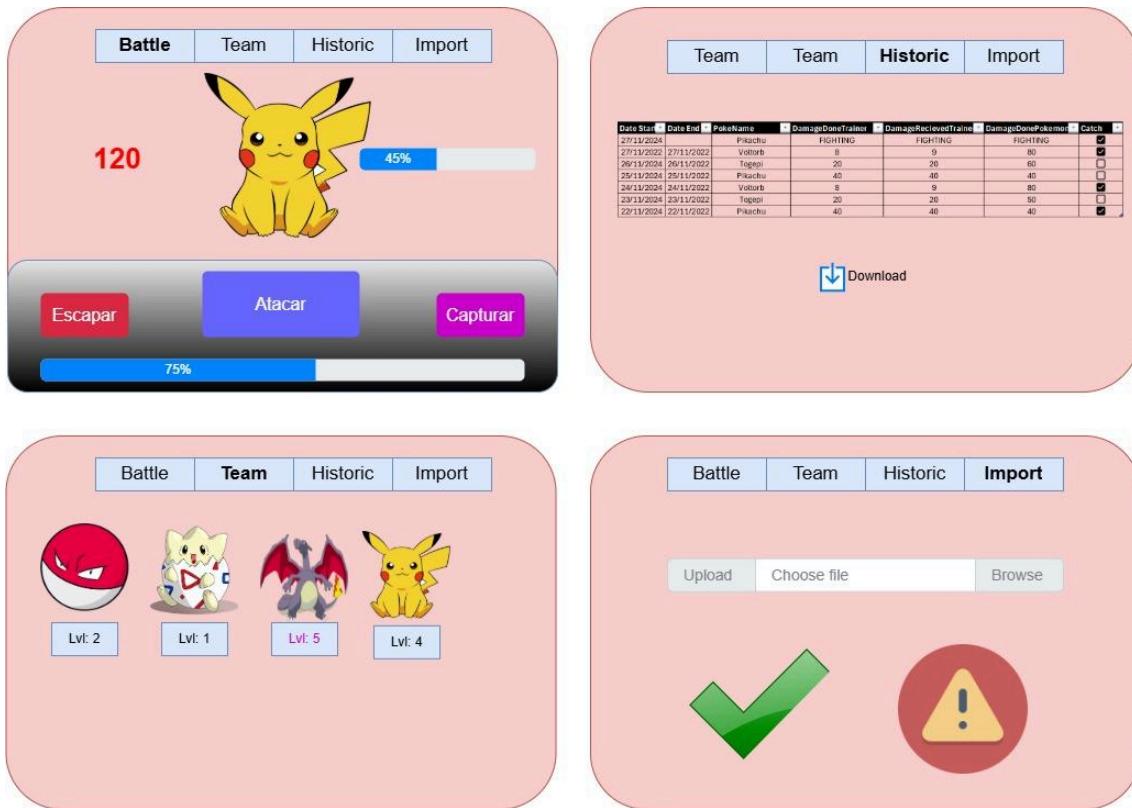


# PokeRogue

Se solicita crear una aplicación usando WPF, utilizando Visual Studio para la generación de la interfaz, con el objetivo de crear un pequeño juego pokémon.



The application consists of four main views:

- Battle View:** Shows a Pikachu icon with a red '120' above it and a blue '45%' below it. Below the icon are three buttons: 'Escapar' (red), 'Atacar' (blue), and 'Capturar' (purple). A blue progress bar at the bottom shows '75%'. The menu bar includes 'Battle' (selected), 'Team', 'Historic', and 'Import'.
- Team View:** Displays a table of pokémon statistics. The columns are Date Start, Date End, Pokéname, Type, DamageDoneTrainer, DamageReceivedTrainer, DamageDonePokémon, and Catch. The table contains the following data:

Date Start	Date End	Pokéname	Type	DamageDoneTrainer	DamageReceivedTrainer	DamageDonePokémon	Catch
29/11/2024	27/11/2022	Pikachu	FIGHTING	5	5	50	<input checked="" type="checkbox"/>
29/11/2024	29/11/2022	Voltorb		20	20	60	<input type="checkbox"/>
29/11/2024	29/11/2022	Togepi		8	9	80	<input checked="" type="checkbox"/>
29/11/2024	29/11/2022	Pikachu		40	40	40	<input type="checkbox"/>
29/11/2024	29/11/2022	Voltorb		20	20	50	<input type="checkbox"/>
29/11/2024	29/11/2022	Togepi		40	40	40	<input checked="" type="checkbox"/>
29/11/2024	29/11/2022	Pikachu		40	40	40	<input checked="" type="checkbox"/>

A 'Download' button with a downward arrow is located below the table.

