

Aplicación conjunta

Introducción	2
Análisis	2
Requisitos funcionales	2
Requisitos no funcionales	2
Casos de uso	3
Diseño	3
Modelo entidad relación	3
Modelo de Clases UML	3
Algoritmos en pseudocódigo y/o diagramas de flujo	4
Registrar usuario	4
Crear servicio	4
Iniciar sesión	5
Codificación	5
Preparación de la base de datos	5
Implementación de la aplicación	7
Conexión con la base de datos	7
Programa principal	8
Introducir un servicio	10
Listar servicios	11
Eliminar servicios	12
Solicitar servicio	13
Pruebas	14
Explotación	14
Creación del paquete JAR	16
Distribución del paquete	16
Pruebas de operación	17
Ejecución en Windows	17
Ejecución en Linux	17

Introducción

Miembros del equipo

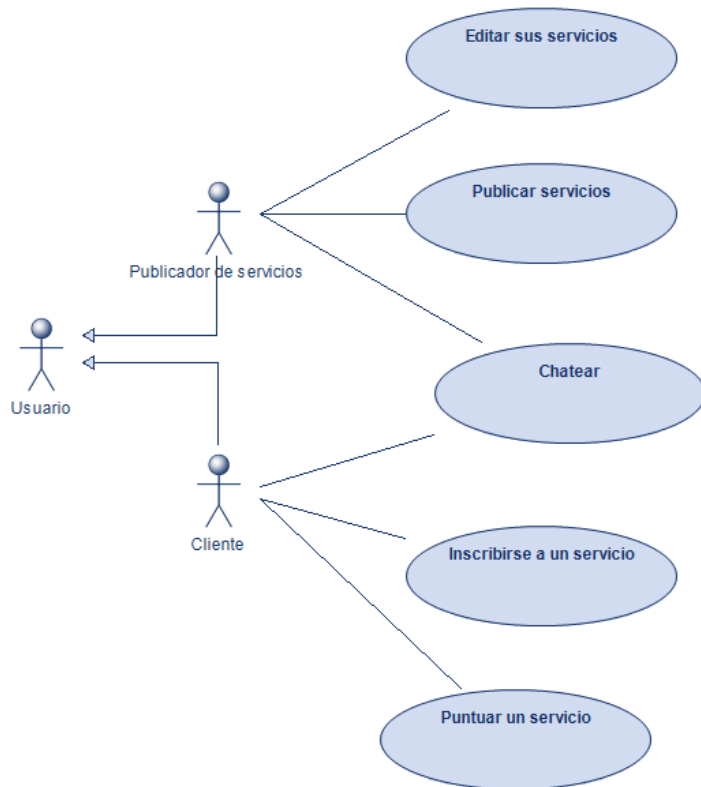
- Alejandro Feijoo Regueira
- Celia Sanmartín Chapela
- Israel Benjamín Gago Acuña
- Hugo Gómez Alonso

Análisis

Se busca crear una aplicación en la que los usuarios puedan promocionar sus servicios y permitir a otros usuarios solicitarlos.

Requisitos funcionales	Requisitos no funcionales
<ul style="list-style-type: none">• El usuario puede publicar nuevos servicios.• El usuario puede ver una lista de todos los servicios publicados.• El usuario puede acceder a los servicios publicados e inscribirse.• El usuario puede borrar o modificar sus servicios publicados.• Acceso a la ubicación de cada servicio.• El usuario tiene acceso a un chat.	<ul style="list-style-type: none">• La aplicación está disponible tanto para android como para ios.• La aplicación es intuitiva y fácil de usar.• Los datos deben estar almacenados en una base de datos.• Las contraseñas deben estar hasheadas

