



# UNAH

UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE  
INGENIERÍA EN SISTEMAS

**POO**

Examen Práctico Unidad II

Catedrático: Erick Marín

La presente evaluación es de carácter individual, evite cualquier tipo de plagio o será penalizado.

### Se le provee

- Frontend HTML, CSS, Bootstrap y JS:
- Backend funcional en NodeJS+Typescript del ejercicio realizado en clases virtuales
- Script de las colecciones de datos para mongoDB

El objetivo principal es crear una funcionalidad básica simulando partidas de Pokemon.

En el frontend proveído encontrará tres archivos html:

- index.html: Solo contiene dos enlaces para acceder dos páginas: Batalla pokemon y Capturar
- catch.html: Página para capturar pokemones (Ejercicio hecho en clase), el controlador de esta UI es catch-controlador.js
- game.html: UI para simular una partida pokemon, a esta UI deberá de darle funcionalidad integrándola con el backend. El controlador en el frontend de esta UI es game-controlador.js

En el frontend, solo debería trabajar en los archivos game.html y game-controlador.js (De preferencia)

El simulador para una batalla pokemon consiste en seleccionar quienes serán los contrincantes (extremo izquierdo y derecho), luego de seleccionar el contrincante se muestra el detalle con su lista de pokemones, se podrá seleccionar un pokemon para la batalla. Luego de seleccionar cada jugador su pokemon respectivo, se podrá dar click en el botón “Luchar” para que se verifique quien es el ganador dependiendo de sus debilidades.





## Funcionalidades:

- Modificar el servicio web de obtener maestros pokemon para que solo retorne los atributos; nombre, imagen y nivel. Con esa información renderizar el select list de cada jugador (extremo izquierdo es player 1 y el extremo derecho es player 2).
- Al seleccionar un jugador en cada extremo se debe consumir otro servicio web que obtenga los pokemons que ha capturado el usuario seleccionado.
- Al dar click sobre la miniatura de cada pokemon se tendrá que consumir el servicio para obtener todos los detalles de cada pokemon, estos se mostrarán en la sección central (La que tiene el background verde).
- Crear una colección de partidas (games), para almacenar cada partida jugada, la estructura de cada documento json a almacenar será la siguiente:

```
_id: Id de la partida  
player1: {_id,nombre, imagen}: Jugador 1  
player2: {_id,nombre, imagen}: Jugador 2  
winner: _id del jugador que ganó
```

- Se podrá dar click en el botón luchar cuando ambos jugadores hayan seleccionado sus pokemon, se calculará quien es el ganador verificando las debilidades de cada pokemon versus el tipo de pokemon contrincante. Este cálculo del ganador **deberá hacerse en el backend** mediante un servicio web que responderá con un json indicando el ganador y confirmando que se guardó la partida en la base de datos.

Ejemplo:

```
Pokemon 1  
  Nombre: Pikachu  
  Tipo (Type): [Electric]  
  Debilidades (Weaknesses): [Ground]  
  
Pokemon 2  
  Nombre: Vaporeon  
  Tipo (Type): [water]  
  Debilidades (Weaknesses): [Electric, Grass]
```

- El ganador en esta partida sería Pikachu ya que Vaporeon es débil al tipo Electric. Para la verificación preste atención a los campos Type(tipo) y Weaknesses(debilidades), ya que estos son arreglos, es decir un pokemon puede ser de varios tipos y puede tener varias debilidades.
- Una partida puede terminar en empate si no se puede concluir una ventaja por las debilidades de cada pokemon.
- En el UI agregar una opción para mostrar el listado de las partidas jugadas en una ventana modal. Crear el servicio web correspondiente en el backend.

## Observaciones:

- El backend debe ser con NodeJS y Typescript
- Para el frontend usar vanilla javascript
- Puede utilizar .ajax de jquery, axios o fetch para consumir el REST API
- Cree solo los servicios web del API Rest a utilizar, desarrolle los que sean necesarios para la solución.
- Estructure el proyecto en carpetas independientes para frontend y backend.
- Puede utilizar MongoDB Atlas o una instalación local. En cualquier caso debe adjuntar los exports de la colección de maestros pokemon y partidas para realizar una correcta revisión de su examen.
- Genere los ids de las colecciones utilizando la función de mongoDB.
- Puede utilizar sus apuntes, google y cualquier recurso, escriba su código y evite copiar y pegar (eso no es programar).
- Al momento de subir el proyecto solución, no incluya la carpeta node\_modules. **Incluya un pequeño script básico para la base de datos**

## Recomendaciones:



# UNAH

UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE  
INGENIERÍA EN SISTEMAS

**P00**

*Examen Práctico Unidad II*

**Catedrático: Erick Marín**

- Para cada acción verifique la estructura del json a utilizar y sus diferentes atributos.
- Enfóquese en su evaluación, aproveche el tiempo y no confunda a sus compañeros.
- Evite copiar código compartido en chats grupales, se penalizará exámenes donde se compruebe copia o plagio (ejem. Códigos similares con solo cambios de nombres de variables, funciones, orden del código, entre otros).