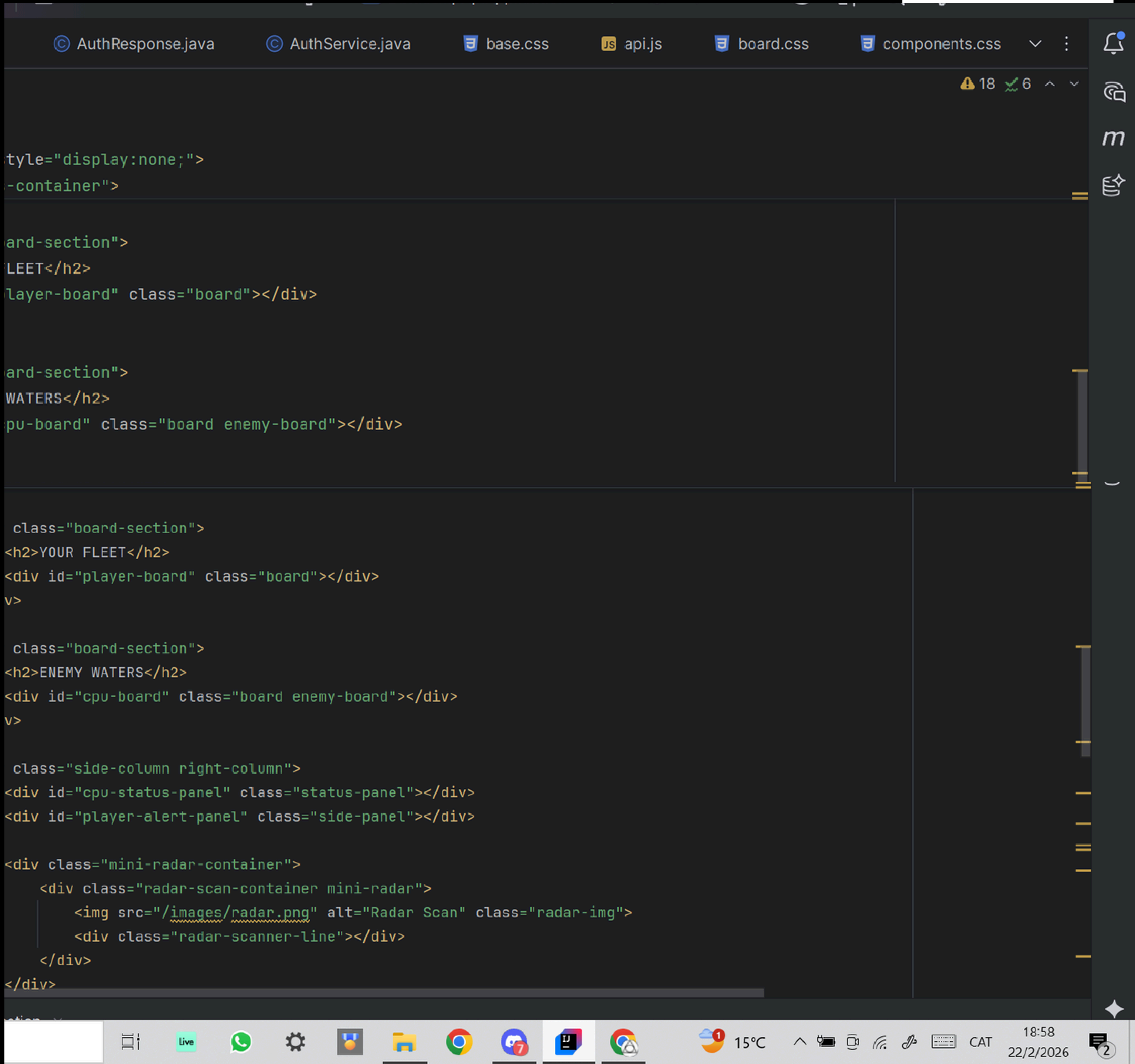
LA IA COMO HERRAMIENTA

- **IA Seleccionada:** Gemini PRO
- **Debugging guiado:** Resolución de errores (500, DOM) explicando el "porqué".
- **Enfoque didáctico:** Mentoría paso a paso.

