

Desarrollo web entorno  
servidor

# Hito individual

---

Sergio Felipe García

8 de diciembre de 2023

<b>Tarea 1</b>	<b>2</b>
Definición del problema:	2
Requisitos del usuario:	2
Áreas críticas:	3
Disponibilidad y confiabilidad:	3
Solución propuesta	3
Seguridad de datos y privacidad:	3
Solución propuesta	4
Solución propuesta	4
Solución propuesta	5
Solución propuesta	5
Mantenimiento y soporte:	5
Solución propuesta	5
Modelo entidad relación	6
Esquema Básico	6
Creación del Modelo de BBDD.	8
Primeros datos en el sistema	9
<b>Tarea 2</b>	<b>10</b>
Entorno de Desarrollo para producir la app.	10
Estructura del proyecto.	11
Oportunidades de Mejora	11
Diseño y Estilos	12
Página de Bienvenida al Servicio.	13
Panel del Operador	15
Especialistas	17
El especialista podrá seleccionar la llamada que desee resolver y modificar los datos con la resolución.	18
Oportunidades de Mejora	18
Modificación del caso	19

## Tarea 1

### Definición del problema:

La empresa de gestión de centros de atención telefónica ha obtenido un contrato para brindar el servicio médico telefónico 111, el contrato exige una cobertura los 365 días del año y las 24 horas al día. El desafío es desarrollar una aplicación de software que permita la operación eficiente de este servicio, gestionando llamadas de personas que requieren atención médica y seguimiento, involucrando operadores, enfermeras, médicos y psicólogos en turnos específicos. La aplicación deberá registrar datos de llamadas, tomar decisiones de derivación, ofrecer consejos, gestionar llamadas molestas y proporcionar informes detallados de desempeño.

### Requisitos del usuario:

**Cobertura:** La aplicación debe estar disponible los 365 días del año y las 24 horas del día para atender las llamadas médicas.

**Gestión de turnos:** La aplicación debe permitir la programación y asignación de operadores, enfermeras, médicos y psicólogos en los siguientes turnos:

- Lunes a viernes no festivos de 7:00 a 15:00.
- Lunes a viernes no festivos de 15:00 a 23:00.
- Lunes a viernes no festivos de 23:00 a 7:00.
- Sábados, domingos y festivos de 7:00 a 15:00.
- Sábados, domingos y festivos de 15:00 a 23:00.
- Sábados, domingos y festivos de 23:00 a 7:00.

**Registro de llamadas:** La aplicación debe permitir el registro de cada llamada, incluyendo el número llamante, el operador que la atiende y los datos del paciente (nombre y DNI).

**Interrogatorio y toma de decisiones:** La aplicación debe permitir a los operadores interrogar a la persona llamante y tomar decisiones basadas en el tipo de llamada, que incluyen dar consejos o derivar la llamada a un médico, enfermera o psicólogo.

**Registro de acciones:** La aplicación debe registrar quién atendió la llamada, a quién se derivó (si corresponde) y qué consejo se ofreció al paciente. También se debe registrar si la llamada es molesta o una broma.

**Sistema de seguimiento y gestión:** La aplicación debe proporcionar la capacidad de generar informes, incluyendo un resumen de trabajo del personal y estadísticas como el número de llamadas atendidas por cada operador y el número de llamadas derivadas a especialistas (médicos, enfermeras o psicólogos).

**Seguridad de datos:** Se debe garantizar la seguridad y privacidad de los datos de los pacientes y la integridad de la información registrada.

**Mantenimiento y soporte:** Garantizar el mantenimiento continuo de la aplicación y brindar soporte técnico para resolver problemas y realizar actualizaciones.

## Áreas críticas:

### Disponibilidad y confiabilidad:

El servicio debe estar disponible las 24 horas del día, los 365 días del año, cualquier fallo en el sistema resultará en una falta de atención médica en momentos críticos.

### Solución propuesta

Para solucionarlo desplegaría el desarrollo en la nube. En servicios del tipo Amazon Web Services, Azure o Google cloud

Redundancia: Mi servicio estaría disponible 24x365 porque Amazon proporciona redundancia de los datos allí alojados una vez que lo contratas. Si en un momento deja de prestar servicio se activa el sistema redundado.

Escalabilidad Automática: Con AWS, las empresas pueden escalar automáticamente sus recursos de computación, almacenamiento y red según la demanda. Esto asegura que puedan manejar cargas de trabajo variables sin interrupciones en el servicio.

### Seguridad de datos y privacidad:

Cómo se van a manejar los datos de los pacientes, hay que garantizar la seguridad y privacidad de la información médica. Un fallo en la seguridad mostrará información sensible de los pacientes y la empresa tendrá graves implicaciones legales y éticas. La información sensible se debería encriptar para que quedase anonimizada.

## Solución propuesta

Encriptación: Para solucionar este problema yo implementaría un sólido sistema de encriptación para proteger la información sensible de los pacientes tanto en almacenamiento con algoritmos tipo hash como en tránsito utilizando un protocolo https. [https://docs.aws.amazon.com/es\\_es/AmazonS3/latest/userguide/UsingEncryption.html](https://docs.aws.amazon.com/es_es/AmazonS3/latest/userguide/UsingEncryption.html)

Control de accesos: AWS proporciona herramientas de gestión de identidad y acceso que permiten a las empresas controlar quién tiene acceso a sus recursos y datos. Esto es esencial para garantizar la seguridad.