- Historic View:** Shows a list of four pokémon icons with their levels: a red Pokéball (Lvl: 2), a yellow cat-like creature (Lvl: 1), a dark dragon-like creature (Lvl: 5), and a yellow Pikachu (Lvl: 4). The menu bar includes 'Battle', 'Team' (selected), 'Historic', and 'Import'.
- Import View:** Contains three buttons: 'Upload', 'Choose file', and 'Browse'. Below these are two large circular icons: a green checkmark on the left and a red warning triangle with an exclamation mark on the right. The menu bar includes 'Battle', 'Team', 'Historic', and 'Import'.

Todas las vistas se encuentran dentro del mismo Window y tendrán un menú que será fijo. Este menú especificará poniendo en **negrita** la vista seleccionada actualmente, por defecto, al entrar será Battle.

## Battle



Esta es la pantalla principal. En esta pantalla aparecerá una imagen de un Pokémon que obtendréis a partir del [pokeapi](#). El Pokémon seleccionado será un pokémon aleatorio. Para que sea aleatorio, buscaréis por un pokémon cuyo id esté contenido entre 0 y 100. El sprint será el front\_default, pero existe un 5% de posibilidades de que sea el front\_shiny.

Aparte del sprite, el pokémon tiene dos atributos:

- **Vida:** aparece a la derecha, será el porcentaje de vida del pokémon, por defecto es 100%, utilizad un slider. El valor numérico de la vida se obtendrá a través del API en el apartado stats, el stats cuyo name es HP.
- **Ataque:** aparecerá a la izquierda, es un número que se obtendrá a través del API en el apartado stats, el stats cuyo name es attack.

Ahora vamos con los datos del entrenador. Por defecto tienes 1000 de vida que aparece como un slider, además tienes 3 botones:

- **Botón escapar:** te aparece otro pokemon y se actualiza la vista con él.
- **Botón atacar:** atacar resta un número aleatorio entre 0 y 40 de la vida del Pokémon actual. Si la vida del pokémon es menor o igual que 0 te aparece otro pokemon. Si la vida es mayor que 0 después de atacar, entonces te atacará a ti y te restará de tu vida el número de su ataque.

- **Botón capturar:** cuando capturas un pokemon lo añade a tu equipo. La probabilidad de capturar un pokemon depende del porcentaje de vida que tiene el Pokémon. Si un Pokémon está el 45% de vida eso implica que tienes un 55% de posibilidades de poder capturarlo. Al capturar un pokémon se incrementa tu vida un 5%, tu vida como máximo puede ser 1000. Si capturas un shiny se recupera toda la vida.

## Team



En este UserControl te aparecerá un listado con los pokemons que has capturado. El listado de pokémon capturados **vendrá desde la API que tienes que crear**. Cuando capturas un pokemon, te aparecerá como lv 1, si ese pokemon lo vuelves a capturar entonces se aumentará a 1 el nivel, en este caso lv 2.

Si capturas un pokemon en su forma shiny, entonces te aparecerá en este listado con el sprite de shiny. Además el texto tendrá color morado.

**Toda la información tendrá que obtenerse desde el API.**

## Historic

Date Start	Date End	PokeName	DamageDoneTrainer	DamageReceivedTrainer	DamageDonePokemon	Catch
27/11/2024		Pikachu	FIGHTING	FIGHTING	FIGHTING	<input checked="" type="checkbox"/>
27/11/2022	27/11/2022	Voltorb	8	9	80	<input checked="" type="checkbox"/>
28/11/2024	28/11/2022	Togepi	20	20	60	<input type="checkbox"/>
25/11/2024	25/11/2022	Pikachu	40	40	40	<input type="checkbox"/>
24/11/2024	24/11/2022	Voltorb	8	9	80	<input checked="" type="checkbox"/>
23/11/2024	23/11/2022	Togepi	20	20	60	<input type="checkbox"/>
22/11/2024	22/11/2022	Pikachu	40	40	40	<input checked="" type="checkbox"/>

 Download

En este UserControl te aparecerá una tabla utilizando la clase **DataGrid** con las siguientes columnas:

- **DataStart:** fecha de inicio de la batalla: AÑO/MES/DIA HORAS:MINUTOS
- **DateEnd:** fecha de final de la batalla, ya sea capturado o derrotado, el formato es AÑO/MES/DIA HORAS:MINUTOS
- **PokeName:** Nombre del pokemon
- **DamageDoneTrainer:** La suma del daño hecho por el entrenador durante toda la batalla.
- **DamageReceivedTrainer:** La suma del daño recibido por el pokémon durante toda la batalla.
- **DamageDonePokemon:** la suma del daño recibido por el pokémon durante toda la batalla, pero con la excepción de que si el pokémon lo has capturado, se resta de ese daño realizado lo que has recuperado de vida, como mucho es 0.
- **Catch:** un checkbox que dice si los has capturado o no.