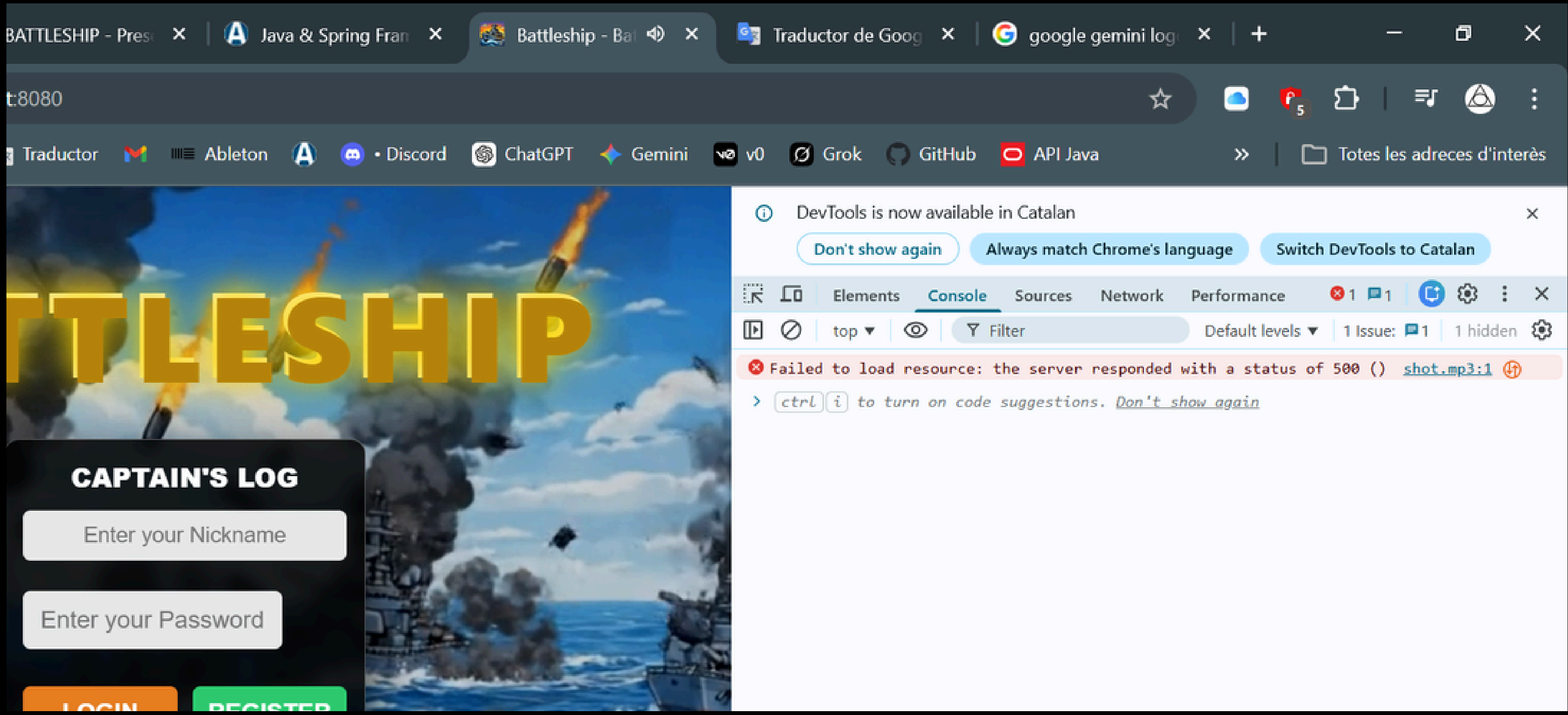


```
tyle="display:none;">
-container">

ard-section">
LEET</h2>
layer-board" class="board"></div>


ard-section">
WATERS</h2>
pu-board" class="board enemy-board"></div>
```