[https://docs.aws.amazon.com/es\\_es/IAM/latest/UserGuide/access\\_controlling.html](https://docs.aws.amazon.com/es_es/IAM/latest/UserGuide/access_controlling.html)

Cumplimiento: AWS implementa medidas de seguridad avanzadas y cumple con numerosos estándares y certificaciones de seguridad. Esto ayuda a proteger los datos y a cumplir con las regulaciones como por ejemplo (La ley de protección de datos).

<https://aws.amazon.com/es/compliance/eu-data-protection/>

Backups y Almacenamiento Duradero: AWS ofrece servicios de almacenamiento duradero y opciones de copia de seguridad automatizadas. Esto ayuda a garantizar la integridad y la disponibilidad de los datos, incluso en situaciones de pérdida de datos o fallas. <https://aws.amazon.com/es/backup/>

## **Toma de decisiones de los operadores:**

La toma de decisiones de los operadores es crucial para determinar si una llamada se deriva a un médico, enfermera o psicólogo. En la que voy a desarrollar aplicación no crearé un protocolo al no tener criterios medicos pero sería necesario para guiar a los operadores en la toma de decisiones, este protocolo que lo debe dictar un medico evitaría graves consecuencias para la atención médica de los pacientes.

## Solución propuesta

Documentación: Para optimizar la toma de decisiones de los operadores en el servicio médico telefónico 111, es esencial implementar protocolos claros y detallados, proporcionando una guía estructurada para evaluar situaciones médicas.

## **Registro y seguimiento de llamadas:**

Si no hay un registro preciso y seguimiento de cada llamada se van a perder datos críticos y no podremos rastrear el historial de atención de los pacientes. Esto afectará

negativamente a la calidad de la atención y a la capacidad de generar informes precisos.

#### Solución propuesta

Para solucionar el problema podría conectar el sistema de registro de llamadas con el sistema de gestión de pacientes. Todo quedaría registrado en la BBDD para que la información de las llamadas se asocia automáticamente con el historial del paciente.

#### **Gestión de llamadas molestas o falsas:**

Si no gestionamos bien las llamadas se podrá abrumar al sistema y agotar los recursos de atención médica, retrasando o impidiendo la atención a llamadas legítimas.

#### Solución propuesta

Para solucionar el problema puede utilizar un sistema de filtrado de llamadas para identificar y bloquear llamadas molestas o fraudulentas empleando tecnologías de reconocimiento de voz, análisis de patrones de llamadas y listas negras para filtrar llamadas no deseadas.

#### **Mantenimiento y soporte:**

No tener un mantenimiento continuo y un soporte técnico efectivo podrá resultar en tiempos de inactividad y no poder enfrentar problemas críticos de manera oportuna.

