



Pokedex webapp

[2022] Documento Funcional

Versión 1.0

26 de mayo, 2022

© Itla 2022

Autopista Las Américas, Km. 27, PCSD,

La Caleta, Boca Chica 11606.

Tel. 809-738-4852

Índice

Objetivo general	3
Funcionalidades generales	3
Home	3
Mantenimiento de pokémones	4
Mantenimiento de regiones	6
Mantenimientos de tipos	7
Consideraciones generales	8

Objetivo general

- Crear un pokedex app, utilizando node.js con sequelize para el manejo de base de datos.


Funcionalidades generales

Home

El sistema en su pantalla inicial tendremos un menú que debe tener la siguientes opciones:

- Una opción para ir al home
- Una opción con el nombre de pokemones (Debe llevar el mantenimiento de pokemones)
- Una opción con el nombre de regiones (Debe llevar el mantenimiento de regiones)
- Una opción con el nombre de Tipos de pokemones (Debe llevar el mantenimiento de tipos)

Ejemplo:



Home Pokemones Regiones Tipo de pokemones

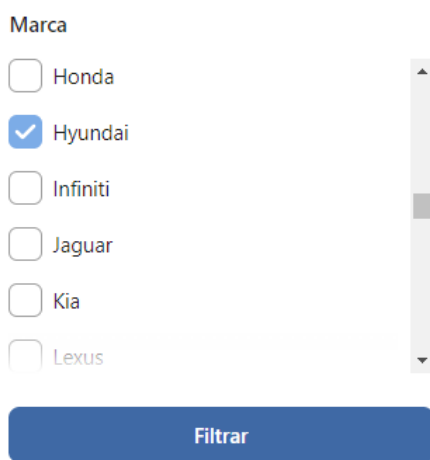
En esta pantalla de home tendrá un listado con todos los pokemones creado en el sistema, de los pokemones se debe mostrar el nombre del pokémon, una imagen del mismo, el tipo que tiene y la región a la cual pertenece.

Aparte del listado en esta pantalla debemos tener la posibilidad de realizar los siguientes filtros o búsqueda:

Se debe poder realizar búsqueda por el nombre del pokémon, para esto debe haber un formulario con un input de texto donde el usuario coloque el nombre del pokemon y un botón para realizar la búsqueda.

Por otra parte, se debe tener un formulario para poder filtrar los pokemones por las regiones para esto debe haber un listado con todas las regiones creada en el sistema de manera que podamos seleccionar una de ellas y al pulsar sobre un botón de filtrar se muestre solo los pokemones de esa región, ejemplo: si seleccionan la región de kanto solo saldrá los pokemones de esa región.

Ejemplo de filtro:



Marca

Honda

Hyundai

Infiniti

Jaguar

Kia

Lexus

Filtrar

Mantenimiento de pokémones

Al ingresar a la opción de pokémones desde el menú principal de la app, esta nos enviará al mantenimiento de pokemones.

En la pantalla inicial de este mantenimiento debemos tener un listado con todos los pokemones creado en el sistema, de los pokemones se debe mostrar el nombre del pokémon, una imagen del mismo, el tipo que tiene y la región a la cual pertenece y aparte de mostrar esta información cada pokémon debe tener un botón para editar el pokemon y otro para eliminarlo.

Arriba del listado de pokemones debe haber un botón para crear un pokemon nuevo, al hacer clic sobre este botón de crear se enviará al usuario a una pantalla nueva donde habrá un formulario con los siguientes campos: Nombre del pokemon(input text), foto del pokemon(input text), región a la que pertenece el pokemon(select con las regiones creada en el sistema), tipo primario(select

con los tipos de pokemones creado en el sistema).

Este formulario debe tener las siguientes validaciones que tanto el nombre, como la región, la foto y el tipo primario son campos requeridos.

Al final del formulario de creación debe haber dos botones, un botón para volver atrás que devuelve a la pantalla inicial del mantenimiento donde están listados los pokemones y otro botón para crear el pokémon, al hacer clic en el botón se envía la información y se crea el pokemon, una vez creado el pokemon se debe redirigir a la pantalla inicial de mantenimiento de pokemones.

Por otra parte, En el listado de pokemones del mantenimiento, cada pokémon tiene su botón para editarlo o borrarlo, al presionar sobre el botón editar se debe enviar al usuario a una pantalla con un formulario con los siguientes campos: Nombre del pokemon(input text), foto del pokemon(input text), región a la que pertenece el pokemon(select con las regiones creada en el sistema), tipo primario(select con los tipos de pokemones creado en el sistema), como es una edición estos campos debe venir con los valores guardados para el pokemon que estamos editando, incluso mostrar la imagen actual que tiene el pokemon.

Este formulario de edición debe tener las siguientes validaciones que tanto el nombre, como la región, la foto y el tipo primario son campos requeridos.

Al final del formulario de edición debe haber dos botones, un botón para volver atrás que devuelve a la pantalla inicial del mantenimiento donde están listados los pokemones y otro botón para guardar el pokémon el cual al hacer clic en el botón se envía la información y se edita el pokémon con los valores nuevo, una vez editado el pokemon se debe redirigir a la pantalla inicial de mantenimiento de pokemones.