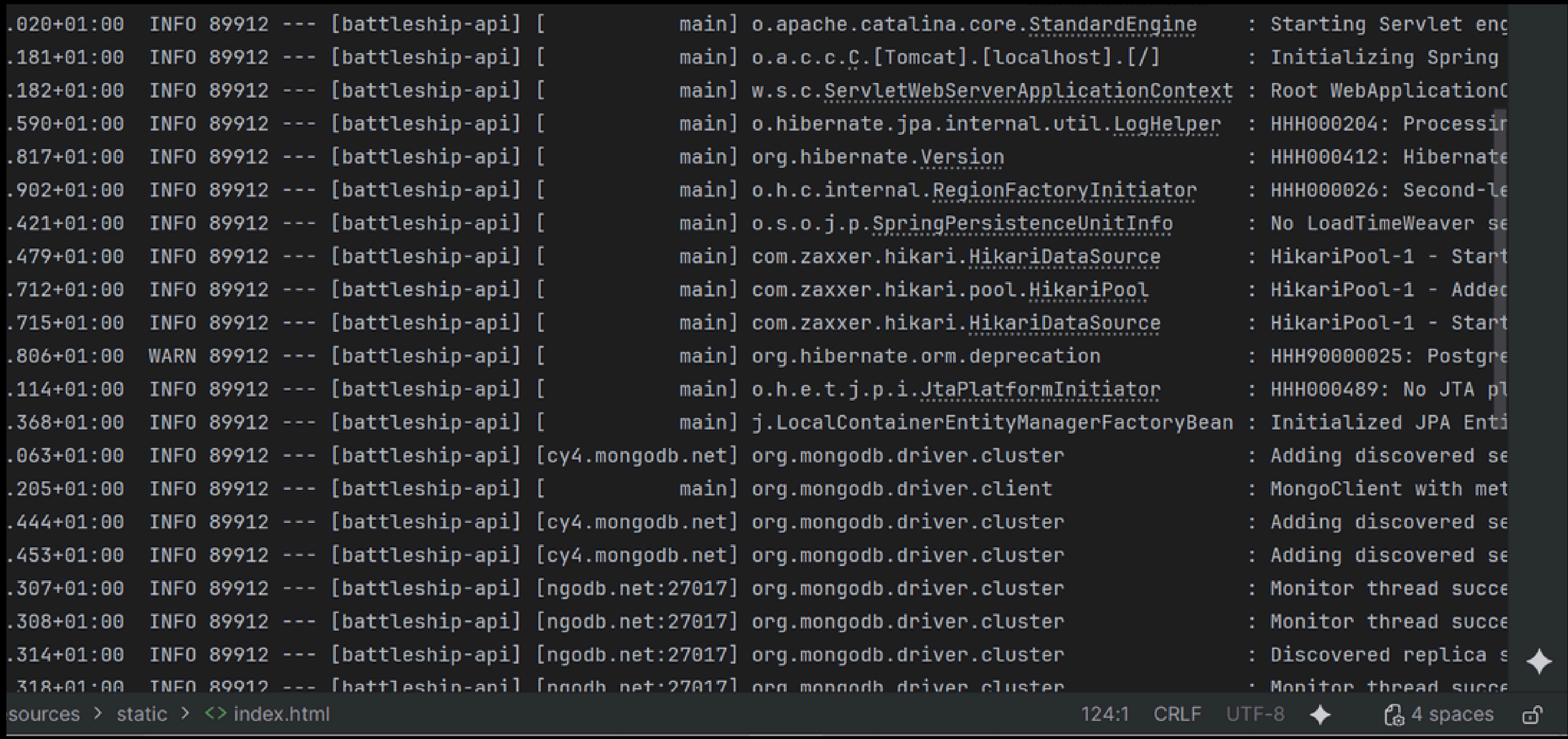
```
 class="board-section">
<h2>YOUR FLEET</h2>
<div id="player-board" class="board"></div>
v>

 class="board-section">
<h2>ENEMY WATERS</h2>
<div id="cpu-board" class="board enemy-board"></div>
v>

 class="side-column right-column">
<div id="cpu-status-panel" class="status-panel"></div>
<div id="player-alert-panel" class="side-panel"></div>

<div class="mini-radar-container">
    <div class="radar-scan-container mini-radar">
        <img src="/images/radar.png" alt="Radar Scan" class="radar-img">
        <div class="radar-scanner-line"></div>
    </div>
</div>
```

