

Presenta : Jersson Sebastian Fuentes, Juan Pablo Espinel , Guadalupe Garcia y  
Maria Jose Duarte

## Plataforma Web de Apoyo en Salud Mental con Chat Box Inteligente

### 1. Introducción

La salud mental es un aspecto fundamental en la vida de los estudiantes, pero muchas veces no cuentan con un espacio de apoyo inmediato. Este proyecto busca desarrollar una plataforma web en Java que permita a los usuarios registrar sus emociones, acceder a recursos de bienestar y contar con un chat box de ayuda basado en inteligencia artificial ligera, que brinde respuestas motivacionales y consejos básicos de autocuidado.

### 2. Objetivo General

Desarrollar una aplicación web en Java que sirva como apoyo para la salud mental, permitiendo a los usuarios llevar un diario emocional y acceder a un chat box inteligente que ofrezca acompañamiento inmediato mediante respuestas automáticas.

### 3. Objetivos Específicos

- Implementar un sistema de registro y autenticación de usuarios.
- Desarrollar un diario emocional donde los usuarios registren cómo se sienten cada día.
- Incorporar un chat box de ayuda con respuestas automáticas basadas en palabras clave relacionadas con estados emocionales.
- Diseñar un módulo de administración para que un psicólogo o administrador pueda alimentar la base de datos de respuestas y recursos.
- Integrar reportes gráficos de evolución emocional para cada usuario.

### 4. Alcance del Proyecto

- Usuarios finales: Estudiantes o personas interesadas en el autocuidado de la salud mental.
- Funciones principales:
  - Registro/Login.
  - Diario emocional.
  - Chat box inteligente.
  - Recursos de bienestar (ejercicios de respiración, tips).
  - Panel administrador.
- No incluirá: Diagnóstico clínico ni sustitución de atención psicológica profesional.

### 5. Justificación

La presión académica y social puede generar estrés, ansiedad y otros problemas emocionales en los estudiantes. Este proyecto ofrece una solución digital accesible

que fomente el autoconocimiento emocional y brinde apoyo inmediato a través de un chat inteligente. Además, servirá como práctica para aplicar conceptos de programación web en Java, Servlets, JSP, MVC y bases de datos relacionales.

---

## 6. Metodología de Desarrollo

Se utilizará una metodología iterativa e incremental, dividiendo el trabajo en fases:

1. Diseño de la base de datos y modelo MVC.
  2. Implementación de registro/login.
  3. Desarrollo del diario emocional.
  4. Creación del chat box con respuestas automáticas.
  5. Integración de gráficos y reportes.
  6. Pruebas y ajustes finales.
- 

## 7. Tecnologías a Utilizar

- Backend: Java (Servlets, JSP, JDBC, MVC).
  - Frontend: HTML, CSS, Bootstrap, JavaScript (AJAX), Angular, Canvas.
  - Base de datos: PostgreSQL o MySQL.
  - Extras: Chart.js para gráficos, JSTL para vistas dinámicas.
- 

**Nota:** Esta sujeta a cambios toda la parte de tecnologías

---

## 8. Modelo de Datos

```
-- Tabla de usuarios CREATE TABLE usuarios ( id INT AUTO_INCREMENT  
PRIMARY KEY, nombre VARCHAR(100) NOT NULL, correo VARCHAR(100) NOT  
NULL UNIQUE, contraseña_hash VARCHAR(255) NOT NULL, rol VARCHAR(50)  
DEFAULT 'usuario' );
```

```
-- Tabla para el diario emocional CREATE TABLE entradas_diario ( id INT  
AUTO_INCREMENT PRIMARY KEY, usuario_id INT NOT NULL, fecha  
TIMESTAMP DEFAULT CURRENT_TIMESTAMP, emocion VARCHAR(50) NOT  
NULL, descripcion TEXT, FOREIGN KEY (usuario_id) REFERENCES usuarios(id) );  
-- Tabla para los mensajes del chat CREATE TABLE mensajes ( id INT  
AUTO_INCREMENT PRIMARY KEY, usuario_id INT NOT NULL, remitente  
VARCHAR(50) NOT NULL, -- 'usuario', 'bot' o 'admin' mensaje TEXT NOT NULL,  
fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP, FOREIGN KEY  
(usuario_id) REFERENCES usuarios(id) );
```

```
-- Tabla para las respuestas automáticas del bot CREATE TABLE  
respuestas_automaticas ( id INT AUTO_INCREMENT PRIMARY KEY, palabra_clave  
VARCHAR(100) NOT NULL UNIQUE, respuesta TEXT NOT NULL ); -- Tabla para  
los recursos de bienestar CREATE TABLE recursos_bienestar ( id INT  
AUTO_INCREMENT PRIMARY KEY, titulo VARCHAR(255) NOT NULL, descripcion  
TEXT NOT NULL, tipo VARCHAR(50) NOT NULL, -- 'ejercicio', 'articulo', etc. url  
VARCHAR(255) );
```

```
-- Tabla para registrar los recursos que un usuario ha visto/completado CREATE  
TABLE usuario_recurso ( id INT AUTO_INCREMENT PRIMARY KEY, usuario_id
```