Finalmente al presionar sobre el botón de eliminar se envía al usuario a una pantalla con un mensaje que debe decir: ¿Está seguro que desea eliminar este pokemon? debajo de este mensaje dos botones uno de cancelar y otro de aceptar, si se pulsa sobre cancelar se envía a la pantalla inicial del mantenimiento de pokemones(El listado), pero al presionar sobre el aceptar se elimina el pokemon y se redirecciona a la pantalla inicial del mantenimiento de pokemones(El listado).

Mantenimiento de regiones

Al ingresar a la opción de regiones desde el menú principal de la app, esta nos enviará al mantenimiento de regiones.

En la pantalla inicial de este mantenimiento debemos tener un listado con todas las regiones creadas en el sistema, de las regiones se debe mostrar el nombre de la región y aparte de mostrar esta información cada región debe tener un botón para editar la región y otro para eliminarla.

Arriba del listado de regiones debe haber un botón para crear una región nueva, al hacer clic sobre este botón de crear se enviará al usuario a una pantalla nueva donde habrá un formulario con los siguientes campos: Nombre de la región.

Este formulario debe tener las siguientes validaciones para que el nombre de la región sea un campo requerido.

Al final del formulario de creación debe haber dos botones, un botón para volver atrás que devuelve a la pantalla inicial del mantenimiento donde están el listado de las regiones y otro botón para crear la región, al hacer clic en el botón se envía la información y se crea la región, una vez creada la región se debe redirigir a la pantalla inicial de mantenimiento de regiones.

Por otra parte, En el listado de regiones del mantenimiento, cada región tiene su botón para editarla o borrarla, al presionar sobre el botón de editar se debe enviar al usuario a una pantalla con un formulario con los siguientes campos: Nombre de la región, como es una edición estos campos debe venir con los valores guardados para la región que estamos editando

Este formulario debe tener las siguientes validaciones para que el nombre de la región sea un campo requerido.

Al final del formulario de edición debe haber dos botones, un botón para volver atrás que devuelve a la pantalla inicial del mantenimiento donde están el listado de las regiones y otro botón para guardar la región el cual al hacer clic en el botón se envía la información y se edita la región con los valores nuevo, una vez editada la región se debe redirigir a la pantalla inicial de mantenimiento de regiones.

Finalmente al presionar sobre el botón de eliminar se envía al usuario a una pantalla con un mensaje que debe decir: ¿Está seguro que desea eliminar esta región? debajo de este mensaje dos botones uno de cancelar y otro de aceptar, si se pulsa sobre cancelar se envía a la pantalla inicial del mantenimiento de regiones(El listado), pero al presionar sobre el aceptar se elimina la región y se redirecciona a la pantalla inicial del mantenimiento de regiones(El listado).

Mantenimientos de tipos

Al ingresar a la opción de tipos de pokemones desde el menú principal de la app, esta nos enviará al mantenimiento de tipos.

En la pantalla inicial de este mantenimiento debemos tener un listado con todos los tipos creados en el sistema, de los tipos se debe mostrar el nombre del tipo y aparte de mostrar esta información de cada tipo debe tener un botón para editar el tipo y otro para eliminarlo.

Arriba del listado de tipos de pokemones debe haber un botón para crear un tipo nuevo, al hacer clic sobre este botón de crear se enviará al usuario a una pantalla nueva donde habrá un formulario con los siguientes campos: Nombre del tipo.

Este formulario debe tener las siguientes validaciones para que el nombre del tipo sea un campo requerido.

Al final del formulario de creación debe haber dos botones, un botón para volver atrás que devuelve a la pantalla inicial del mantenimiento donde están el listado de los tipos y otro botón para crear el tipo, al hacer clic en el botón se envía la información y se crea el tipo, una vez creado el tipo se debe redirigir a la pantalla inicial de mantenimiento de tipos.

Por otra parte, En el listado de tipos del mantenimiento, cada tipo tiene su botón para editarlo o borrarlo, al presionar sobre el botón de editar se debe enviar al usuario a una pantalla con un formulario con los siguientes campos: Nombre del tipo, cómo es una edición estos campos debe venir con los valores guardados para el tipo que estamos editando.

Este formulario debe tener las siguientes validaciones para que el nombre del tipo sea un campo requerido.

Al final del formulario de edición debe haber dos botones, un botón para volver atrás que devuelve a la pantalla inicial del mantenimiento donde están el listado de los tipos y otro botón para guardar el tipo el cual al hacer clic en el botón se envía la información y se edita el tipo con los valores nuevo, una vez editado el tipo se debe redirigir a la pantalla inicial de mantenimiento de tipos.

Finalmente al presionar sobre el botón de eliminar se envía al usuario a una pantalla con un mensaje que debe decir: ¿Está seguro que desea eliminar este tipo? debajo de este mensaje dos botones uno de cancelar y otro de aceptar, si se pulsa sobre cancelar se envía a la pantalla inicial del mantenimiento de tipo(El listado), pero al presionar sobre el aceptar se elimina el tipo y se redirecciona a la pantalla inicial del mantenimiento de tipos(El listado).

Consideraciones generales

- Deben aplicar la arquitectura de MVC en node.js (Usando express.js)
- Se debe usar Sequelize para la persistencia de los datos
- El proyecto tiene que ser visualmente entendible (Usar bootstrap)
- Se debe configurar un layout para la app