INTERACCIONES CON LA IA

- **Problema:** Error 500 al consultar el Ránquing después de añadir el sistema de Login.
- **Consulta a la IA:** Análisis de logs del servidor (Spring Boot).
- **Solución aprendida:** Comprensión del conflicto entre el código Java .



```
Failed to load resource: the server responded with a status of 500 ()   shot.mp3:1
> ctrl i to turn on code suggestions. Don't show again
```



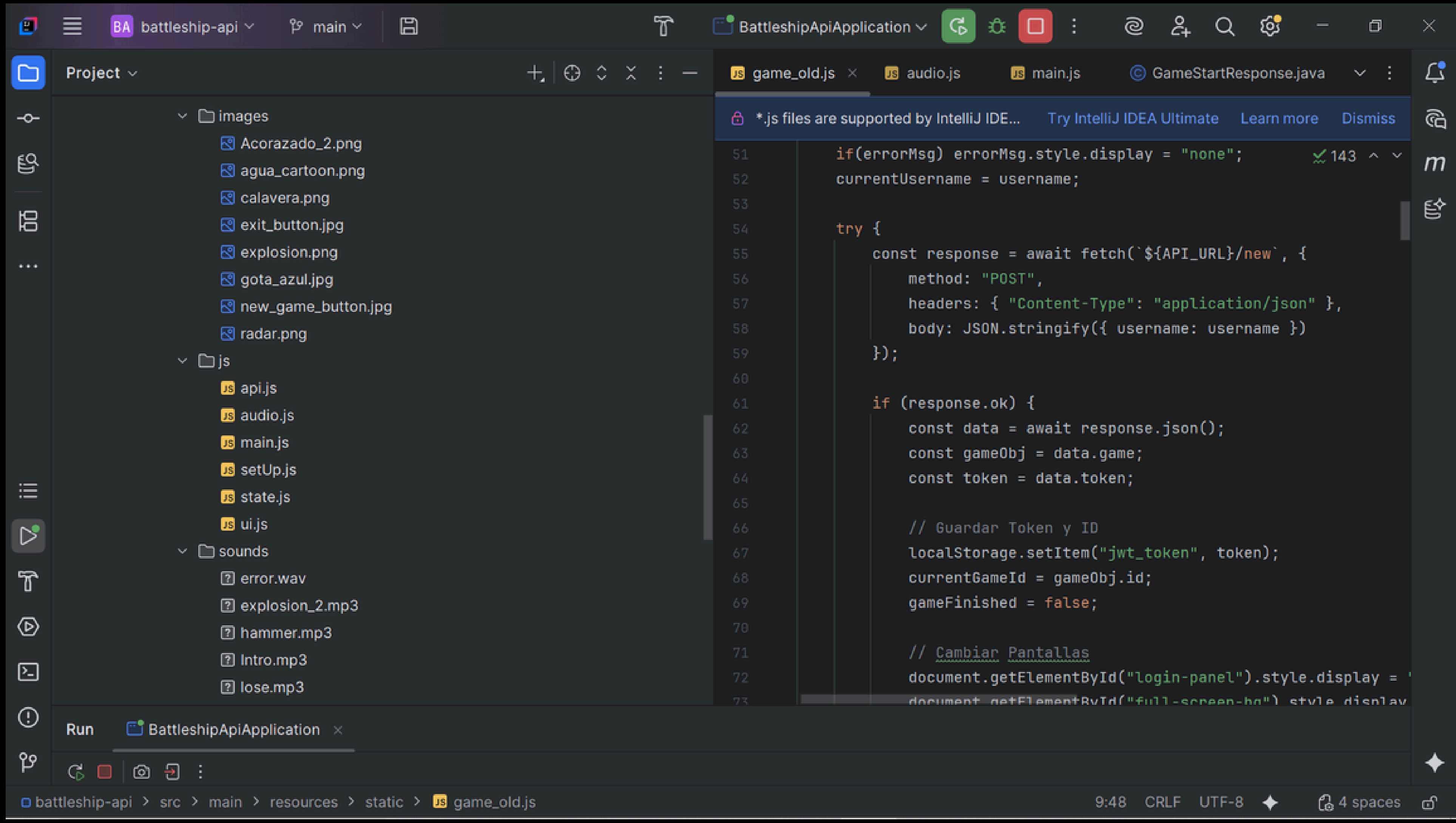
```
.020+01:00  INFO 89912 --- [battleship-api] [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet eng
.181+01:00  INFO 89912 --- [battleship-api] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring
.182+01:00  INFO 89912 --- [battleship-api] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplication(
.590+01:00  INFO 89912 --- [battleship-api] [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processir
.817+01:00  INFO 89912 --- [battleship-api] [           main] org.hibernate.Version                    : HHH000412: Hibernate
.902+01:00  INFO 89912 --- [battleship-api] [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Second-le
.421+01:00  INFO 89912 --- [battleship-api] [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver se
.479+01:00  INFO 89912 --- [battleship-api] [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start
.712+01:00  INFO 89912 --- [battleship-api] [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Addec
