Generalitat de Catalunya
Consorci per a la Formació Contínua de Catalunya

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
SERVICIO PÚBLICO DE EMPLEO ESTATAL
SEPE

Vue JS

Crear una Pokedex i opcionalment, un combat de cartes

Introducció

L'objectiu d'aquesta pràctica es migrar a Vue la Pokedex creada a la Unitat 3 amb JavaScript.

Vue JS és un framework de JavaScript que permet crear aplicacions i webs optimitzant el procés de desenvolupament i utlitzant funcionalitats que costaria cert esforç crear-les desde zero amb JavaScript natiu.
Vue es pot treballar directament com una llibreria més o es pot treballar tot el projecte al voltant de Vue. Es veuran les dues maneres als exemples.
Aquesta entrega es treballarà utilitant tot el framework de Vue, utilitzant Vue CLI o Vue UI (si es prefereix un entorn més visual).

Tasques de l'exercici

Crear un nou projecte amb Vue CLI o Vue UI amb les especificacions indicades a la documentació que us anirem facilitant al llarg del mòdul.

L'objectiu final és replicar la mateixa pràctica treballada a la Unitat 3, però aquest cop creada a partir de Vue.

Crear una aplicació amb Vue que tingui 2 vistes (l'equivalent a pàgines en HTML). La vista principal rebrà el nom de Llista de Pokemons i és on hi haurà una llista de Pokemons amb informació bàsica (nom, imatge, valor d'atac i valor de defensa), on al seleccionar-ne un, es mostrarà una vista individual per mostrar més informació del Pokemon seleccionat (nom, imatge, valor d'atac, valor de defensa i tipus).

La tercera vista, és opcional, i s'anomenarà Combat. Aquesta vista haurà de mostrar un llistat de Pokemons com si fóssin cartes i estiguéssin girades boca per avall. De manera que no sabem quin Pokemon hi ha a cada carta. S'hauran de seleccionar 2 cartes fent clic, aquestes s'hauran de girar per revelar el Pokemon de cada carta i s'hauran de comparar els valors d'atac i defensa per decidir quin dels 2 guanya el combat.

També s'ha de crear un apartat de navegació (router) que permeti navegar entre totes les vistes.

Vista Llista de Pokemons

Al carregar la vista, s'ha d'utilitzar un servei per fer una consulta a la API de Pokemons (PokeAPI) i recuperar les dades de 10 Pokemons de forma aleatoria. Cadascun dels Pokemons s'ha de mostrar maquetat com una targeta o carta i ha de contindre la següent informació: nom, atac, defensa i una imatge. Cadascuna de les targetes o cartes ha de ser un component.

Servei de consulta de Pokemons

S'ha de crear un servei que gestioni totes les crides a la API, i cridar aquest servei des de les vistes quan es requereixi.

Component Carta Pokemon

S'ha de crear component anomenat Carta que representi cadascun dels Pokemons de la llista. Un cop s'hagi recupertat tota la informació de la consulta a la API, aquesta informació s'ha de mostrar, iterant tots els Pokemons resultants, i per cada un d'ells s'ha de pintar una carta.

Per tal de reutilitzar codi, es crearà un component Carta que, creat un sol cop, es podrà reutilizar tants cops com calgui. Haurà de rebre la informació del Pokemos com a paràmetre i haurà de contenir la lògica necessària per carregar la vista individual quan el fagi click a l'enllaç o bóto per veure la informació complerta.

Vista individual de Pokemon

Les targetes (components) de la vista principal també han de contenir un botó o un enllaç (router link) que permeti carregar una vista amb informació més detallada del Pokemon.

Aquesta vista ha de contenir la mateixa informació que es mostra en la pàgina inividual de l'exercici fet en JavaScript.

Filtre de cerca

La vista Llista de Pokemons ha de tenir un filtre textual de Pokemons. Ha de ser un input en el que, a mesura que es van escrivint lletres, només s'han de mostrar aquelles cartes amb Pokemon que el seu nom contingui el text escrit. Aquesta comprovació s'ha de fer per cada lletra que s'escrigui al filtre.

Per exemple, si tenim les cartes: Charmander, Bulbasaur, Charizard i Blastoise, si s'escriu al filtre “cha”, només han de ser visibles les cartes Charmander i Charizard.

Configuració transversal: selecció de tema clar/fosc

A la barra superior de menú, a banda dels enllaços per poder navegar de la pàgina d'inici a la de combat i viceversa, també hi ha d'haver un selector de tipus radio button amb 2 opcions: clar i fosc. Aquest selector, per defecte ha d'estar seleccionat amb l'opció de tema clar.