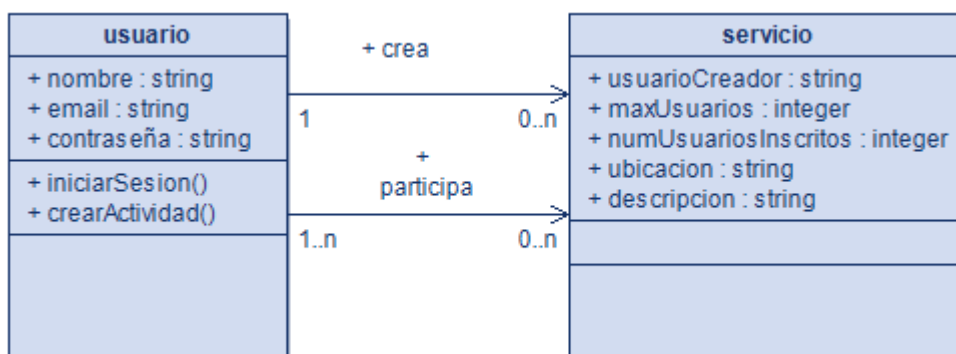
Casos de uso



Diseño

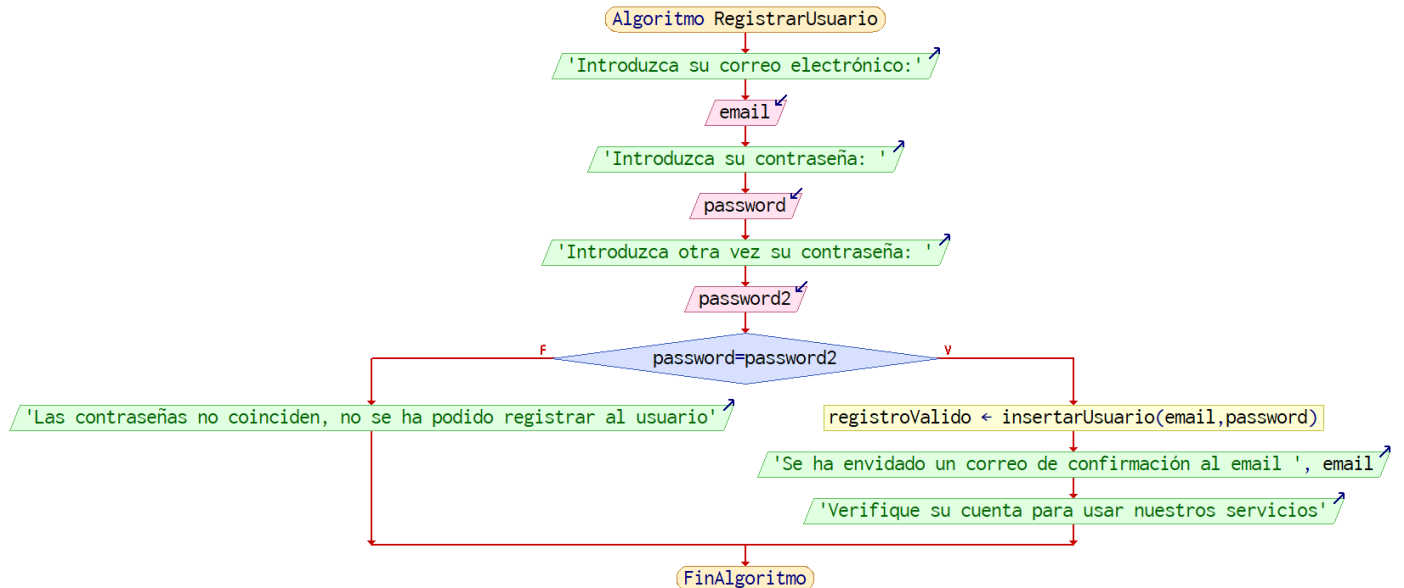
Modelo entidad relación

Modelo de Clases UML

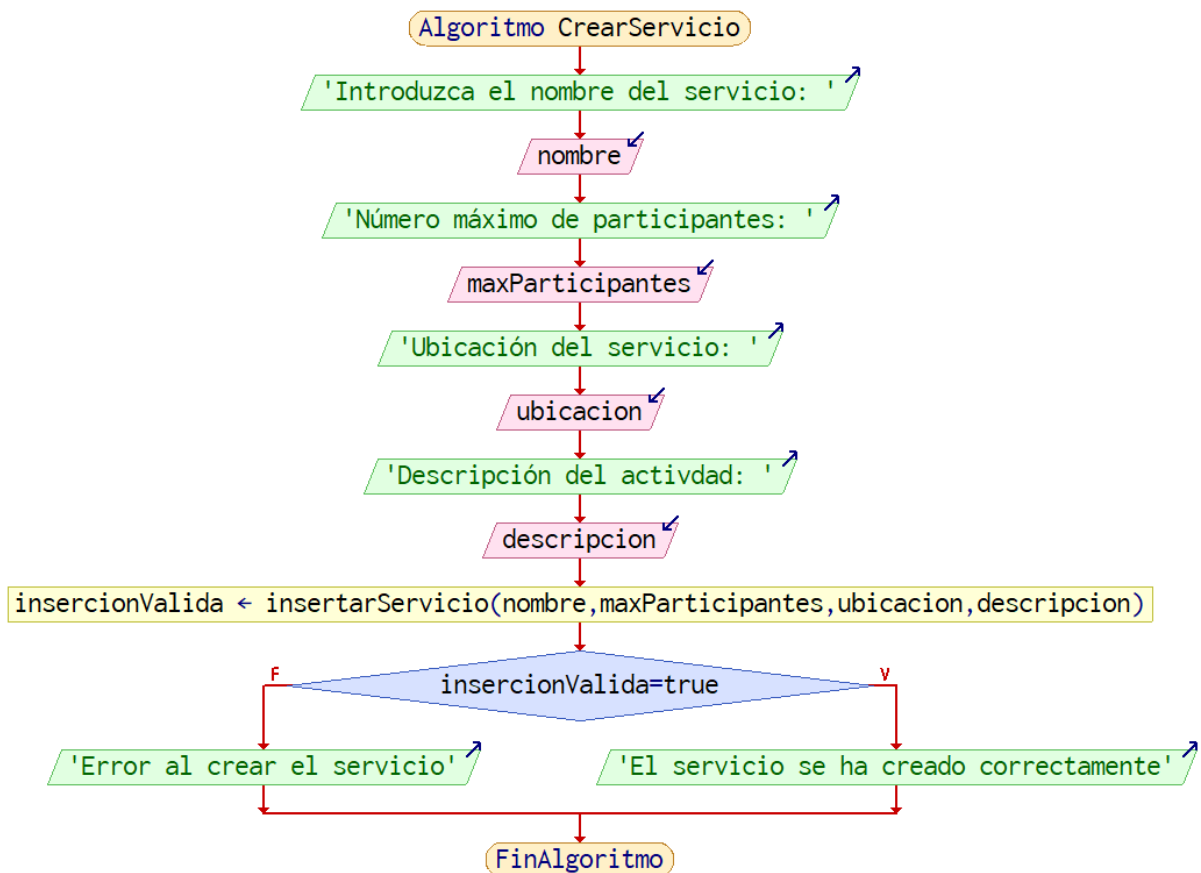


Algoritmos en pseudocódigo y/o diagramas de flujo

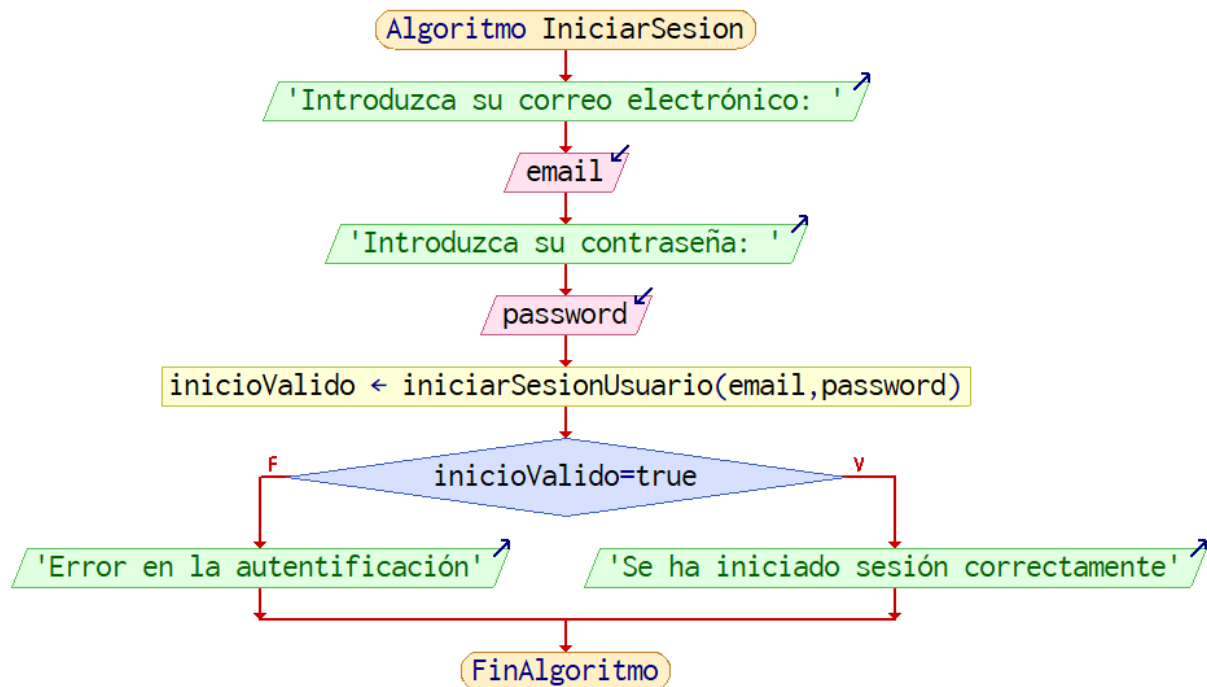
Registrar usuario



Crear servicio



Iniciar sesión



Codificación

Preparación de la base de datos

1. Instalación de XAMPP con MariaDB y phpMyAdmin
2. Creación de la base de datos “servicios”
3. Creación de la tabla “usuarios”

The screenshot shows the phpMyAdmin interface for the 'servicios' database, specifically the 'usuarios' table. The table structure is as follows:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	codUsuario	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	usuario	varchar(50)	utf8mb4_spanish2_ci		No	Ninguna			Cambiar Eliminar Más
3	password	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna			Cambiar Eliminar Más
4	fechaAlta	timestamp			No	current_timestamp()			Cambiar Eliminar Más
5	admin	tinyint(1)			No	0			Cambiar Eliminar Más