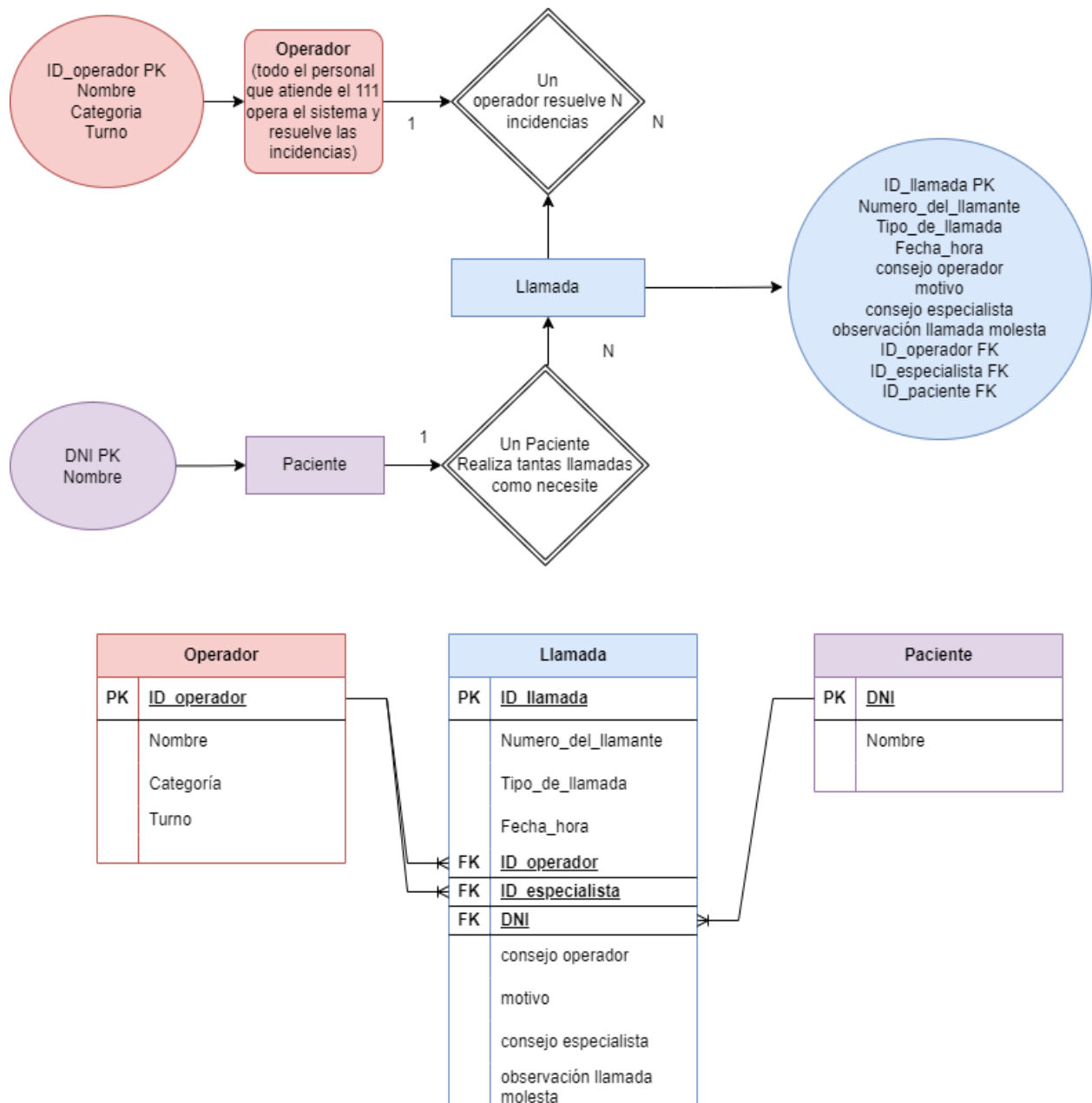
#### Solución propuesta

Para solucionar el problema habrá que implementar un programa de mantenimiento preventivo regular y establecer un equipo de soporte técnico dedicado para abordar problemas de manera oportuna y eficiente, garantizando así la disponibilidad continua y la resolución rápida de cualquier incidencia crítica en la aplicación de atención médica.

Recuperación ante desastres: AWS proporciona herramientas y servicios para implementar planes de recuperación ante desastres. La replicación de datos y la capacidad de lanzar instancias en diferentes regiones contribuyen a la continuidad del negocio en caso de eventos adversos. <https://aws.amazon.com/es/disaster-recovery/>

## Modelo entidad relación

### Esquema Básico



## **Operador**

Un operador es aquel que opera el servicio independientemente de su rol (teleoperador, enfermera, medico o psiquiatra).Cada operador puede resolver múltiples llamadas a lo largo del tiempo (uno a muchos).

### **Paciente (relación uno a muchos):**

Cada llamada está asociada a un único paciente. Un paciente puede estar asociado a varias llamadas en diferentes momentos dependiendo de sus necesidades y dolencias. Un paciente puede tener múltiples llamadas a lo largo del tiempo.

### **Registro de Llamada está relacionado con Llamada (relación uno a uno):**

Cada registro de llamada está vinculado a una única llamada. Cuando un paciente llama este es respondido por un operador, que realiza un registro previo con los datos y el motivo de la llamada del paciente. Si el operador puede resolver la llamada la dara por cerrada y esa llamada quera registrada en ese momento.

En el caso que desee derivar la llamada a un operador (especialista ) sea medico, enfermera o psiquiatra, este podra modificar el registro de la llamada para concluirla.

Por lo tanto, está vinculado a un único paciente, un único operador y un único especialista.

### **Registro de Llamada está relacionado con Operador (relación uno a muchos):**

La clave foránea sería el ID de operador en la entidad Registro de Llamada, que referencia la PK ID de operador en la entidad Operadora.

Esto permitirá que un operador esté asociado con varios registros de llamadas a lo largo del tiempo.



## Creación del Modelo de BBDD.

Para la realización de este hito he utilizado una BBDD relacional MySQL, a continuación muestro el código para la creación de tablas en la base de datos.

```
drop database hito;
create database hito;
use hito;

CREATE TABLE Operador (
  ID_operador INT PRIMARY KEY,
  nombre VARCHAR(255),
  categoria VARCHAR(50),
  turno VARCHAR(50)
);

CREATE TABLE Paciente (
  DNI VARCHAR(9) PRIMARY KEY,
  Nombre VARCHAR(255)
);

CREATE TABLE Llamada (
  ID_llamada INT PRIMARY KEY,
  numero_del_llamante VARCHAR(20),
  Tipo_de_llamada VARCHAR(50),
  Fecha_hora DATETIME,
  ID_operador INT,
  ID_especialista INT,
  DNI_paciente varchar(9),
  consejo_operador TEXT,
  motivo VARCHAR(255),
  consejo_especialista TEXT,
  observacion_llamada_molesta BOOLEAN,
  FOREIGN KEY (ID_operador) REFERENCES Operador(ID_operador),
  FOREIGN KEY (ID_especialista) REFERENCES Operador(ID_operador),
  FOREIGN KEY (DNI_paciente ) REFERENCES Paciente(DNI)
);
```

