Zelda BOTW API & Frontend (Materials i Monstres)

Aquest projecte inclou una **API RESTful** i un **frontend interactiu** que simula una **GameBoy** per gestionar informació de **materials** i **monstres** del joc *The Legend of Zelda: Breath of the Wild.* També inclou funcionalitats avançades com la valoració amb estrelles, la selecció automàtica de categories i la gestió de vots.

Contingut

- Zelda BOTW API & Frontend (Materials i Monstres)
 - Contingut
 - M Com utilitzar la interfície (GameBoy)
 - A Característiques
 - Sackend (API)
 - Endpoints de l'API
 - Materials
 - Monstres
 - Vots
 - Frontend
 - Estructura del projecte
 - Funcionalitats del frontend
 - Walidacions
 - Validacions generals
 - Validacions específiques per a materials
 - Validacions específiques per a monstres
 - Comportament en cas d'errors
 - Millores
 - 1. Valoració amb estrelles
 - 2. Càlcul de la suma
 - 3. Selecció automàtica de categories
 - 4. Camps específics per a materials i monstres
 - W Instal·lació amb Docker
 - Requisits
 - Instruccions
 - 護題
 - Instal·lació en local
 - Requisits
 - Instruccions
 - Notes

Com utilitzar la interfície (GameBoy)

El frontend està dissenyat per simular una **GameBoy**, amb botons interactius que permeten navegar i gestionar els elements. A continuació es detallen els botons i les seves funcions:

2025-03-29 docu.md

Botó	Funció	
▲ / ▼	Navegar entre els elements (anterior/següent).	
√/ ►	Navegar entre els elements (anterior/següent).	
E	Editar l'element seleccionat.	
A	Afegir un nou element (material o monstre, segons la vista actual).	
D	Eliminar l'element seleccionat (mostra un modal de confirmació).	
Materials	Canviar a la vista de materials.	
Monsters	Canviar a la vista de monstres.	



Característiques

Característica	Descripció
Base de dades	MongoDB
CRUD complet	Per a materials i monstres
Frontend interactiu	React.js amb funcionalitats avançades
Valoració amb estrelles	Permet als usuaris valorar materials i monstres
Gestió de vots	Inclou càlcul de la suma
Selecció automàtica	La categoria es defineix automàticament segons el context
Camps específics	Mostra camps diferents segons si és un material o un monstre
Backend	Node.js + Express
Documentació interactiva	Swagger disponible a http://localhost:3001/api-docs



📚 Backend (API)

Endpoints de l'API

Materials

Mètode	Endpoint	Descripció	
GET	/materials	Obté tots els materials	
GET	/materials/:id	Obté un material específic	
POST	/materials	Crea un nou material	
PUT	/materials/:id	Actualitza un material existent	
DELETE	/materials/:id	Elimina un material	

Monstres

Mètode	Endpoint	Descripció	
GET	/monsters	Obté tots els monstres	
GET	/monsters/:id	Obté un monstre específic	
POST	/monsters	Crea un nou monstre	
PUT	/monsters/:id	Actualitza un monstre existent	
DELETE	/monsters/:id	Elimina un monstre	

Vots

Mètode	e Endpoint	Descripció	
GET	/votes	Obté el total de vots per element	
POST	/votes	Afegeix un vot a un element	

Frontend

Estructura del projecte

El frontend està desenvolupat amb **React.js** i té la següent estructura:

Funcionalitats del frontend

Funcionalitat	Descripció
Llistat d'elements	Mostra materials i monstres en targetes
Creació/edició d'elements	Permet afegir o editar materials i monstres mitjançant un modal
Eliminació d'elements	Mostra un modal de confirmació abans d'eliminar un element
Valoració amb estrelles	Els usuaris poden valorar elements amb un sistema d'estrelles
Càlcul de la suma	La suma dels vots es calcula i es mostra en temps real
<u> </u>	·

Funcionalitat	Descripció
Selecció automàtica	La categoria es defineix automàticament segons el context (materials/monstres)
Camps específics	Mostra camps diferents segons si és un material o un monstre



Validacions

El projecte inclou diverses validacions per assegurar que les dades introduïdes pels usuaris siguin correctes abans de guardar-les. A continuació es detallen les validacions aplicades:

Validacions generals

Camp	Validació	Missatge d'error
Nom	Ha de tenir almenys 3 caràcters després d'eliminar els espais inicials i finals. Es permeten espais intermedis (exemple: "Nom Exemple").	"El nom ha de tenir almenys 3 caràcters."
Descripció	Ha de tenir almenys 3 caràcters després d'eliminar els espais inicials i finals. Es permeten espais intermedis (exemple: "Aquesta és una descripció.").	"La descripció ha de tenir almenys 3 caràcters."

Validacions específiques per a materials

Camp	Validació	Missatge d'error
Hearts	Ha de ser un número	"Els cors recuperats han de ser un número
Recovered	positiu.	positiu."

Validacions específiques per a monstres

Camp	Validació	Missatge d'error
Drops	Ha d'existir almenys un drop . Cada drop ha de tenir almenys 1 caràcter (no pot estar buit).	"Els monstres han de tenir almenys un drop." "Els drops no poden estar buits."

Comportament en cas d'errors

Quan es detecta un error en el formulari:

- Es mostra un missatge d'error en **color vermell** sota el camp corresponent.
- L'usuari no pot guardar l'element fins que es resolguin tots els errors.



1. Valoració amb estrelles

- Descripció: Els usuaris poden valorar materials i monstres amb un sistema d'estrelles (1 a 5).
- Implementació:
 - Els vots es guarden al backend i es mostren al frontend.
 - Es calcula el **total** de vots per a cada element.

2. Càlcul de la suma

- Descripció: La suma dels vots es calcula al frontend per representar millor les valoracions.
- Implementació:
 - Els vots es processen en un array, s'ordenen i es calcula la suma.

3. Selecció automàtica de categories

• **Descripció**: Quan es crea un nou element, la categoria es defineix automàticament segons si l'usuari està a la vista de materials o monstres.

4. Camps específics per a materials i monstres

- Materials:
 - Cooking Effect: Permet especificar l'efecte de cuina del material.
 - **Hearts Recovered**: Permet indicar el nombre de cors recuperats pel material. Ha de ser un número positiu.
- Monstres:
 - **Drops**: Permet especificar els objectes que deixa caure el monstre. Es poden afegir, editar i eliminar múltiples drops.

🐳 Instal·lació amb Docker



- Docker
- Docker Compose

Instruccions

1. Clona el repositori:

```
git clone https://github.com/usuari/zelda-botw-api.git
cd zelda-botw-api
```

2. Executa el projecte amb Docker Compose:

```
docker compose up --build
```

3. Accedeix a l'API a http://localhost:3001.



Instal·lació en local



- Node.js (versió 14 o superior)
- MongoDB

Instruccions

1. Clona el repositori:

```
git clone https://github.com/usuari/zelda-botw-api.git
cd zelda-botw-api
```

2. Instal·la les dependències:

```
npm install
```

3. Inicia el servidor:

```
npm start
```

4. Accedeix a l'API a http://localhost:3001.



- Pots fer servir Swagger a http://localhost:3001/api-docs per explorar i provar els endpoints.
- Les dades de materials i monstres estan basades en l'API original: https://botw-compendium.herokuapp.com.