4. Creación de la tabla “servicios”

Servidor: 127.0.0.1 » Base de datos: servicios » Tabla: servicios									
<div> <div>Examinar</div> <div>Estructura</div> <div>SQL</div> <div>Buscar</div> <div>Insertar</div> <div>Exportar</div> <div>Importar</div> <div>Privilegios</div> <div>Operaciones</div> <div>Seguimiento</div> </div>									
<div> <div>Estructura de tabla</div> <div>Vista de relaciones</div> </div>									
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	codServicio	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
<input type="checkbox"/> 2	codUsuario	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	titulo	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 4	fechaCreacion	timestamp			No	current_timestamp()			Cambiar Eliminar Más
<input type="checkbox"/> 5	descripcion	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 6	ubicacion	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 7	numUsuariosActuales	int(11)			No	0			Cambiar Eliminar Más
<input type="checkbox"/> 8	maxUsuarios	int(11)			No	Ninguna			Cambiar Eliminar Más

5. Creación de la tabla “usuarios_servicios”

Servidor: 127.0.0.1 » Base de datos: appconjunta » Tabla: usuarios_servicios									
<div> <div>Examinar</div> <div>Estructura</div> <div>SQL</div> <div>Buscar</div> <div>Insertar</div> <div>Exportar</div> <div>Importar</div> <div>Privilegios</div> <div>Operaciones</div> </div>									
<div> <div>Estructura de tabla</div> <div>Vista de relaciones</div> </div>									
#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/> 1	codUsuario	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 2	codServicio	int(11)			No	Ninguna			Cambiar Eliminar Más
<input type="checkbox"/> 3	fechaInsercion	timestamp			No	current_timestamp()			Cambiar Eliminar Más

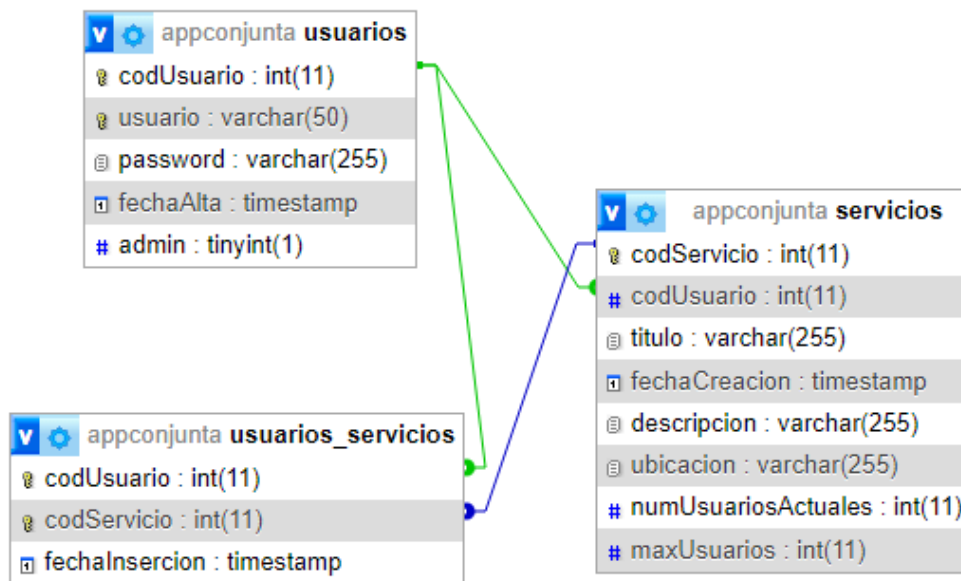
6. Creación de las claves foráneas.

En el apartado “SQL” introduzco y ejecuto el siguiente texto

```
ALTER TABLE servicios ADD CONSTRAINT FK_servicios_usuarios
FOREIGN KEY (codUsuario) REFERENCES usuarios (codUsuario);

ALTER TABLE usuarios_servicios ADD CONSTRAINT
FK_USUARIOS_USUARIOS FOREIGN KEY (codUsuario) REFERENCES
usuarios (codUsuario);

ALTER TABLE usuarios_servicios ADD CONSTRAINT
FK_SERVICIOS_SERVICIOS FOREIGN KEY (codServicio) REFERENCES
servicios (codServicio);
```



[Exportación de la base de datos en SQL](#)

Implementación de la aplicación

Implementación de algoritmos y clases en el lenguaje de programación Java.

Repositorio en [github](#).