## Primeros datos en el sistema

```
INSERT INTO Operador (ID_operador, nombre, categoria, turno) VALUES
(1, 'Juan Perez', 'Operador', 'Lunes a viernes no festivos 7:00 a 15:00'),
(2, 'María García', 'Operador', 'Lunes a viernes no festivos 15:00 a 23:00'),
(3, 'Carlos Rodríguez', 'Operador', 'Lunes a viernes no festivos 23:00 a 7:00'),
(4, 'Sara Lopez', 'Enfermera', 'Sábados, domingos y festivos 7:00 a 15:00'),
(5, 'Roberto Fernández', 'Enfermera', 'Sábados, domingos y festivos 15:00 a 23:00'),
(6, 'Luisa Martín', 'Enfermera', 'Sábados, domingos y festivos 23:00 a 7:00'),
(7, 'David Herrera', 'Médico', 'Lunes a viernes no festivos 7:00 a 15:00'),
(8, 'Ana María Gómez', 'Médico', 'Lunes a viernes no festivos 15:00 a 23:00'),
(9, 'Javier Pérez', 'Médico', 'Lunes a viernes no festivos 23:00 a 7:00'),
(10, 'Isabel Torres', 'Psicólogo', 'Sábados, domingos y festivos 7:00 a 15:00');

-- Pacientes
INSERT INTO Paciente (DNI, Nombre) VALUES
('12345678A', 'Ana Martínez'),
('98765432B', 'Pedro Sánchez'),
('56789012C', 'Laura Ruiz'),
('23456789D', 'Carlos González'),
('34567890E', 'Elena Rodríguez'),
('45678901F', 'Miguel Martínez'),
('56789012G', 'Raquel Pérez'),
('67890123H', 'Francisco Gómez'),
('78901234I', 'Lucía Sánchez'),
('89012345J', 'Diego Fernández');

-- Llamadas (ID de 1 a 10)
INSERT INTO Llamada (ID_llamada, numero_del_llamante, Tipo_de_llamada, Fecha_hora, ID_operador, ID_especialista, DNI_paciente, consejo_operador, motivo, c
(1, '555-7777', 'Enfermera', '2023-12-02 14:45:00', 2, 2, '12345678A', 'Aplicar desinfectante en la herida', 'Corte en el dedo', 'Agendar cita con la enfe
(2, '555-6666', 'Médica', '2023-12-03 16:30:00', 3, 1, '98765432B', 'Tomar medicamento según prescripción', 'Dolor de cabeza', 'Reposo y tomar agua', fals
(3, '555-5555', 'Psicológica', '2023-12-04 10:15:00', 4, 3, '56789012C', 'Escuchar música relajante', 'Estrés laboral', 'Programar una cita para sesión ps
(4, '555-4444', 'Médica', '2023-12-05 08:00:00', 1, NULL, '12345678A', NULL, 'Llamada de broma', NULL, true),
(5, '555-3333', 'Médica', '2023-12-06 11:45:00', 3, 1, '56789012C', 'Descansar y aplicar hielo', 'Lesión en la pierna', 'Visitar a la enfermera para revis
(6, '555-2222', 'Psicológica', '2023-12-07 14:30:00', 4, 3, '98765432B', 'Practicar técnicas de relajación', 'Estrés laboral', 'Agendar sesión con el psic
(7, '555-1111', 'Enfermera', '2023-12-08 09:15:00', 5, 2, '23456789D', 'Aplicar vendaje en la herida', 'Quemadura leve', 'Agendar revisión con la enfermer
(8, '555-0000', 'Médica', '2023-12-09 15:45:00', 6, 1, '45678901F', 'Descansar y mantener la pierna elevada', 'Torcedura en el tobillo', 'Agendar cita con
(9, '555-9999', 'Enfermera', '2023-12-10 12:30:00', 9, 2, '67890123H', 'Aplicar crema en la picadura', 'Picadura de insecto', 'Agendar revisión con la enf
(10, '555-8888', 'Molestia', '2023-12-11 07:00:00', 10, NULL, '89012345J', NULL, 'Llamada de broma', NULL, true);
```

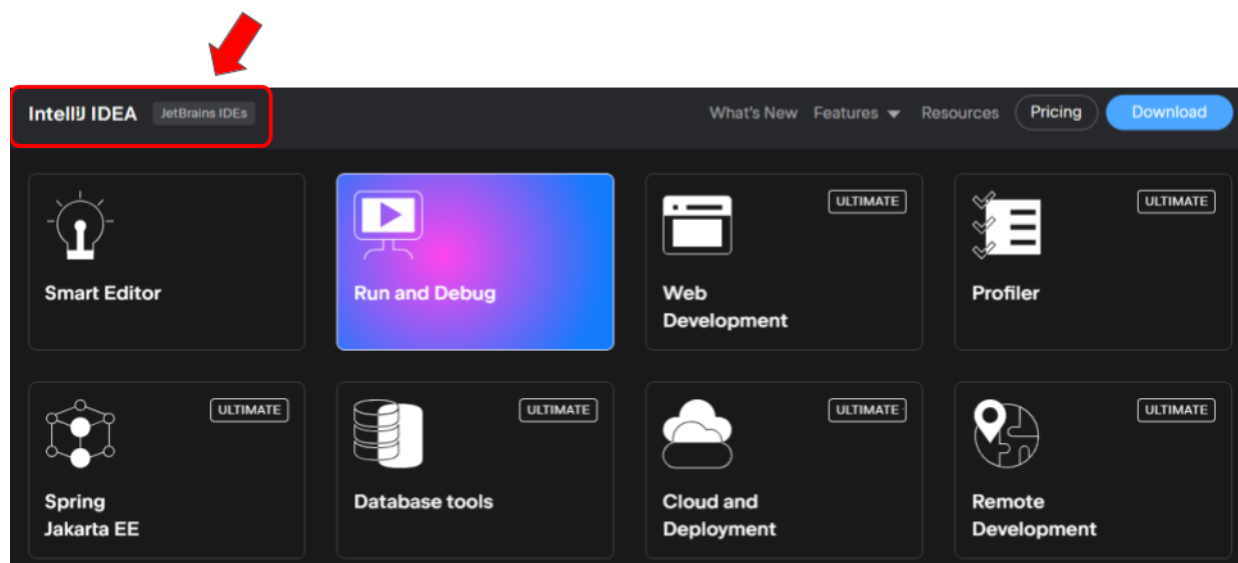
## Tarea 2

### Entorno de Desarrollo para producir la app.

Para la realización de este proyecto se ha seleccionado el IDE de IntelliJ IDEA.