*INT NOT NULL, recurso\_id INT NOT NULL, fecha\_completado TIMESTAMP  
DEFAULT CURRENT\_TIMESTAMP, FOREIGN KEY (usuario\_id) REFERENCES  
usuarios(id), FOREIGN KEY (recurso\_id) REFERENCES recursos\_bienestar(id) );*

---

## 9. Impacto esperado

- Académico: Demostrar habilidades en desarrollo web con Java.
- Social: Proporcionar un espacio digital de apoyo emocional accesible a estudiantes.
- Tecnológico: Integrar un chat inteligente básico con aprendizaje incremental (alimentado por el administrador).

---

## 10. Cronograma (ejemplo 6 semanas)

- Semana 1: Diseño BD y arquitectura MVC.
- Semana 2: Registro/Login.
- Semana 3: Diario emocional.
- Semana 4: Chat box inteligente.
- Semana 5: Panel admin + reportes gráficos.
- Semana 6: Pruebas, documentación y despliegue.

---

## 11. Resumen

Aplicación web en Java (Servlets + JSP) para apoyo en salud mental que incluye: diario emocional, test de autoevaluación, recursos de bienestar y un chat box con IA ligera (respuestas por palabras clave)

---

## 12. Actores

- Usuario: estudiante o persona que utiliza la plataforma para registrar emociones y usar el chat.
- Administrador :Gestiona respuestas automáticas, revisa historiales y responde chats.
- Sistema (Bot): agente automático que responde a mensajes según palabras clave.

---

## 13. Requerimientos Funcionales (RF)

1. RF01	2. Registro y autenticación: El sistema permitirá registrar usuarios y autenticar sesiones.
3. RF02	4. Perfil de usuario: El usuario podrá ver y editar su perfil.
5. RF03	6. Diario emocional: El usuario podrá crear, editar y eliminar entradas diarias (fecha, emoción, descripción).

7. RF04	8. Test de autoevaluación: El usuario podrá completar un test y guardar resultados.
9. RF05	10. Recursos de bienestar: Mostrar ejercicios y artículos, marcarlos como "completado".
11. RF06	12. Chat box: Enviar mensajes al chat y recibir respuestas del Bot
13. RF07	14. Respuestas automáticas: El Bot devolverá respuestas basadas en palabras clave almacenadas en la BD.
15. RF08	16. Historial de mensajes: Guardar y consultar el historial de mensajes por usuario.
17. RF09	18. Panel Administrador: CRUD de respuestas automáticas, gestión de ejercicios y revisión de usuarios.
19. RF10	20. Reportes gráficos: Mostrar evolución del estado de ánimo por periodos.

<b>14. Requerimientos No Funcionales (RNF)</b>
<ul style="list-style-type: none"> <li>• RNF01 - Privacidad: Los datos personales y mensajes deben almacenarse con acceso restringido.</li> </ul>
<ul style="list-style-type: none"> <li>• RNF2 - Disponibilidad: La app debe ser accesible desde navegadores modernos.</li> </ul>
<ul style="list-style-type: none"> <li>• RNF03 - Rendimiento: Respuesta del chat en &lt; 2s para consultas simples.</li> </ul>
<ul style="list-style-type: none"> <li>• RNF04 - Escalabilidad: Diseño modular para agregar NLP o API externas en el futuro.</li> </ul>

## 15. Casos de Uso

CU-01: Registrarse / Iniciar sesión

Actores: Usuario

Descripción: El usuario crea una cuenta o inicia sesión; el sistema valida credenciales y crea sesión.

Precondición: No autenticado.

Flujo principal:

1. Usuario ingresa correo y contraseña (o registra datos).
2. Sistema valida y crea sesión.

3. Usuario es redirigido al panel.  
Postcondición: Usuario autenticado.

#### CU-02: Registrar entrada en diario emocional

Actores: Usuario

Descripción: El usuario añade una entrada con fecha, emoción y descripción.