**Toda esta información se obtendrá desde el API.**

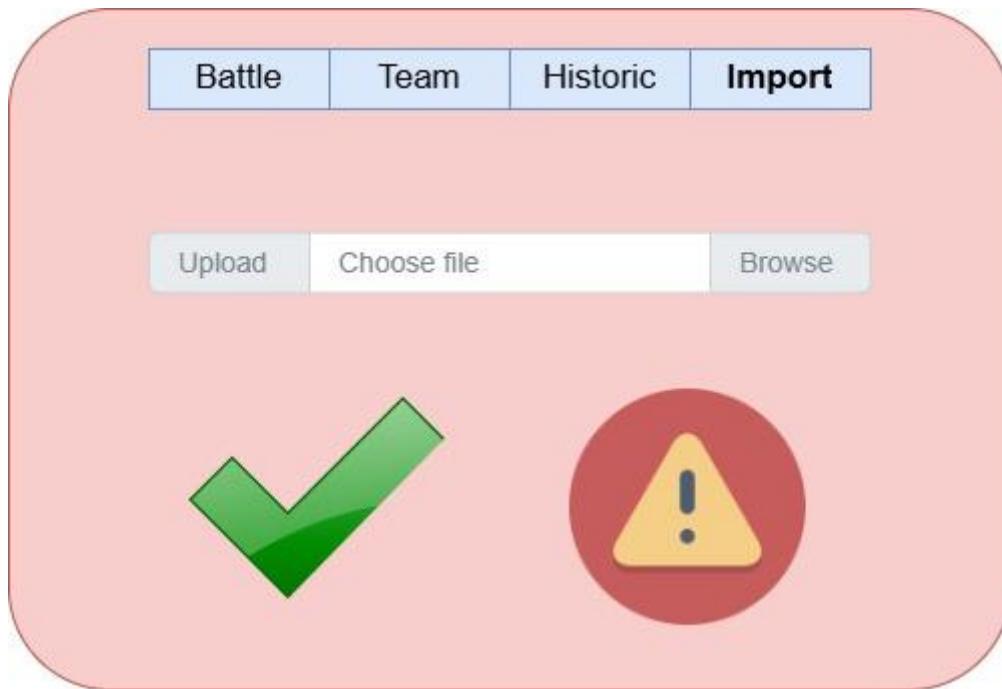
Además aparecerá un botón que te permite descargar en formato **json** los datos de la tabla usando un SaveFileDialog. El formato será el siguiente:

```
[  
 {  
   "DataStart": "2024/11/27 14:35",  
   "DateEnd": "2024/11/27 14:50",  
   "PokeName": "Pikachu",  
   "DamageDoneTrainer": 450,  
   "DamageReceivedTrainer": 320,  
   "DamageDonePokemon": 280,  
   "Image": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/132.png",  
   "Catch": true,  
   "Shiny": false  
 },  
 {  
   "DataStart": "2024/11/27 15:10",  
   "DateEnd": "2024/11/27 15:25",  
   "PokeName": "Charizard",  
   "DamageDoneTrainer": 600,  
   "DamageReceivedTrainer": 500,  
   "DamageDonePokemon": 0,  
   "Image": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/134.png",  
   "Catch": true,  
   "Shiny": true  
 }  
 ]
```

**Importante:** El campo imagen debe ser distinto cuando capturas un pokemon normal o shiny, front\_default,front\_shiny.

## Import

Esta ventana te permite cargar un archivo JSON que tendrá como formato el especificado en histórico. Tendréis que usar OpenFileDialog. **La importación sobreescribirá los datos de tu API.**



En el caso de que la operación se realice con éxito aparecerá abajo el icono del check verde, en el caso de que la importación se realice de manera incorrecta aparecerá el icono de error.

## Evaluación

Se valorará el uso de SOLID, KISS y DRY a la hora de realizar el código. En la rúbrica se especificarán los valores en cada entrega que serán necesarios llegar y en clase se comentará cómo obtenerlos.