Conexión con la base de datos

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Conexion {
    static final String HOST = "6gpe8.h.files.io";
    static final String DATABASE = "appConjunta_turndance";
    static final String USER = "appConjunta_turndance";
    static final String PASSWORD =
"6264969cbd635585af7ef3ea17a3b74b1348774d";
    static final String PORT = "3305";

    /**
     * Conecta con la base de datos
     *
     * @return Conexión con la base de datos
     */
}
```

```

    */
    public static Connection conectar() {
        Connection con = null;

        String url = "jdbc:mysql://" + Conexion.HOST + ":" +
Conexion.PORT + "/" + Conexion.DATABASE;

        try {
            con = DriverManager.getConnection(url, Conexion.USER,
Conexion.PASSWORD);
        } catch (SQLException e) {
            System.out.println("Error al conectar con la BD.");
        }

        return con;
    }
}

```

Programa principal

```

import java.util.Scanner;

public class Principal {
    static Scanner sc = new Scanner(System.in);

    static int menu() {
        boolean respuestaValida = false;
        int respuesta = -1;

        while (!respuestaValida) {
            try {
                System.out.println("1. LISTAR TODOS LOS SERVICIOS");
                System.out.println("2. LISTAR MIS SERVICIOS");
                System.out.println("3. LISTAR SERVICIOS
SOLICITADOS");
                System.out.println("4. AÑADIR SERVICIO");
            }
        }
    }
}

```



```

        System.out.println("5. SOLICITAR SERVICIO");
        System.out.println("6. ELIMINAR UNO DE MIS
SERVICIOS");

        System.out.println("7. CERRAR SESIÓN");
        System.out.println("0. SALIR");
        System.out.println("-----");
        System.out.print("Respuesta: ");
        respuesta = sc.nextInt();
        sc.nextLine();
        respuestaValida = true;
    } catch (Exception e) {
        System.out.println("ERROR: LA RESPUESTA DEBE SER UN
NÚMERO ENTERO.");
        sc.nextLine();
    }
}

return respuesta;

}

public static void main(String[] args) {
    int codUsuario = -1;
    int respuesta = -1;
    while (codUsuario == -1) {
        codUsuario = UsuariosBD.iniciarSesion();

        while (respuesta != 0 && codUsuario != -1) {
            respuesta = menu();
            switch (respuesta) {
                case 0:
                    System.out.println("SALIENDO DEL PROGRAMA");
                    break;
                case 1:
                    Servicios.listarServicios();
                    break;
                case 2:
                    Servicios.listarServicios(codUsuario);
                    break;
                case 3:
                    Servicios.listarServiciosSolicitados(codUsuario);
                    break;
            }
        }
    }
}

```

```

        case 4:
            Servicios.crearServicio(codUsuario);
            break;
        case 5:
            Servicios.solicitarServicio(codUsuario);
            break;
        case 6:
            Servicios.eliminarMiServicio(codUsuario);
            break;
        case 7:
            System.out.println("CERRANDO SESIÓN");
            codUsuario = -1;
            break;
        default:
            System.out.println("Respuesta inválida.");
            break;
    }
}
}
}
}
}

```

Introducir un servicio

```

public static boolean crearServicio(int codUsuario) {
    Connection conexion = Conexion.conectar();
    Statement sentencia;
    try {
        System.out.println("Crear servicio");
        sentencia = conexion.createStatement();
        System.out.print("Título: ");
        String titulo = System.console().readLine();
        System.out.print("Descripción: ");
        String descripcion = System.console().readLine();
        System.out.print("Ubicación: ");
        String ubicacion = System.console().readLine();
        System.out.print("Número máximo de personas: ");
        String maxUsuarios = System.console().readLine();
    }
}

```

```

        String sql = "INSERT INTO servicios
(codUsuario,titulo,descripcion,ubicacion,maxUsuarios) VALUES ('"
        + codUsuario + "','" + titulo + "','" +
descripcion + "','" + ubicacion + "','" + maxUsuarios
        + "')";

        int resultado = sentencia.executeUpdate(sql);
        sentencia.close();
        conexion.close();
        if (resultado == 1) {
            System.out.println("Servicio insertado con éxito.");
        }
        return resultado == 1;
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
}

```

Servidor: 127.0.0.1 » Base de datos: servicios » Tabla: servicios

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento Disparadores

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0003 segundos.)

SELECT * FROM `servicios`

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

	codServicio	codUsuario	titulo	fechaCreacion	descripcion	ubicacion	numUsuariosActuales	maxUsuarios
<input type="checkbox"/> Editar <input type="text" value="Copiar"/> <input type="text" value="Borrar"/>	1	1	servicio 1	2024-11-19 13:04:34	Prueba de inserción del primer servicio	Seixo	0	7

Listar servicios

```

public static void listarServicios() {
    Connection conexion = Conexion.conectar();

    Statement sentencia;
    try {
        sentencia = conexion.createStatement();
    }
}

```