Flujo principal:

1. Usuario abre formulario "Nueva entrada".
2. Completa emoción y descripción.
3. Sistema guarda en `entradas_diario`.

#### CU-03: Usar Chat Box (Conversación)

Actores: Usuario, Bot, Administrador

Descripción: Usuario envía mensajes; el Bot intenta responder automáticamente;

Flujo principal:

1. Usuario escribe mensaje y presiona enviar.
2. Sistema guarda mensaje en `mensajes` (`remitente=usuario`).
3. Sistema consulta respuestas buscando palabra clave.
  - o Si se encuentra respuesta -> guardar respuesta del Bot y mostrarla.

#### CU-04: Administrar respuestas automáticas

Actores: Administrador

Descripción: El Admin crea/edita/elimina entradas en la tabla `respuestas` (`palabra_clave`, `respuesta`).

#### CU-05: Visualizar reportes de estado anímico

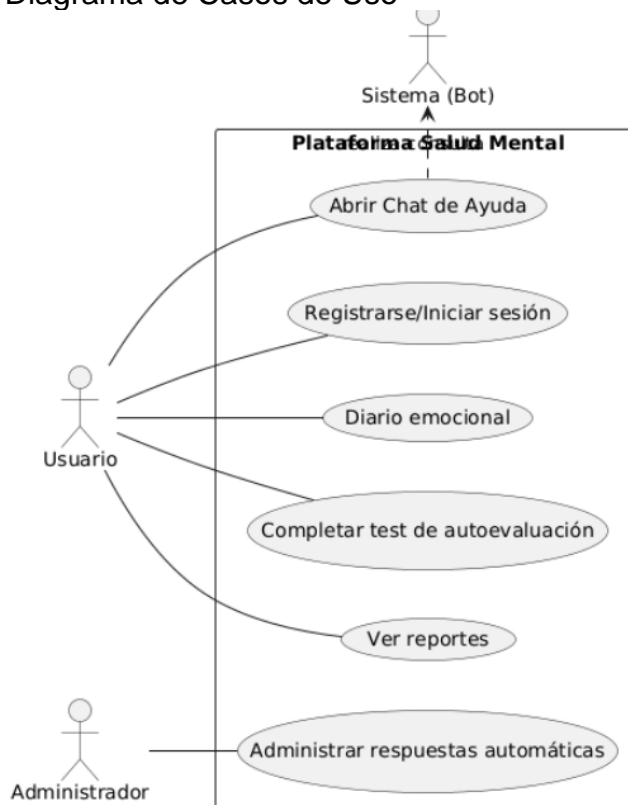
Actores: Usuario

Descripción: El usuario visualiza gráficos (por semana/mes) con la evolución de emociones.