.715+01:00  INFO 89912 --- [battleship-api] [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start
.806+01:00  WARN 89912 --- [battleship-api] [           main] org.hibernate.orm.deprecation            : HHH90000025: Postgre
.114+01:00  INFO 89912 --- [battleship-api] [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA pl
.368+01:00  INFO 89912 --- [battleship-api] [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA Enti
.063+01:00  INFO 89912 --- [battleship-api] [cy4.mongodb.net] org.mongodb.driver.cluster               : Adding discovered se
.205+01:00  INFO 89912 --- [battleship-api] [           main] org.mongodb.driver.client                : MongoClient with met
.444+01:00  INFO 89912 --- [battleship-api] [cy4.mongodb.net] org.mongodb.driver.cluster               : Adding discovered se
.453+01:00  INFO 89912 --- [battleship-api] [cy4.mongodb.net] org.mongodb.driver.cluster               : Adding discovered se
.307+01:00  INFO 89912 --- [battleship-api] [ngodb.net:27017] org.mongodb.driver.cluster               : Monitor thread succe
.308+01:00  INFO 89912 --- [battleship-api] [ngodb.net:27017] org.mongodb.driver.cluster               : Monitor thread succe
.314+01:00  INFO 89912 --- [battleship-api] [ngodb.net:27017] org.mongodb.driver.cluster               : Discovered replica s
.318+01:00  INFO 89912 --- [battleship-api] [ngodb.net:27017] org.mongodb.driver.cluster               : Monitor thread succe
```