```

        ResultSet resultado = sentencia.executeQuery("SELECT
codServicio,usuario,fechaCreacion,titulo FROM servicios inner join
usuarios on servicios.codUsuario = usuarios.codUsuario");

        while (resultado.next()) {
            // Procesa los datos
            int codServicio = resultado.getInt("codServicio");
            String usuario = resultado.getString("usuario");
            Timestamp fechaCreacion =
resultado.getTimestamp("fechaCreacion");
            String titulo = resultado.getString("titulo");

            // Procesa los datos
            System.out.println(
                "Código del servicio: " + codServicio + ",
título del servicio: "+titulo +", usuario creador: " + usuario + ",
fecha de creación: " + fechaCreacion);
        }

        System.out.println("-----
-----");

        resultado.close();
        sentencia.close();
        conexion.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

Eliminar servicios

```

public static boolean eliminarMiServicio(int codUsuario) {
    Connection conexion = Conexion.conectar();
    Statement sentencia;
    listarServicios(codUsuario);
    System.out.print("Introduce el código de servicio del
servicio que quieres eliminar: ");
    try {
        sentencia = conexion.createStatement();
    }
}

```

```

        String codServicio = System.console().readLine();
        String sql = "Delete FROM servicios where codServicio =
"+codServicio+" AND codUsuario = "+codUsuario+"";
        int resultado = sentencia.executeUpdate(sql);
        sentencia.close();
        conexion.close();
        if (resultado == 1) {
            System.out.println("Servicio eliminado con exito.");
        } else {
            System.out.println("Error al intentar eliminar el
servicio");
        }

System.out.println("-----");
        return resultado == 1;
    } catch (Exception e) {
        e.printStackTrace();
        return false;
    }
}

```

Solicitar servicio

```

public static boolean solicitarServicio(int codUsuario) {
    Connection conexion = Conexion.conectar();
    Statement sentencia;
    try {
        listarServiciosNoSolicitados(codUsuario);
        sentencia = conexion.createStatement();
        System.out.print("Código del servicio que quiere
solicitar (0 para cancelar): ");
        String codServicio = System.console().readLine();

        if (codServicio.equals("0")) {
            return false;
        }

        String sql = "INSERT INTO usuarios_servicios
(codUsuario,codServicio) VALUES ('"
            + codUsuario + "','" + codServicio + "')";
    }
}

```

```

        int resultado = sentencia.executeUpdate(sql);
        sentencia.close();
        conexion.close();
        if (resultado == 1) {
            System.out.println("Servicio solicitado con éxito.");
        }
        System.out.println("-----");
        return resultado == 1;
    } catch (SQLException e) {
        // e.printStackTrace();
        System.out.println("ERROR: no se ha podido solicitar el
servicio");
        return false;
    }
}

```

Pruebas

Explotación

Se pondrá la base de datos en un servidor compartido en línea. Luego se empaquetará la aplicación en un archivo JAR y se pondrá a disposición de los usuarios.

Para ejecutar la aplicación en plataformas Windows o Linux se debe disponer de una máquina virtual Java y ejecutar el siguiente comando

```
java -jar servicios.jar.
```

Creación de la base de datos



Base de datos en línea alojada en filess.io.



MariaDB v10.7.4 Status Up

Web Client Code

Size

0.00 MB / 10 MB

Request Backup

Delete AppConjunta

appConjunta

Host

6gpe8.h.filess.io

Database

appConjunta_tu

User

appConjunta_tu

Port

3305

Do not drop this database. It won't be included in the backup if it does not exists.

Password

.....

Connection formats

MariaDB URI

mariadb://appConjunta_turndance:.....@6gpe8.h.filess.io:3305/appCo

MariaDB login command

mariadb -u appConjunta_turndance -P 3305 -p..... -h 6gpe8.h.filess.io :