IntelliJ IDEA esta preparado para el desarrollo web, ya que admite JavaScript y tecnologías relacionadas, como TypeScript, React, Vue, Angular, Node.js, HTML y hojas de estilo. Para todos estos lenguajes, IntelliJ IDEA proporciona funciones esenciales como finalización de código, acciones contextuales, resaltado de errores y más.

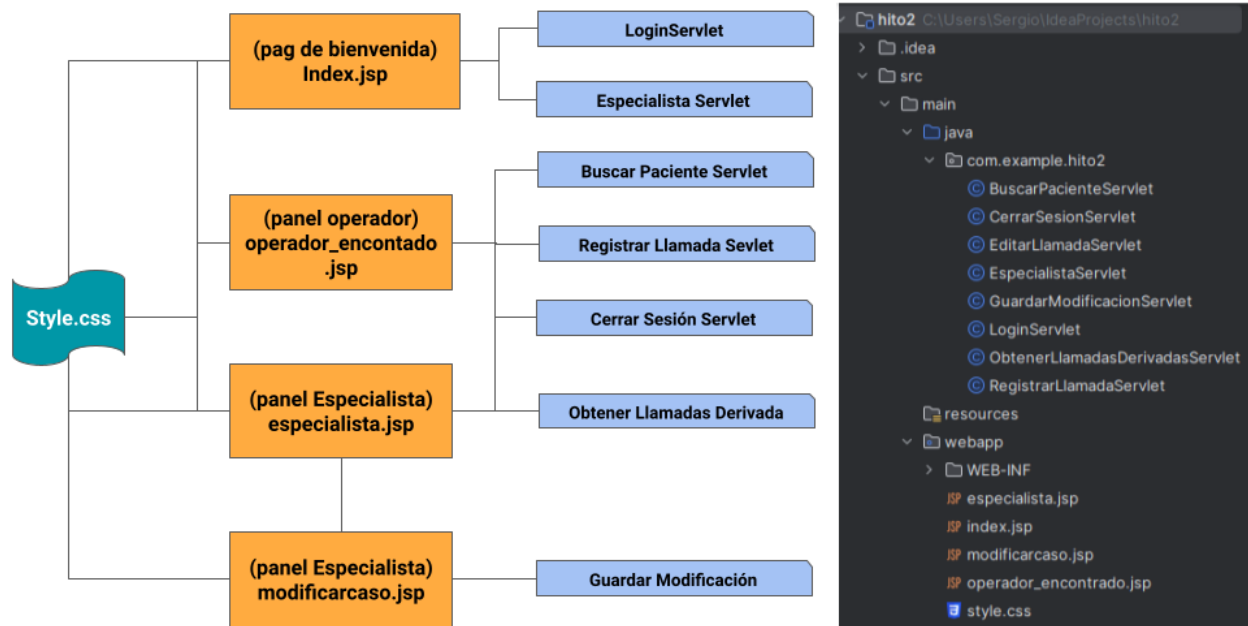
Es una latima no poder ver una pequeña introducción en la escuela de las tecnologias que se mencionan (TypeScript, React, Vue, Angular). De todas formas y aunque no estoy familiarizado con este IDE, he tratado de introducir además de HTML y CSS3, JavaScript.



## Estructura del proyecto.

En la siguiente imagen se muestra la estructura del proyecto con las páginas que he creado y los servlets asociados a cada una de ellas. En los siguientes apartados ire explicando de forma breve la funcionalidad de cada una de ellas.

### Estructura del proyecto



### Oportunidades de Mejora

Aunque el enunciado no lo solicitaba, esta aplicación carece de un **panel de administrador** para poder dar de alta a los empleados del servicio de atención al paciente (los que operan el sistema). Ahora si se desea dar de alta un nuevo operador debería darse de alta directamente en la BBDD lo cual requiere de personal especializado y con los conocimientos necesarios. Sería conveniente introducirlo en futuras versiones para poder delegar esa funcion a una persona sin conocimientos.

Por otro lado, creo que para este tipo de servicios, no solo es importante registrar las llamadas, tambien es importante conocer que ha ocurrido en un determinado periodo de tiempo, sería conveniente añadir un **panel de control con las estadísticas del servicio** ( cuantas llamadas se han atendido, cuantas se han cerrado, que tipo de problemas, a quien se han derivado, quien ha resuelto más problemas,etc). Esto permitiría tomar decisiones a la hora de dimensionar el personal por ejemplo.

## Diseño y Estilos

CSS permite separar la estructura HTML (contenido) del diseño y estilo (presentación). Esto facilita la modificación y actualización del diseño sin afectar el contenido, y viceversa. En mi proyecto he tratado de realizar un diseño sencillo con la información necesaria y que visualmente sea correcto.

No he usado bootstrap, pero para la maquetación he tratado de introducir flex.

```
body {
  font-family: "Roboto Light", sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f4f4f4;
}

header {
  background-color: #333;
  color: white;
  text-align: center;
  padding: 10px;
}

footer {
  background-color: #2bbe75;
  color: white;
  text-align: center;
  padding: 10px;
  position: fixed;
  bottom: 0;
  width: 100%;
}

section {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 80vh;
}

.login-form {
  width: 300px;
  padding: 20px;
  background-color: #fff;
  border-radius: 8px;
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);
  margin: 10px;
  transition: box-shadow 0.3s ease-in-out;
}
```

### Style.css

Al usar hojas de estilo, los navegadores pueden almacenar en caché el archivo CSS, lo que mejora la velocidad de carga de las páginas y reduce el consumo de ancho de banda.

La separación de estilos del código HTML facilita la identificación y corrección de errores, ya que los cambios en la apariencia no afectan la estructura del contenido.

Página de Bienvenida al Servicio.

The screenshot shows a web browser window displaying the login page of a medical platform. The browser's address bar shows the URL `localhost:8080/hito2_war_exploded/index.jsp`. The page has a dark header with the title "Plataforma Médica". Below the header, the text "Bienvenido a la Plataforma" and "Sistema de Atención Médica Telefónica." is displayed. There are two login forms: one for "Operador" and one for "Especialista". Each form has fields for "Su Nombre:" and "Su ID:", and a "Iniciar Sesión" button. The footer is green and contains the text "© 2023 Plataforma Médica".

**Operadores** son los que reciben la llamada en un primer momento y después la deriva si es necesario a un **especialista** una enfermera, un médico o un psicólogo.

La página de ofrece la posibilidad de loguearse a personal independientemente de su rol.

La funcionalidad para loguearse en el sistema esta desarrollada en un servlet que se encargar de buscar al usuario en la BBDD y comprobar que ha introducido la contraseña correctamente. Si es así le da acceso al panel correspondiente.

```
try {
    // Cargar el controlador JDBC
    Class.forName( className: "com.mysql.cj.jdbc.Driver");
} catch (ClassNotFoundException e) {
    e.printStackTrace(); // o manejar la excepción de acuerdo a tus necesidades
}

try (Connection connection = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/hito", user: "root", password: "curso");
    PreparedStatement statement = connection.prepareStatement( sql: "SELECT * FROM Operador WHERE ID_operador = ? AND nombre = ?")) {

    statement.setString( parameterIndex: 1, ID);
    statement.setString( parameterIndex: 2, nombre);

    // Ejecutar la consulta
    try (ResultSet resultSet = statement.executeQuery()) {
        // Verificar si se encontró el operador
        if (resultSet.next()) {
            // Operador encontrado
            // Realiza aquí las acciones necesarias si el operador existe
            String nombreOperador = resultSet.getString( columnLabel: "nombre");
            System.out.println("Operador encontrado: " + nombreOperador);
            // Obtiene o crea la sesión
            HttpSession session = request.getSession();

            // Guarda el nombre de usuario en la sesión
            session.setAttribute( s: "username", nombreOperador);
            session.setAttribute( s: "Id_operador", ID);

        } else {
            // Operador no encontrado
            mensajeError = "Usuario o contraseña incorrectos.";
        }
    }
} catch (SQLException e) {
    e.printStackTrace(); // Manejar las excepciones adecuadamente en un entorno de producción
    mensajeError = "Error en la base de datos.";
}
```

## Panel del Operador

### Plataforma Médica

Bienvenido, Juan Perez!

Cerrar Sesión

#### Operador

DNI del paciente:

Buscar paciente

#### Registro de Llamada

Nombre:  
Amelia

DNI:  
222222220

Teléfono:  
123432353

Motivo:  
Me duele la muñeca

Seleccionar Tipo de LLamada:  
Médica

Consejo del operador:  
Derivar la llamada a un especialista

Seleccionar:  
Derivar a Especialista

Registrar Llamada

© 2023 Plataforma Médica

Una vez iniciada la sesión el Operador podrá comenzar a recibir llamadas. El operador se encarga de solicitar el número del DNI del paciente y lo busca en el sistema.

Si lo encuentra, en el formulario de registro de llamada aparecerá relleno el campo DNI y nombre del paciente y podrá comenzar a rellenar el motivo de la llamada. Si no lo encuentra solamente dejara el campo DNI cumplimentado con el dato que introdujo anteriormente.



Si el paciente no existe en la BBDD antes de registrar la llamada registramos al paciente en la tabla pacientes y posteriormente se registra la llamada en la tabla correspondiente, dejando constancia de que operador fue el que registro la llamada.

```
try {
    // Cargar el controlador JDBC
    Class.forName( className: "com.mysql.cj.jdbc.Driver");
} catch (ClassNotFoundException e) {
    e.printStackTrace(); // o manejar la excepción de acuerdo a tus necesidades
}

try (Connection connection = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/hito", user: "root", password: "curso")){
    if (!pacienteok) {
        System.out.println ("antes del insertar el paciente" + DNI + nombrePaciente);
        // La condición es falsa, por lo tanto, se ejecuta el bloque de código dentro del if
        String insertPacienteQuery = "INSERT INTO Paciente (DNI, Nombre) VALUES (?, ?)";
        try (PreparedStatement pacienteStatement = connection.prepareStatement(insertPacienteQuery)) {
            // Configurar los valores para la inserción
            pacienteStatement.setString( parameterIndex: 1, DNI);
            pacienteStatement.setString( parameterIndex: 2, nombrePaciente);
            // Ejecutar la inserción
            pacienteStatement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace(); // Manejar la excepción según tus necesidades
            mensajeError = "Error al insertar en la tabla Paciente.";
        }
    }

    // Insertar datos de la llamada en la base de datos
    String insertCallQuery = "INSERT INTO llamada (numero_del_llamante, Tipo_de_llamada, Fecha_hora, ID_operador, DNI_paciente, consejo_operado, motivo, observacion_llamada_molesta, estado) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
    try (PreparedStatement statement = connection.prepareStatement(insertCallQuery)) {
        statement.setString( parameterIndex: 1, telefono); // numero_del_llamante
        statement.setString( parameterIndex: 2, tipollamada); // Tipo_de_llamada
        statement.setString( parameterIndex: 3, fechaHoraFormateada);
        statement.setString( parameterIndex: 4, idOpe);
        statement.setString( parameterIndex: 5, DNI); // DNI_paciente
        statement.setString( parameterIndex: 6, consejo); // consejo_operado
        statement.setString( parameterIndex: 7, motivo); // Motivo
        statement.setBoolean( parameterIndex: 8, observacion_llamada_molesta); // observacion_llamada_molesta
        statement.setString( parameterIndex: 9, estado); // Estado
        statement.executeUpdate();
    }
}
```

El operador, si así lo cree conveniente podrá dar por cerrado el caso o derivarlo a un especialista.

Especialistas

Plataforma Médica

Bienvenido a la Plataforma

Sistema de Atención Médica Telefónica.

Operador

Su Nombre:

Su ID:

Iniciar Sesión

Especialista

Su Nombre:

Su ID:

Iniciar Sesión

© 2023 Plataforma Médica

En el caso de que sea un especialista quien inicie sesion, este podrá visualizar todas las llamadas que han sido derivadas por los operadores, independientemente de su rol. Solo muestras las llamadas derivadas no las cerradas por el operador.

Plataforma Médica - Especialista

Bienvenido, David Herrera!

Cerrar Sesión

Llamadas

ID Llamada	Numero del llamante	Tipo de llamada	Fecha y Hora	Motivo	Consejo Operador	Consejo Especialista	Observación Llamada Molesta	Acciones
1	555-7777	Enfermera	2023-12-02 14:45:00	Corte en la raja el culo	Aplicar desinfectante en la herida	Agendar cita con la enfermera para revisión	false	<a href="#">Editar</a>
2	555-6666	Médica	2023-12-03 16:30:00	Dolor de cabeza	Tomar medicamento según prescripción	Reposo y tomar agua	false	<a href="#">Editar</a>
3	555-5555	Psicológica	2023-12-04 10:15:00	Estrés laboral	Escuchar música relajante	Programar una cita para sesión psicológica	false	<a href="#">Editar</a>
5	555-3333	Médica	2023-12-06 11:45:00	Lesión en la pierna	Descansar y aplicar hielo	Visitar a la enfermera para revisión	false	<a href="#">Editar</a>
6	555-2222	Psicológica	2023-12-07 14:30:00	Estrés laboral	Practicar técnicas de relajación	Agendar sesión con el psicólogo	false	<a href="#">Editar</a>
7	555-1111	Enfermera	2023-12-08 09:15:00	Quemadura leve	Aplicar vendaje en la herida	Agendar revisión con la enfermera	false	<a href="#">Editar</a>
8	555-0000	Médica	2023-12-09 15:45:00	Torcedura en el tobillo	Descansar y mantener la pierna elevada	Agendar cita con el médico	false	<a href="#">Editar</a>
9	555-9999	Enfermera	2023-12-10 12:30:00	Picadura de insecto	Aplicar crema en la picadura	Agendar revisión con la enfermera	false	<a href="#">Editar</a>
10	555-8888	Molestia	2023-12-11 07:00:00	Llamada de broma	null	null	true	<a href="#">Editar</a>
19	5554534546	medica	2023-12-08 10:29:15	no sale	derivarlo a un medico	null	false	<a href="#">Editar</a>
21	123432353	medica	2023-12-08 13:21:30	Me duele la muñeca	Derivar la llamada a un especialista	null	false	<a href="#">Editar</a>

© 2023 Plataforma Médica

.El especialista podrá seleccionar la llamada que desee resolver y modificar los datos con la resolución.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // logica para obtener las llamadas desde la base de datos
    List<Llamada> llamadas = obtenerLlamadasDesdeBaseDeDatos();

    // Construir el contenido de la tabla en HTML
    String contenidoTabla = construirContenidoTabla(llamadas);

    // Enviar la respuesta como HTML
    response.setContentType("text/html");
    response.getWriter().write(contenidoTabla);
}

// logica para obtener las llamadas desde la base de datos utilizando JDBC
private List<Llamada> obtenerLlamadasDesdeBaseDeDatos() {
    List<Llamada> llamadas = new ArrayList<>();

    try (Connection connection = DriverManager.getConnection(url, user, password);
        PreparedStatement statement = connection.prepareStatement(sql)) {
        try (ResultSet resultSet = statement.executeQuery()) {
            while (resultSet.next()) {
                Llamada llamada = new Llamada();
                llamada.setIdLlamada(resultSet.getInt(columnName.getIdLlamada()));
                llamada.setNumeroLlamante(resultSet.getString(columnName.getNumeroLlamante()));
                llamada.setTipoLlamada(resultSet.getString(columnName.getTipoDeLlamada()));
                llamada.setFechaHora(resultSet.getString(columnName.getFechaHora()));
                llamada.setMotivo(resultSet.getString(columnName.getMotivo()));
                llamada.setConsejoOperador(resultSet.getString(columnName.getConsejoOperador()));
                llamada.setConsejoEspecialista(resultSet.getString(columnName.getConsejoEspecialista()));
                llamada.setObservacionMolestia(resultSet.getBoolean(columnName.getObservacionMolestia()));
                // Seiran mas atributos segun la modelo de datos
                llamadas.add(llamada);
            }
        }
    } catch (SQLException e) {
        e.printStackTrace(); // Muestra las excepciones de conexión a la base de datos
    }
}
```