Els estils inicials de la pàgina i les targetes han de ser amb un color de fons clar amb una tipografia de color fosc. Si l'usuari selecciona l'opció de tema fosc, els estils de la pàgina han de canviar i els colors de fons i de les targetes han de ser foscos i la tipografia ha de ser clara (sempre a la inversa del color de fons per tal que hi ha un bon contrast i la legibilitat sigui accessible).

La selecció que fagi l'usuari s'ha de guardar al navegador com a Local Storage, de manera que quan l'usuari torni a carregar la web, el sistema recordi les seves preferències. Si havia seleccionat el mode fosc, la web s'ha de carregar directament en mode fosc. I el mateix pel mode clar. Si l'usuari no ha seleccionat cap preferència, s'haurà de carregar en mode clar.

Opcionalment es pot configurar perquè l'aplicació es carregui segons estigui configurat el navegador o el sistema operatiu de l'usuari. En aquest cas, s'haurà d'afegir una tercera opció al radio button que sigui "sistema" i que agafi els estils segons l'usuari tingui configurat el seu equip (clar o fosc).

Vista de combat (opcional)

Aquest apartat és opcional. Si es vol implementar, es recomana primer tenir acabada la part obligatòria i deixar aquesta pel final.

La vista de combat, al carregar-se ha de fer una consulta i retornar 10 Pokemons aleatoris. Aquests s'han de mostrar en format carta com a

l'apartat anterior, però inicialment tota la informació ha d'estar oculta. Les cartes han d'estar “girades” com si fóssin cartes boca avall i només es mostra la part posterior de la targeta. Per tant, inicialment l'usuari només veurà 10 cartes totes igual.

Per generar la llista s'ha d'utilitzar el servei corresponent.

Cadascuna de les cartes ha de ser un component CartaLluita, amb la seva lògica corresponent per poder realitzar els combats.

L'usuari haurà de seleccionar dos targetes fent clic sobre les que vulgui. Les targetes selecciones hauran mostrar la informació de la targeta: nom, imatge, atac i defensa.

Un cop seleccionades les dues cartes, aquestes entren en combat seguint la següent lògica:

- La primera carta seleccionada és la carta atacant. S'utilitzarà el seu valor “atac”.
- La segona carta seleccionada és la carta que defensa. S'utilitzarà el seu valor “defensa”.
- Si el valor d'atac de la primera carta és superior al valor de defensa de la segona, la primera carta guanya el combat. S'ha de mostrar un missatge “{nom del Pokemon de la primera carta} ataca i guanya a {nom del Pokemon de la segona carta}”.
- Si, en canvi, el valor d'atac de la primera carta és igual o inferior al valor de defensa de la segona carta, guanya la segona carta. El missatge ha de ser: “{nom del Pokemon de la primera carta} ataca i perd contra {nom de Pokemon de la segona carta}”.

A continuació s'ha d'activar un botó (que inicialment ha d'estar desactivat) per tornar la pàgina i les cartes al seu estat inicial i poder tornar a efectuar un nou combat.

Per mostrar la informació de la targeta, es pot fer de cop o fent ús d'alguna animació amb CSS. Es pot utilitzar l'efecte anomenat flip card, que simula una targeta en 3D dontant la volta.

A tenir en compte

No cal estar familiaritzat amb el món de Pokemon. S'ha decidit utilitzar aquesta API perquè és molt complerta i és gratuïta. És una API àmpliament utilitzada en exemples de programació de treball amb API, i és fàcil trobar exemples.

Objectius

El principal objectiu d'aquesta pràctica és treballar amb Vue sobre un exercici que ja s'ha treballat amb JavaScript, per veure les diferències que hi ha a l'hora de treballar amb un framework. Encara que la base de Vue sigui JavaScript, cosa que facilita la corba d'aprenentage, es treballa de manera diferent.

Es veurà com treballar amb una API externa (https://pokeapi.co/), realitzant operacions de consulta, i s'aprendrà a interpretar les dades rebudes en format JSON. Per fer les consultes, enlloc d'utlitzar la Fetch API, s'utlitzarà Axios, una llibreria específica per fer consultes d'aquest tipus.

Finalment es treballarà amb Local Storage, per aprendre a utilitzar el navegador per guardar informació sense haver de dependre d'una Base de dades.

Recursos recomanats

https://vuejs.org
Sessió online del dilluns 19 de setembre
Sessió online dimecres 21 de setembre
Search Gif Vue - repositori
Repositori del codi dels videos vue
lenguajejs.com/vuejs/componentes/ciclo-vida
https://vitejs.dev/
https://nodejs.org/