Backups

☰

🗄️

CONNECTIONS

Search connection or database

appConjunta_turndance on 6gpe8.h.filess.io - unsa

🗄️+

Add new connection

TABLES, VIEWS, FUNCTIONS

Search in tables, objects, # prefix in columns

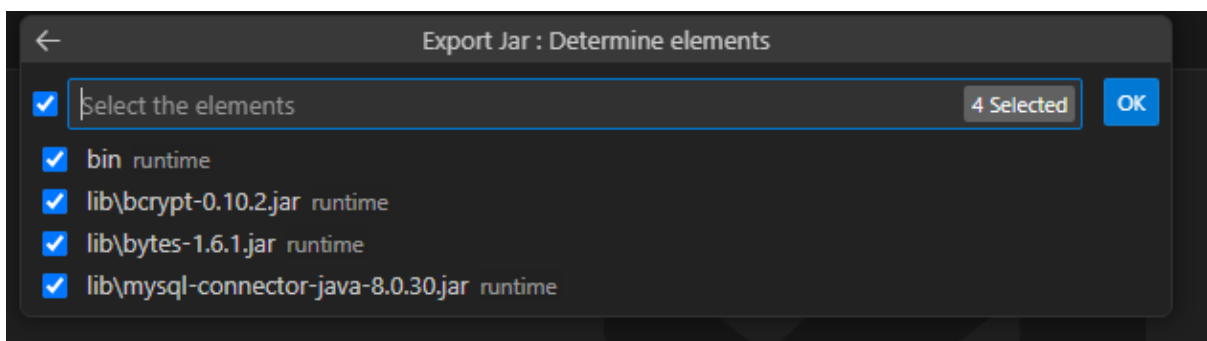
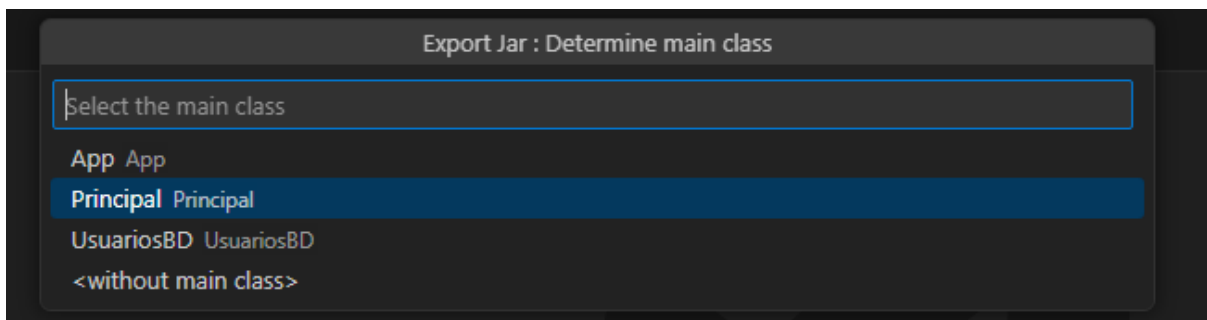
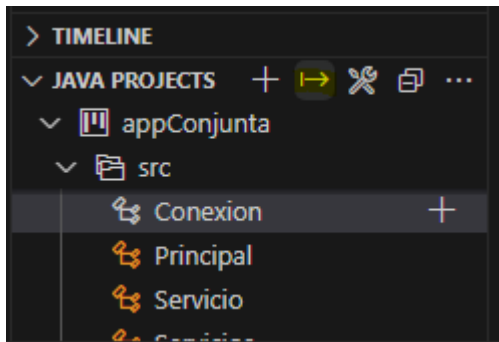
🗄️ Tables (3)

> 🗄️ servicios 3 rows

> 🗄️ usuarios 2 rows

> 🗄️ usuarios_servicios 1 rows

Creación del paquete JAR



Replicar el proceso para crear un jar para la gestión de usuarios

Distribución del paquete

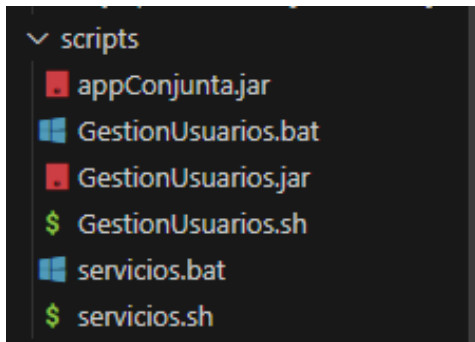
Además de los paquetes JAR se incluirán también los scripts para ejecutar del program desde entornos de escritorio Linux y Windows

