

Prueba - Desarrollo de interfaces interactivas con framework Vue

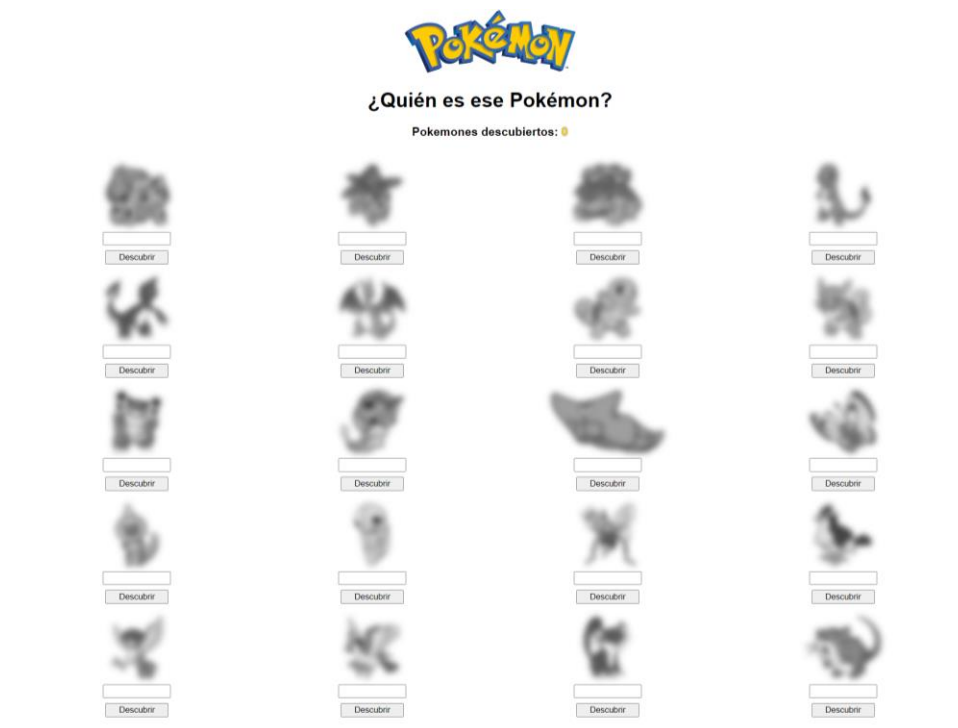
En esta prueba validaremos nuestros conocimientos del framework Vue Js obtenidos a lo largo de este módulo. Para lograrlo, necesitarás usar todas las competencias aprendidas en las unidades para crear una aplicación con temática de la serie Pokémon haciendo uso de la [PokéAPI](#).

Lee todo el documento antes de comenzar el desarrollo individual para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Entre los capítulos de Pokémon aparecía un momento de interacción con los espectadores donde se debía adivinar el nombre de un pokemon viendo solo la silueta.


La aplicación que deberás construir sigue esta misma temática mostrando 20 pokemones cuyas imágenes poseen en primera instancia un filtro que no deja ver con claridad cuál pokémon es.















Para que el usuario intente descubrir y adivinar cada pokémon deberás permitir el ingreso de su nombre por medio de un input debajo de cada una de las imágenes.



Al presionar en el botón “Descubrir” o simplemente apretar la tecla Enter, en caso de que el usuario haya escrito correctamente el nombre del pokémon deberá descubrirse quitando el filtro aplicado por defecto, de lo contrario se debe mostrar una ventana emergente indicando que el nombre ingresado es incorrecto.


¿Quién es ese Pokémon?
Pokémones descubiertos: 3

 bulbasaur	 <input type="text"/> <input type="button" value="Descubrir"/>	 <input type="text"/> <input type="button" value="Descubrir"/>	 charmander
 <input type="text"/> <input type="button" value="Descubrir"/>	 charizard	 <input type="text"/> <input type="button" value="Descubrir"/>	 <input type="text"/> <input type="button" value="Descubrir"/>
			

Al descubrir un pokémon se deben tomar 2 consideraciones:

1. Ocultar el input y el botón y dejar el nombre del pokémon descubierto



bulbasaur

2. Mostrar un contador con la cantidad de pokémons descubiertos hasta el momento.



¿Quién es ese Pokémon?

Pokémones descubiertos: 3 ←

Consultando el siguiente endpoint podrás ver 20 nombres de pokémones y las URL que te entregará el detalle de cada uno, por lo que no es necesario que conozcas los nombres de los pokémones para realizar este desafío y comprobar su correcto funcionamiento.

<https://pokeapi.co/api/v2/pokemon>

```
// https://pokeapi.co/api/v2/pokemon

{
  "count": 1154,
  "next": "https://pokeapi.co/api/v2/pokemon?offset=20&limit=20",
  "previous": null,
  "results": [
    {
      "name": "bulbasaur", ←
      "url": "https://pokeapi.co/api/v2/pokemon/1/"
    },
    {
      "name": "ivysaur", ←
      "url": "https://pokeapi.co/api/v2/pokemon/2/"
    },
    {
      "name": "venusaur".
    }
  ]
}
```

Requerimientos

1. Usar las directivas para enlazar variables del estado con el template.
(1 Punto)
2. Importar y ocupar componentes hijos que reciban datos a través de **props**.
(1 Punto)
3. Renderizar dinámicamente componentes hijos usando el v-for.
(1 Punto)
4. Usar los eventos para agregar interacciones en la aplicación y enlazarlos con métodos locales.
(1 Punto)
5. Comunicar componentes hijos con su padre para la ejecución de un método.
(1 Punto)
6. Usar la renderización condicional de elementos para mostrar u ocultar contenido.
(1 Punto)
7. Asignar condicionalmente estilos o clases por medio del style y class binding.
(1 Punto)
8. Utilizar Axios para el consumo de la API.
(1 Punto)
9. Hacer uso del ciclo de vida para ejecutar código al cargar la aplicación.
(1 Punto)
10. Utilizar computed properties para el procesamiento de datos del estado.
(1 Punto)



¡Mucho éxito!

Consideraciones y recomendaciones

1. La PokeAPI retorna un JSON con 20 pokemones usando el siguiente endpoint:
<https://pokeapi.co/api/v2/pokemon>

```
"results": [  
  {  
    "name": "bulbasaur",  
    "url": "https://pokeapi.co/api/v2/pokemon/1/"  
  },  
  {  
    "name": "ivysaur",  
    "url": "https://pokeapi.co/api/v2/pokemon/2/"  
  },  
]
```

Para obtener la URL de la imagen de cada pokémon deberás consumir cada uno de los endpoint que contiene el atributo **url** dentro de cada objeto en este arreglo **results**.

2. El filtro que oculta cada pokemon se puede conseguir con la siguiente línea de CSS:

```
filter: blur(5px) grayscale(100%);
```