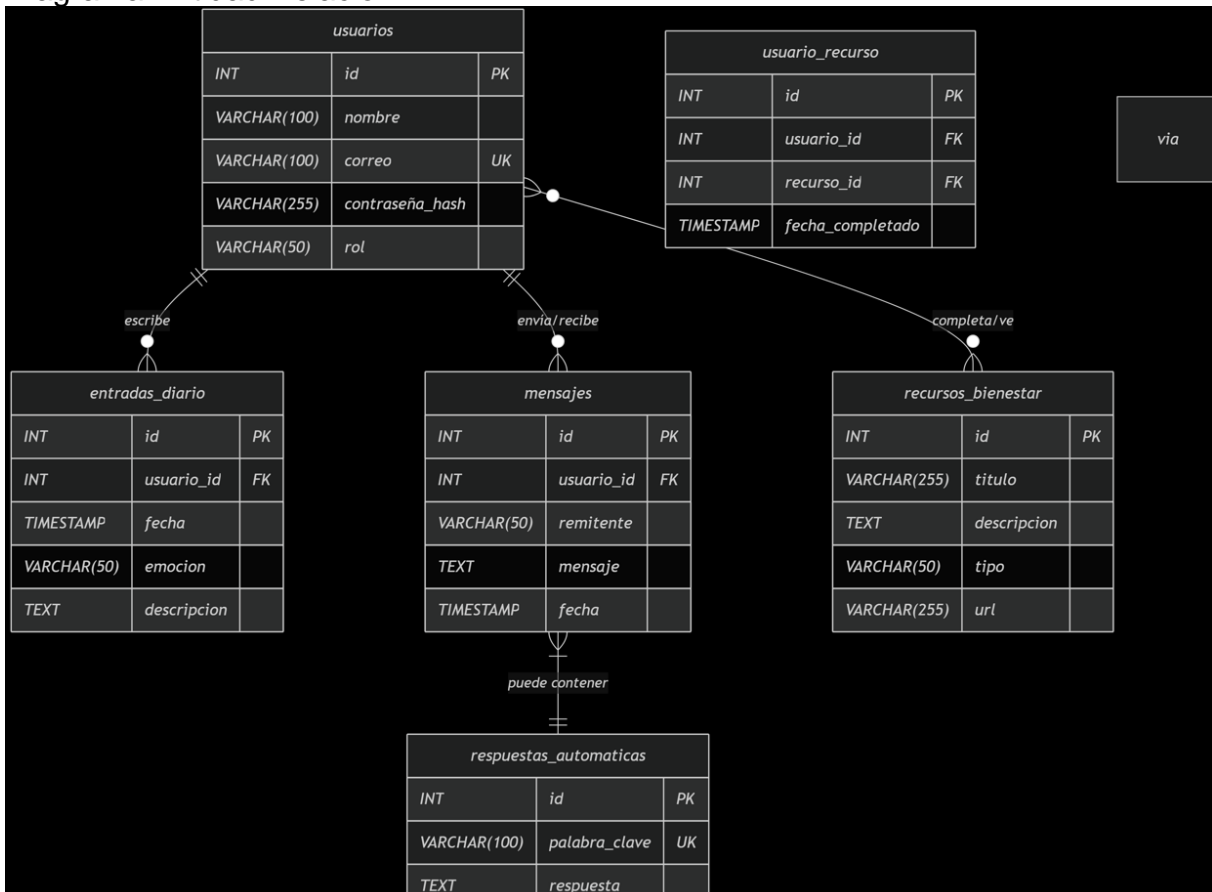
---

## 16. Diagramas

### Diagrama de Casos de Uso

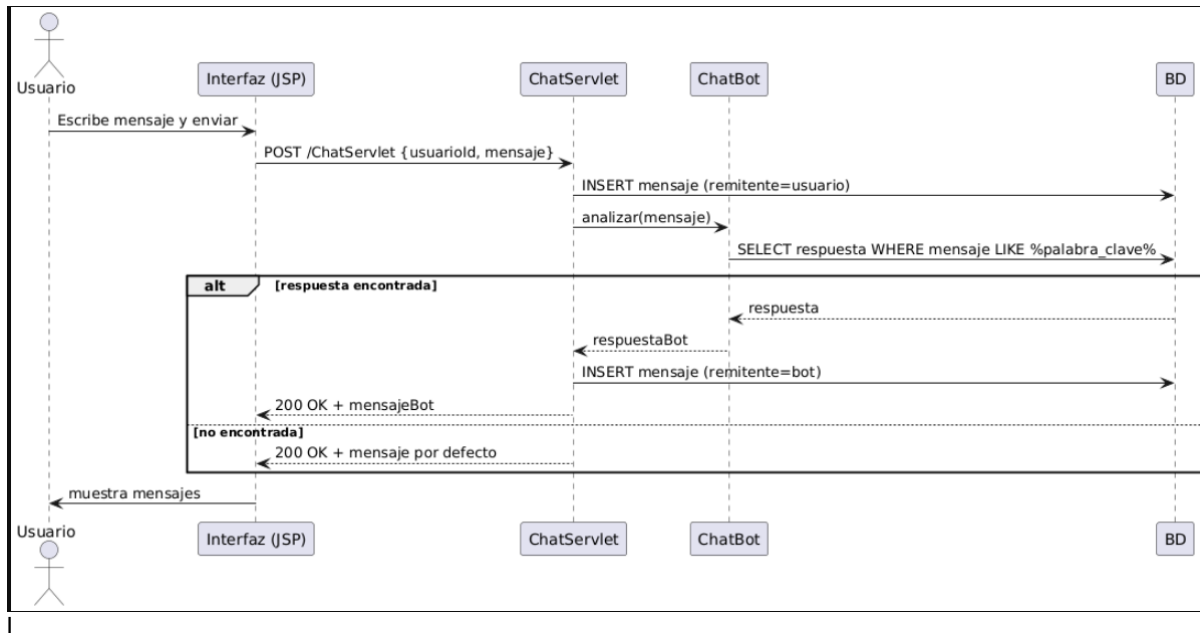


### Diagrama Entidad Relación



Nota: Base de datos aun sujeta a cambios a confirmar con el docente

### Diagrama de Secuencia (Envio de mensaje y respuesta automática)



## 17. Criterios de Aceptación / Casos de Prueba

1. Registro/Login: Usuario puede registrarse y luego iniciar sesión con credenciales válidas.
2. Diario: Crear una entrada y verla listada en el historial.
3. Chat - Bot: Enviar mensaje con palabra clave conocida y recibir respuesta correcta.
4. Chat - Historial: Conversación queda persistida y se recupera al recargar la página.
5. Administración: Admin puede crear una nueva palabra clave y respuesta y ésta funciona en el chat





