



UNIVERSIDAD DEL BÍO-BÍO
La Universidad de la Región del Biobío

Bases de datos No Relacionales

Guía 2

Integrantes : Pablo Vallejos Alarcón

Docente: Tatiana Gutierrez Bunster

Fecha: 18/04/2024

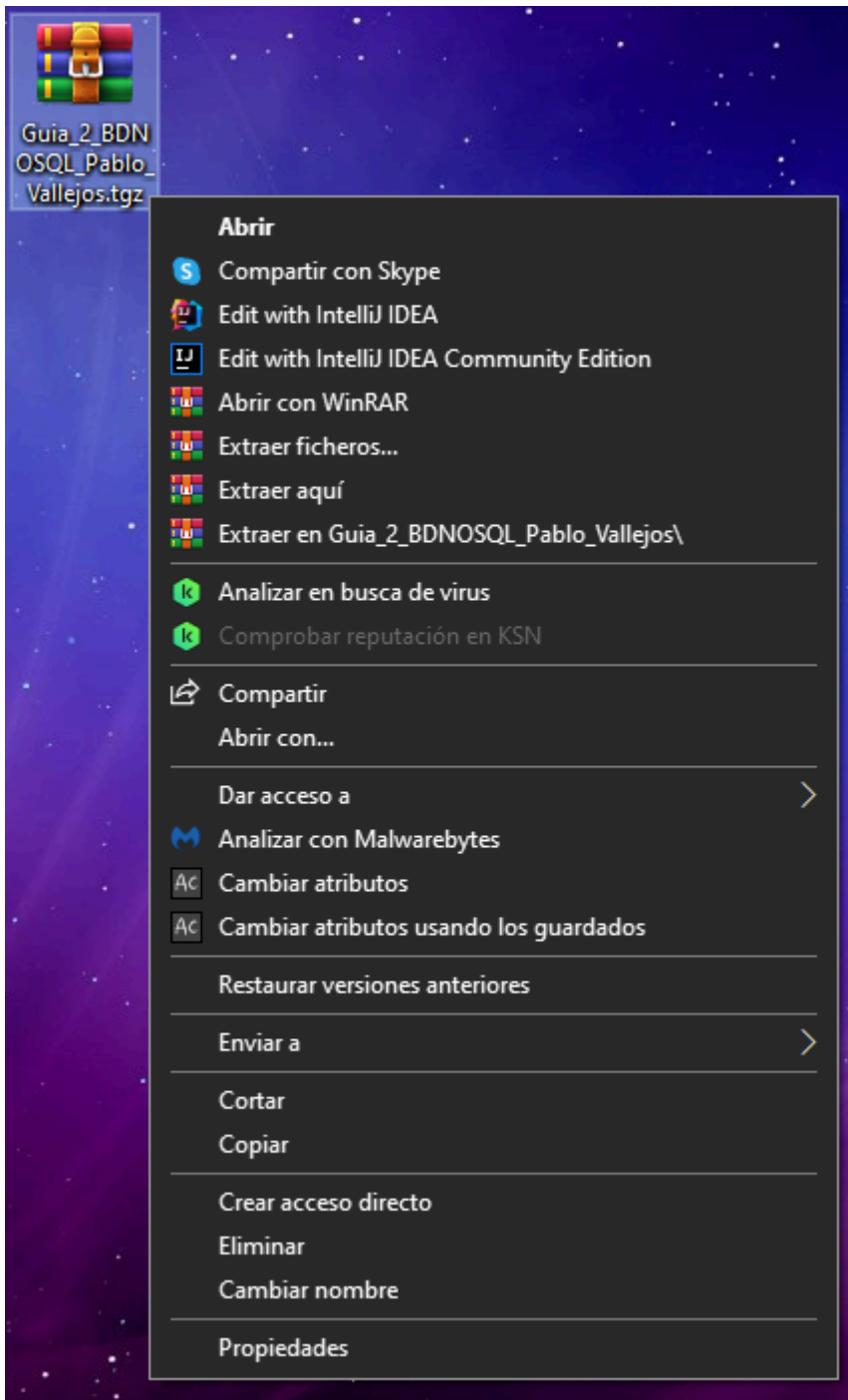
Introducción

En el siguiente informe se detalla el proceso y la construcción de una aplicación que ayuda a generar un gráfico utilizando la biblioteca D3.js a partir de los datos obtenidos de una API de Pokémon.