servicios.bat

```
java -jar servicios.jar  
pause
```

servicios.sh

```
#!/bin/bash  
java -jar servicios.jar  
echo "Fin"  
read
```

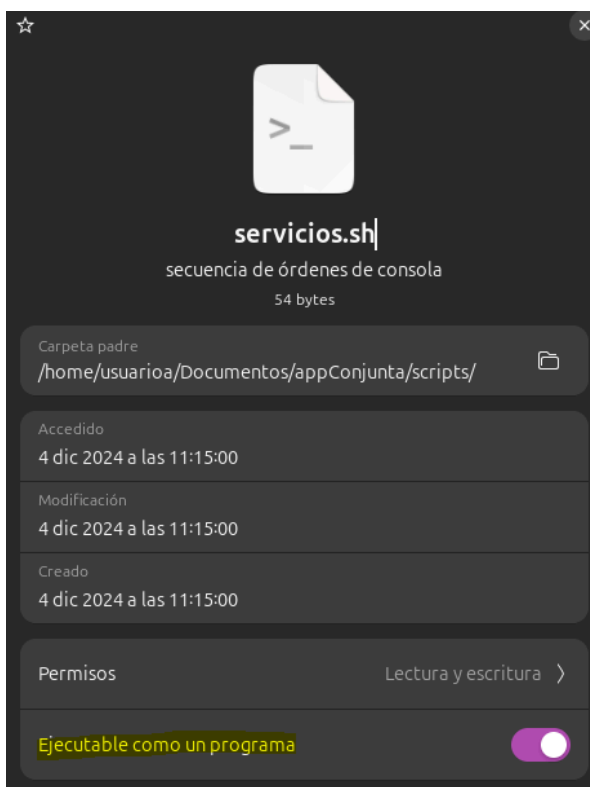
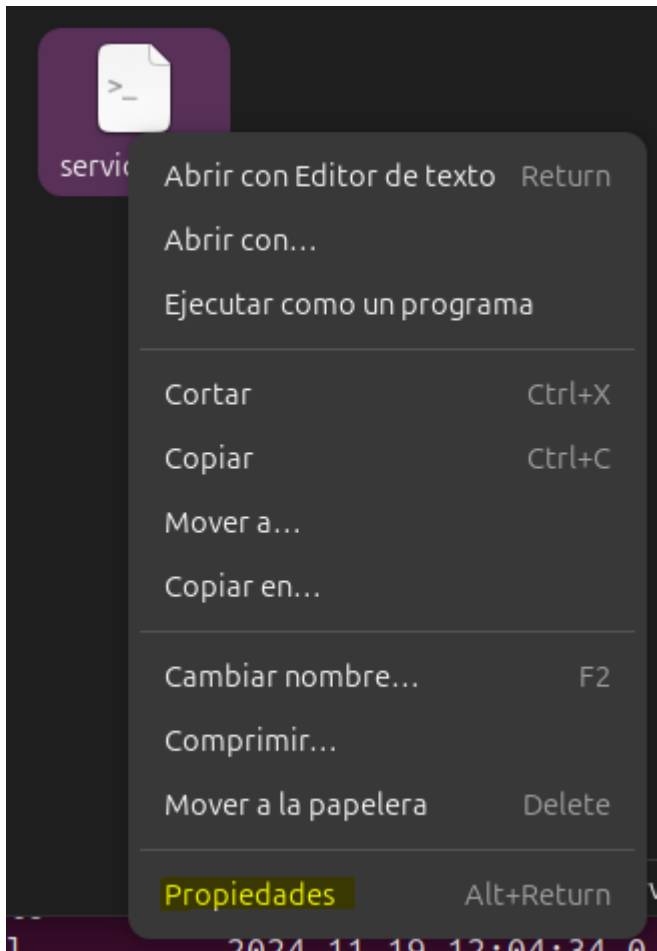
Pruebas de operación

Ejecución en Windows

```
Support... 2138ms
C:\Windows\system32\cmd.exe
C:\Users\usuarioa\Documents\appConjunta\scripts>java -jar appConjunta.jar
LOGIN DE USUARIO
Usuario: israel
Contraseña:
1. LISTAR TODOS LOS SERVICIOS
2. LISTAR MIS SERVICIOS
3. LISTAR SERVICIOS SOLICITADOS
4. AÑADIR SERVICIO
5. SOLICITAR SERVICIO
6. ELIMINAR UNO DE MIS SERVICIOS
7. CERRAR SESIÓN
8. SALIR
-----
Respuesta: 1
or codServicio    titulo    usuario    fecha de creacion
1      servicio 1    israel     2024-11-19 12:04:34.0
3      Clases de inglés    israel     2024-12-04 09:31:33.0
4      Limpiar coches    admin     2024-12-04 09:33:52.0
-----
1. LISTAR TODOS LOS SERVICIOS
2. LISTAR MIS SERVICIOS
3. LISTAR SERVICIOS SOLICITADOS
4. AÑADIR SERVICIO
5. SOLICITAR SERVICIO
6. ELIMINAR UNO DE MIS SERVICIOS
7. CERRAR SESIÓN
8. SALIR
-----
Respuesta:
```

Ejecución en Linux

Para ejecutar el software primero hay que dar permisos de ejecución a los scripts con los siguientes comandos.





LOGIN DE USUARIO

Usuario: israel

Contraseña:

1. LISTAR TODOS LOS SERVICIOS
2. LISTAR MIS SERVICIOS
3. LISTAR SERVICIOS SOLICITADOS
4. AÑADIR SERVICIO
5. SOLICITAR SERVICIO
6. ELIMINAR UNO DE MIS SERVICIOS
7. CERRAR SESIÓN
0. SALIR

Respuesta: 1

codServicio	titulo	usuario	fecha de creacion
1	servicio 1	israel	2024-11-19 12:04:34.0
3	Clases de inglés	israel	2024-12-04 09:31:33.0
4	Limpiar coches	admin	2024-12-04 09:33:52.0

1. LISTAR TODOS LOS SERVICIOS
2. LISTAR MIS SERVICIOS
3. LISTAR SERVICIOS SOLICITADOS
4. AÑADIR SERVICIO
5. SOLICITAR SERVICIO
6. ELIMINAR UNO DE MIS SERVICIOS
7. CERRAR SESIÓN
0. SALIR

Respuesta: