers have more choices

Also available for development, personal use, and to run other licensed Oracle products.

Java 8 **Java 8 Enterprise Performance Pack** **Java 11**

Java SE Development Kit 8u411

Java SE subscribers will receive JDK 8 updates until at least **December 2030**.

Manual update required for some Java 8 users on macOS.

The Oracle JDK 8 license changed in April 2019



The [Oracle Technology Network License Agreement for Oracle Java SE](#) is substantially different from prior Oracle JDK 8 licenses. This license permits certain uses, such as personal use and development use, at no cost -- but other uses authorized under prior Oracle JDK licenses may no longer be available. Please review the terms carefully before downloading and using this product. FAQs are available [here](#).

Commercial license and support are available for a low cost with Java SE Universal Subscription.

JDK 8 software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#).

Java SE 8u411 checksums and OL 8 GPG Keys for RPMs

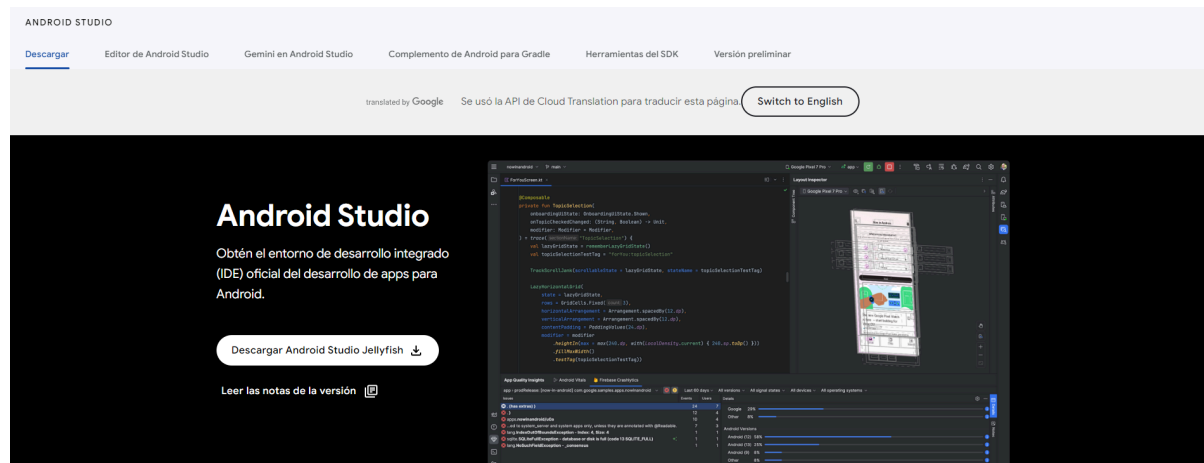
Linux **macOS** **Solaris** **Windows**

Product/file description	File size	Download
x86 Installer	141.01 MB	 jdk-8u411-windows-i586.exe
x64 Installer	150.55 MB	 jdk-8u411-windows-x64.exe

[Documentation Download](#)

Después de haber instalado el JDK correspondiente al sistema operativo, es necesario realizar la descarga y configuración del Android Studio, para esto se debe dirigir a la URL:

<https://developer.android.com/studio?hl=es-419>



Después de realizada la instalación procedemos a abrir el Android Studio y crear un nuevo proyecto. Lo primero que solicitará será el nombre que le queremos dar al proyecto y la ruta donde será almacenado.

Luego de realizada la instalación procedemos a abrir el Android Studio y abrir un proyecto. Lo que solicitará será la ruta donde está almacenado. Una vez seleccionado, se abrirá automáticamente.



Grand Hotel
+34 123 123 123
info@grandhotel.es
www.grandhotel.es