ANALISIS Y ADAPTACION DEL CODIGO

- **Características del código generado:** Código limpio, moderno (uso de async/await, Flexbox) y orientado a módulos.

- **Integración en la arquitectura:** Adaptación de código genérico a mis propios módulos personalizados (uiManager, audioManager, gameState).

- **DOM y HTML:** Corrección del anidamiento de etiquetas.



*.js files are supported by IntelliJ IDE...

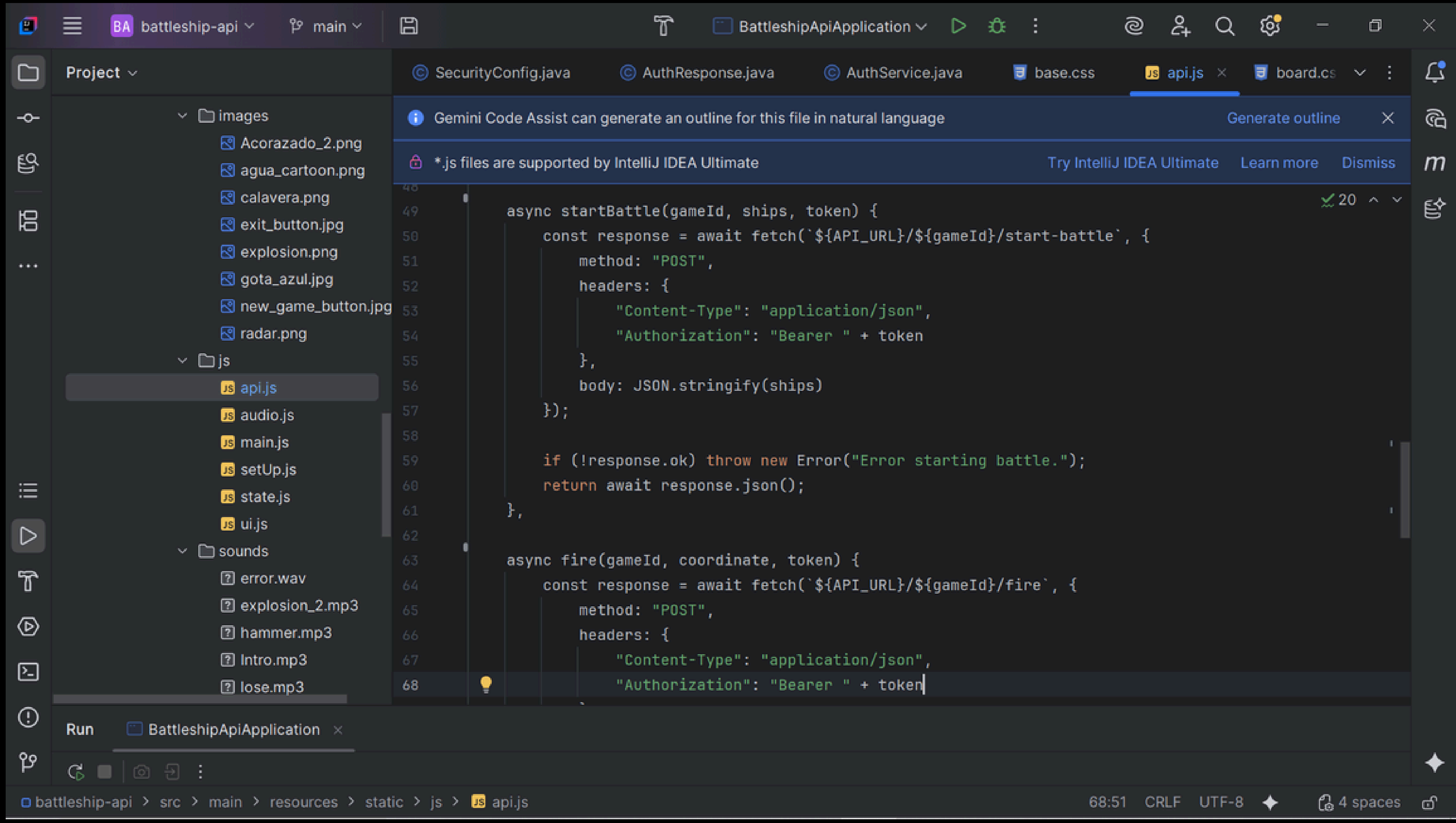
```
    if(errorMsg) errorMsg.style.display = "none";
    currentUsername = username;

    try {
        const response = await fetch(`${API_URL}/new`, {
            method: "POST",
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ username: username })
        });

        if (response.ok) {
            const data = await response.json();
            const gameObj = data.game;
            const token = data.token;

            // Guardar Token y ID
            localStorage.setItem("jwt_token", token);
            currentGameId = gameObj.id;
            gameFinished = false;

            // Cambiar Pantallas
            document.getElementById("login-panel").style.display = '
            document.getElementById("full-screen-bg").style.display
```

CONEXION FRONTEND - BACKEND

JavaScript (fetch) ➔ JSON ➔ Spring Boot (Java)

- **Arquitectura de Comunicación:** Consumo de la API REST de Spring Boot mediante la función nativa fetch de JavaScript, intercambiando datos en formato JSON.

- **Seguridad y Sesiones:** Implementación de JWT (JSON Web Tokens).



Gemini Code Assist can generate an outline for this file in natural language

*.js files are supported by IntelliJ IDEA Ultimate

```
    async startBattle(gameId, ships, token) {
        const response = await fetch(`${API_URL}/${gameId}/start-battle`, {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
                "Authorization": "Bearer " + token
            },
            body: JSON.stringify(ships)
        });

        if (!response.ok) throw new Error("Error starting battle.");
        return await response.json();
    },

    async fire(gameId, coordinate, token) {
        const response = await fetch(`${API_URL}/${gameId}/fire`, {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
                "Authorization": "Bearer " + token
```

REFLEXION SOBRE EL APRENDIZAJE

METODOLOGIA DEL CURSO Y LINEA DE APRENDIZAJE PERFECTA

Java

IT ACADEMY

MongoDB

JS

PostgreSQL