```
private String construirContenidoTabla(List<Llamada> llamadas) {
    StringBuilder contenidoTabla = new StringBuilder();

    // Cabecera de la tabla
    contenidoTabla.append("<tr>" +
        "<th>ID Llamada</th>" +
        "<th>Numero del Llamante</th>" +
        "<th>Tipo de Llamada</th>" +
        "<th>Fecha y Hora</th>" +
        "<th>Motivo</th>" +
        "<th>Consejo Operador</th>" +
        "<th>Consejo Especialista</th>" +
        "<th>Observación Llamada Molestia</th>" +
        "</tr>");

    // Contenido de la tabla
    for (Llamada llamada : llamadas) {
        contenidoTabla.append("<tr>" +
            "<td>" + llamada.getIdLlamada() + "</td>" +
            "<td>" + llamada.getNumeroLlamante() + "</td>" +
            "<td>" + llamada.getTipoLlamada() + "</td>" +
            "<td>" + llamada.getFechaHora() + "</td>" +
            "<td>" + llamada.getMotivo() + "</td>" +
            "<td>" + llamada.getConsejoOperador() + "</td>" +
            "<td>" + llamada.getConsejoEspecialista() + "</td>" +
            "<td>" + llamada.isObservacionMolestia() + "</td>" +
            "<td>" + "<a href='\"" + construirLlamadaServletId() + llamada.getIdLlamada() + "\">Editar</a>" + "</td>" +
            "</tr>");
    }

    return contenidoTabla.toString();
}
```

## Oportunidades de Mejora

Aunque el enunciado no lo solicitaba, esta aplicación carece de un **filtro por rol** en el panel de especialista. Es muy posible que a un psicologo no le interese ver todas las llamadas derivadas si no solamente las que afecten a su rol.

## Modificación del caso

Una vez que un especialista seleccione una llamada de la lista para modificar los datos, aparece un formulario con los datos recogidos en los pasos anteriores de la llamada atendida. En el formulario podrá realizar las modificaciones y comentarios que crea oportuno.

### Introduzca las modificaciones del Caso

ID Llamada:  
21

Número del Llamante:  
123432353

Tipo de Llamada:  
medica

Fecha y Hora:  
2023-12-08 13:21:30

Motivo:  
Me duele la muñeca

Consejo Operador:  
Derivar la llamada a un especialista

Consejo Especialista:  
toma un paracetamol

Observación Llamada Molesta:  
☐

Seleccionar:  
Cerrar\_Caso

Guardar Cambios

© 2023 Plataforma Médica

Además, podrá dar por cerrada la llamada o derivar la llamada a otro especialista en cuyo caso volverá a verse reflejada en el listado de llamadas derivadas hasta que esta se haya dado por cerrada.

Todo quedará registrado en la BBDD.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // Obtener los parámetros del formulario
    int idLlamada = Integer.parseInt(request.getParameter(s: "idLlamada"));
    String numerollamante = request.getParameter(s: "numerollamante");
    String tipoLlamada = request.getParameter(s: "tipoLlamada");
    String fechaHora = request.getParameter(s: "fechaHora");
    String motivo = request.getParameter(s: "motivo");
    String consejoOperador = request.getParameter(s: "consejoOperador");
    String consejoEspecialista = request.getParameter(s: "consejoEspecialista");
    boolean observacionMolesta = request.getParameter(s: "observacionMolesta") != null;
    String estado = request.getParameter(s: "estado");
    // Lógica para actualizar los datos en la base de datos
    actualizarLlamadaEnBaseDeDatos(idLlamada, numerollamante, tipoLlamada, fechaHora, motivo, consejoOperador, consejoEspecialista, observacionMolesta, estado);

    // Redirigir a la página de éxito o donde desees
    response.sendRedirect(s: "especialista.jsp");
}

1 usage
private void actualizarLlamadaEnBaseDeDatos(int idLlamada, String numerollamante, String tipoLlamada,
    String fechaHora, String motivo, String consejoOperador,
    String consejoEspecialista, boolean observacionMolesta, String estado) {
    // Lógica para actualizar los datos de la llamada en la base de datos utilizando JDBC
    try (Connection connection = DriverManager.getConnection(url: "jdbc:mysql://localhost:3306/nito", user: "root", password: "curso")) {
        String sql = "UPDATE Llamada SET numero_del_llamante=?, Tipo_de_llamada=?, Fecha_hora=?, motivo=?, " +
            "consejo_operador=?, consejo_especialista=?, observacion_llamada_molesta=?, estado=? WHERE ID_llamada=?";
        System.out.println(sql);
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(parameterIndex: 1, numerollamante);
            statement.setString(parameterIndex: 2, tipoLlamada);
            statement.setString(parameterIndex: 3, fechaHora);
            statement.setString(parameterIndex: 4, motivo);
            statement.setString(parameterIndex: 5, consejoOperador);
            statement.setString(parameterIndex: 6, consejoEspecialista);
            statement.setBoolean(parameterIndex: 7, observacionMolesta);
            statement.setString(parameterIndex: 8, estado);
            statement.setInt(parameterIndex: 9, idLlamada);

            // Ejecutan la actualización
            statement.executeUpdate();
        }
    }
}
```