Para esto se han llevado una serie de pasos y procesos técnicos que incluyen la generación de Scripts de Python para conseguir la extracción, la carga y la transformación de datos provenientes de la API para ser almacenados en la base de datos no relacional “Mongo DB”, posteriormente la extracción de estos mismos datos a un archivo .JSON el cual será clave para finalmente la visualización del gráfico por medio de un navegador web utilizando HTML y JavaScript










Paso 1:

En primer lugar se deben extraer los archivos provenientes del ".tar.gz" recién descargado



Paso 2:


Una vez ya tenga los siguientes archivos exportados:

Nombre	Fecha de modificación	Tipo	Tamaño
 cargar.py	18-04-2024 21:03	Python File	1 KB
 charts.js	18-04-2024 21:21	Archivo JavaScript	5 KB
 conexionBD.py	15-04-2024 19:28	Python File	1 KB
 export_to_JSON.py	18-04-2024 21:04	Python File	1 KB
 extraer.py	18-04-2024 21:03	Python File	4 KB
 index.html	18-04-2024 21:15	Vivaldi HTML Doc...	1 KB
 main.py	18-04-2024 21:03	Python File	1 KB
 transformar.py	18-04-2024 21:03	Python File	2 KB
 verificar_pokemones.py	18-04-2024 21:03	Python File	1 KB

Abra MongoDB Compass y agregue una conexión "<mongodb://localhost:27017/>", luego presione **Connect**. Posterior a esto asegúrese de haberse conectado con éxito.

New Connection

Connect to a MongoDB deployment

 FAVORITE

URI ⓘ Edit Connection String ☒

> Advanced Connection Options

Save

Save & Connect

Connect

Paso 3:

Luego desde la terminal ejecute el archivo "**main.py**", este archivo hará que se agreguen datos en MongoDB provenientes de la API de pokemon

**/Cabe mencionar que el archivo fue testeado con el rango (1,30) pokemons siendo este un número recomendable para su ejecución y es el que por defecto viene en el archivo "main.py", sin embargo puede modificarse el tope del rango si se desea extraer más pokemons*

Terminal y Mongo DB:

```
PS C:\Users\pvall\Desktop\Bases de Datos No Relacionales\Guia 2 final> python main.py
```

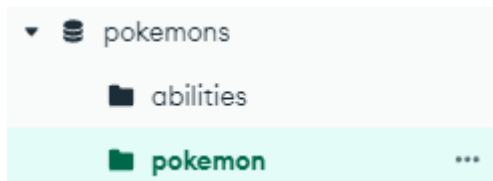
```
Se ha insertado un nuevo documento de Ability en la colección 'abilities'.  
Conexión cerrada con MongoDB  
Conexión exitosa con MongoDB  
Se ha insertado un nuevo documento de Pokémon en la colección 'pokemon'.  
Conexión cerrada con MongoDB
```

Código:

```
main.py > ...
1  from extraer import extract_ability_data, extract_pokemon_data
2  from transformar import transform_ability_data
3  from cargar import cargar_ability, cargar_pokemon
4
5  def main():
6      # Rango de IDs de Pokémon que deseas extraer, el numero "30" puede ser modificado según la cantidad de pokemones que deseas extraer
7      pokemon_ids = range(1, 30)
8
9      for pokemon_id in pokemon_ids:
10         ability_data = extract_ability_data(pokemon_id)
11         if ability_data:
12             transformed_ability_data = transform_ability_data(ability_data)
13             cargar_ability(transformed_ability_data)
14
15         pokemon_data = extract_pokemon_data(pokemon_id)
16         if pokemon_data:
17             cargar_pokemon(pokemon_data)
18
19 if __name__ == "__main__":
20     main()
21
22
```

Paso 4:

Refresque MongoDB y podrá ver los datos almacenados recientemente



```
_id: ObjectId('6621cd6abe011080313e8eb5')
id: 1
name: "bulbasaur"
base_experience: 64
height: 7
is_default: true
order: 1
weight: 69
▶ abilities: Array (2)
▶ forms: Array (1)
▶ game_indices: Array (20)
▶ held_items: Array (empty)
location_area_encounters: "https://pokeapi.co/api/v2/pokemon/1/encounters"
▶ moves: Array (86)
```

o si desea puede ejecutar desde terminal el archivo **"verificar_pokemones.py"** que le mostrará los datos registrados y la cantidad de pokemones ingresados

```
PS C:\Users\pvall\Desktop\Bases de Datos No Relacionales\Guia 2 final> python verificar_pokemones.py
```

```
{'_id': ObjectId('6621cd81be011080313e8f25'), 'id': 29, 'name': 'nidoran-f', 'base_experience': 55, 'height': 4, 'is_default': True, 'order': 5
7, 'weight': 70, 'abilities': ['poison-point', 'rivalry', 'hustle'], 'forms': ['nidoran-f'], 'game_indices': ['red', 'blue', 'yellow', 'gold',
'silver', 'crystal', 'ruby', 'sapphire', 'emerald', 'firered', 'leafgreen', 'diamond', 'pearl', 'platinum', 'heartgold', 'soulsilver', 'black',
'white', 'black-2', 'white-2'], 'held_items': [], 'location_area_encounters': 'https://pokeapi.co/api/v2/pokemon/29/encounters', 'moves': ['sc
ratch', 'cut', 'double-kick', 'headbutt', 'tackle', 'body-slam', 'take-down', 'double-edge', 'tail-whip', 'poison-sting', 'bite', 'growl', 'sup
ersonic', 'disable', 'ice-beam', 'blizzard', 'counter', 'strength', 'thunderbolt', 'thunder', 'dig', 'toxic', 'rage', 'mimic', 'double-team', '
defense-curl', 'reflect', 'focus-energy', 'bide', 'skull-bash', 'fury-swipes', 'rest', 'super-fang', 'substitute', 'thief', 'snore', 'curse', '
protect', 'sludge-bomb', 'mud-slap', 'detect', 'endure', 'charm', 'swagger', 'attract', 'sleep-talk', 'return', 'frustration', 'pursuit', 'iron
-tail', 'hidden-power', 'rain-dance', 'sunny-day', 'crunch', 'rock-smash', 'beat-up', 'flatter', 'facade', 'helping-hand', 'secret-power', 'poi
son-fang', 'aerial-ace', 'poison-tail', 'shock-wave', 'water-pulse', 'natural-gift', 'toxic-spikes', 'poison-jab', 'earth-power', 'shadow-claw'
, 'captivate', 'hone-claws', 'venoshock', 'round', 'echoed-voice', 'chip-away', 'confide', 'venom-drench']}
```

```
*****
Total de Pokémon en la colección 'pokemon': 29
*****
Conexión cerrada con MongoDB
```



Código:

```
verificar_pokemones.py > ...
1  from conexionBD import get_mongo_client, close_mongo_client
2
3  def verificar_pokemones():
4      client = get_mongo_client()
5      db = client["pokemons"]
6      collection = db["pokemon"]
7
8      pokemones = list(collection.find())
9      for pokemon in pokemones:
10         print(pokemon)
11         print()
12
13     print("\n" + "*" * 20)
14     print(f"Total de Pokémon en la colección 'pokemon': {len(pokemones)}")
15     print("*" * 20)
16     close_mongo_client(client)
17
18 if __name__ == "__main__":
19     verificar_pokemones()
```


Paso 5:

Una vez se haya asegurado del correcto ingreso de los datos, deberá ejecutar el archivo "**export_to_JSON.py**", luego de esto se le generará un archivo .JSON con el contenido de la base de datos, estos archivos se llamarán "**pokemon.JSON**" y "**abilities.JSON**"

```
PS C:\Users\pvall\Desktop\Bases de Datos No Relacionales\Guia 2 final> python export_to_JSON.py
Conexión exitosa con MongoDB
Se ha exportado la colección 'abilities' a 'abilities.json'.
Se ha exportado la colección 'pokemon' a 'pokemon.json'.
Conexión cerrada con MongoDB
PS C:\Users\pvall\Desktop\Bases de Datos No Relacionales\Guia 2 final> |
```

 abilities.json	18-04-2024 21:56	Archivo JSON	23 KB
 pokemon.json	18-04-2024 21:56	Archivo JSON	68 KB

Ejemplo contenido JSON:

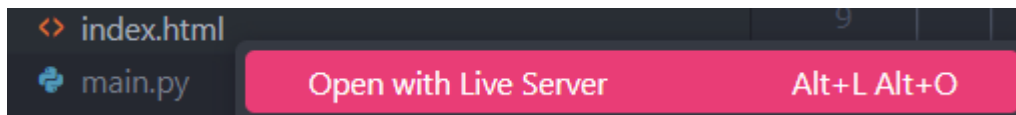
```
{ } pokemon.json > ...
1  [
2    {
3      "_id": "6621cd6abe011080313e8eb5",
4      "id": 1,
5      "name": "bulbasaur",
6      "base_experience": 64,
7      "height": 7,
8      "is_default": true,
9      "order": 1,
10     "weight": 69,
11     "abilities": [
12       "overgrow",
13       "chlorophyll"
14     ],
15     "forms": [
16       "bulbasaur"
```

Código:

```
export_to_JSON.py > ...
1  import json
2  from bson import ObjectId
3  from bson import json_util
4  from conexionBD import get_mongo_client, close_mongo_client
5
6  def custom_json_serializer(obj):
7      """Función de serialización personalizada para manejar los ObjectId."""
8      if isinstance(obj, ObjectId):
9          return str(obj)
10         raise TypeError(f"Object of type {obj.__class__.__name__} is not JSON serializable")
11
12  def export_to_json():
13      client = get_mongo_client()
14      db = client["pokemons"]
15
16      collections = ["abilities", "pokemon"]
17      for collection_name in collections:
18          collection = db[collection_name]
19          cursor = collection.find({})
20          data = list(cursor)
21          with open(f"{collection_name}.json", "w") as outfile:
22              json.dump(data, outfile, default=custom_json_serializer, indent=2)
23          print(f"Se ha exportado la colección '{collection_name}' a '{collection_name}.json'.")
24
25      close_mongo_client(client)
26
27  if __name__ == "__main__":
28      export_to_json()
```

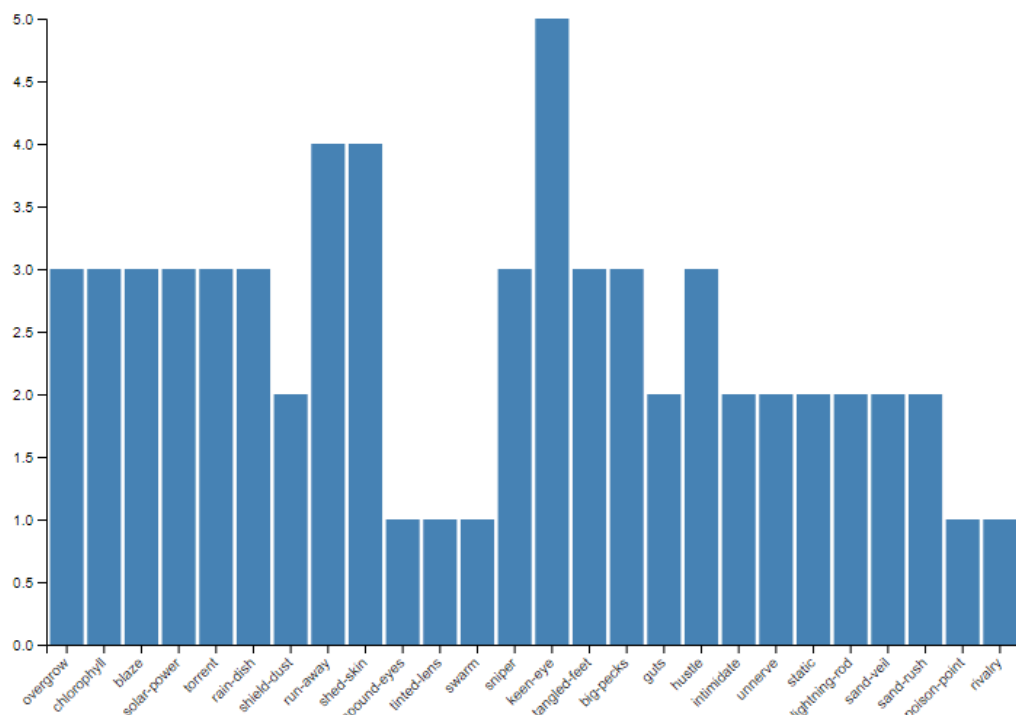
Paso 6:

Luego de haber exportado los archivos JSON , deberá abrir el archivo **"index.html"** pero, en un LIVE SERVER, si está en *Visual Studio Code* bastará con hacer click derecho sobre el archivo **"index.html"** y presionar la opción **"Open with Live Server"**, esta opción es recomendada para no tener problemas en la visualización del gráfico



Una vez sea redireccionado al navegador podrá observar el gráfico basado en **"Pokemones según sus habilidades"**:

Pokemones según Habilidades



Fragmento del código charts.js:

```
JS charts.js > drawAbilitiesChart > then() callback
1  function drawPokemonChart() {
2    d3.json('pokemon.json').then(function(data) {
3      var margin = { top: 20, right: 30, bottom: 50, left: 100 };
4      var width = 800 - margin.left - margin.right;
5      var height = 500 - margin.top - margin.bottom;
6
7      var svg = d3.select("#chart")
8        .append("svg")
9        .attr("width", width + margin.left + margin.right)
10       .attr("height", height + margin.top + margin.bottom)
11       .append("g")
12       .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
13
14      var allAbilities = data.flatMap(pokemon => pokemon.abilities);
15      var uniqueAbilities = Array.from(new Set(allAbilities));
16
17      var abilities = uniqueAbilities.map(ability => ({
18        ability,
19        count: data.filter(pokemon => pokemon.abilities.includes(ability)).length
20      }));
21
22      var xScale = d3.scaleBand()
23        .domain(abilities.map(d => d.ability))
24        .range([0, width])
25        .padding(0.1);
26
27      var yScale = d3.scaleLinear()
28        .domain([0, d3.max(abilities, d => d.count)])
29        .range([height, 0]);
30
31      svg.selectAll("rect")
32        .data(abilities)
33        .enter()
34        .append("rect")
35        .attr("x", d => xScale(d.ability))
36        .attr("y", d => yScale(d.count))
37        .attr("width", xScale.bandwidth())
38        .attr("height", d => height - yScale(d.count))
39        .attr("fill", "steelblue");
40    });
  }
```

HTML:

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Gráfico</title>
7      <script src="https://d3js.org/d3.v7.min.js"></script>
8      <style>
9          .chart {
10              display: inline-block;
11              margin-right: 20px;
12          }
13      </style>
14  </head>
15  <body>
16      <h1></h1>
17      <div class="chart">
18          <h2>Pokemones según Habilidades</h2>
19          <div id="chart"></div>
20      </div>
21      <div class="chart">
22
23          <script src="charts.js"></script>
24      </div>
25  </body>
26  </html>
```

Conclusión

A lo largo de este trabajo, he conocido más sobre diversas herramientas y tecnologías para el procesamiento y visualización de datos. Desde la extracción de datos de una API hasta su almacenamiento en una base de datos y posterior visualización en forma de gráfico interactivo, he adquirido habilidades prácticas en el manejo de flujos de datos. Además, he comprendido la importancia de la planificación y ejecución ordenada de pasos técnicos para lograr resultados efectivos. Este proyecto me ha proporcionado una visión integral sobre el manejo de datos en aplicaciones web, reforzando mis conocimientos en análisis de datos.