

Integración de sistemas

Examen Práctico Progreso 2

(Estudiante) Sahid Carchi

APP WEB - IDE VisualCode - Framework REACT

Se desarrollo una aplicacion web con REACT para visualizar la informacion de la API

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
import Papa from 'papaparse';
import './App.css';

const App = () => {
  const [pokemons, setPokemons] = useState([]);
  const [nombreCliente, setNombreCliente] = useState('');

  // MANDO A LLAMAR LOS DATOS E IMAGEN DEL POKEMON
  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get('https://pokeapi.co/api/v2/pokemon?limit=151');
        const pokemonResults = await Promise.all(
          response.data.results.map(async (pokemon) => {
            const pokemonData = await axios.get(pokemon.url);
            return {
              id: pokemonData.data.id,
              name: pokemonData.data.name,
              image: pokemonData.data.sprites.front_default,
              stock: 100,
            };
          })
        );
        setPokemons(pokemonResults);
      } catch (error) {
        console.error('Error fetching Pokémon data:', error);
      }
    };
    fetchData();
  }, []);
}
```

Llamado de la API para obtener datos e Imagen de un Pokemon

Generación de los Archivo CSV (Orden Compra - Factura Cliente)

```
// GENERO EL CSV DEL POKEMON COMPRADO
const generateCSV = (pokemon) => {
  const csvData = [
    ['ID', 'Name', 'Stock'],
    [pokemon.id, pokemon.name, pokemon.stock - 1]
  ];

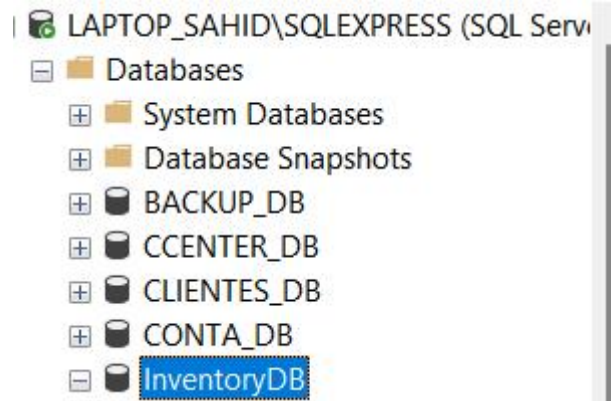
  const csv = Papa.unparse(csvData);
  const blob = new Blob([csv], { type: 'text/csv;charset=utf-8;' });
  const link = document.createElement('a');
  link.href = URL.createObjectURL(blob);
  link.download = `${pokemon.name}_purchase.csv`;
  link.click();
};

// GENERO EL ARCHIVO CSV DE LA FACTURA
const generateClienteCSV = (pokemon) => {
  const csvData = [
    ['Nombre del Cliente', 'ID', 'Pokemon Comprado'],
    [nombreCliente, pokemon.id, pokemon.name]
  ];

  const csv = Papa.unparse(csvData);
  const blob = new Blob([csv], { type: 'text/csv;charset=utf-8;' });
  const link = document.createElement('a');
  link.href = URL.createObjectURL(blob);
  link.download = `${pokemon.name}_cliente.csv`;
  link.click();
};
```

BASE DE DATOS - SQL Server Managment Studio

Creación de la base de datos para inventario



Creación tabla para datos de inventario

```
USE InventoryDB;  
  
CREATE TABLE PokeInventory (  
    PokemonID INT PRIMARY KEY,  
    Name NVARCHAR(50),  
    Stock INT  
);
```

Inserción y visualización de datos en la tabla

```
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (1, 'Bulbasaur', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (2, 'Ivysaur', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (3, 'Venusaur', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (4, 'Charmander', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (5, 'Charmeleon', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (6, 'Charizard', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (7, 'Squirtle', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (8, 'Wartortle', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (9, 'Blastoise', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (10, 'Caterpie', 100);
INSERT INTO PokeInventory (PokemonID, Name, Stock) VALUES (11, 'Metapod', 100);

SELECT * FROM PokeInventory;
```

Results

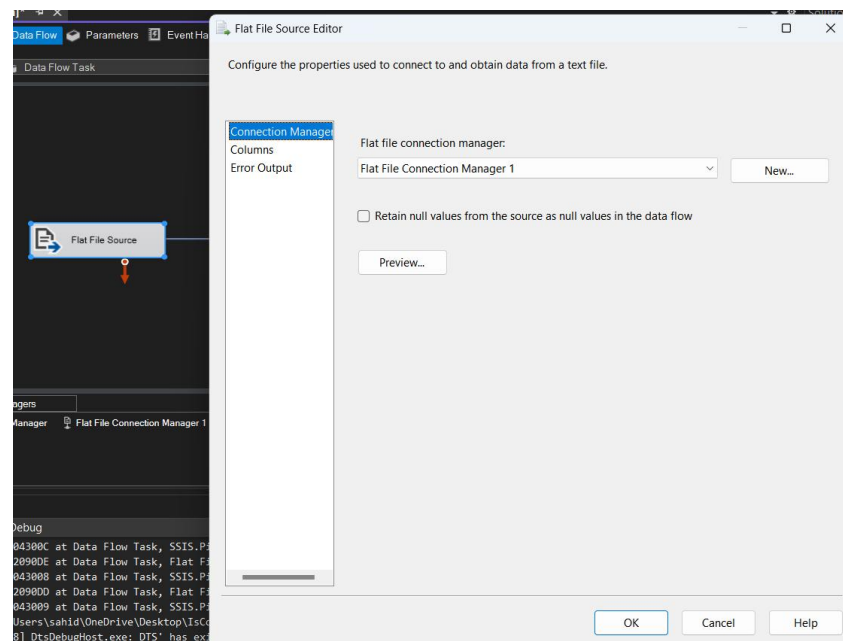
ID	Name	Stock
1	Bulbasaur	100
2	Ivysaur	100
3	Venusaur	100
4	Charmander	100
5	Charmeleon	100
6	Charizard	99
7	Squirtle	99
8	Wartortle	100
9	Blastoise	100
10	Caterpie	100
11	Metapod	100

INTEGRACION - Visual Studio Community - Integration Services Project

Creación del proyecto de integracion

The screenshot displays the Visual Studio Community interface for an SSIS (SQL Server Integration Services) project named 'Package.dtsx'. The 'Data Flow' tab is active, showing a 'Data Flow Task' with a single task named 'Data Flow Task'. The task is configured with a 'Flat File Source' connected to an 'OLE DB Command' destination. The 'Flat File Source' is connected to 'Flat File Connection Manager 1', and the 'OLE DB Command' is connected to 'LAPTOP_SAHIDSQLEXPRESS.InventoryDB'. The 'OLE DB Command' task is currently set to '100%' completion. The 'SSIS Toolbox' on the left shows various data flow tasks, with 'OLE DB Command' highlighted. The 'Connection Managers' section at the bottom lists 'Flat File Connection Manager', 'Flat File Connection Manager 1', 'LAPTOP_SAHIDSQLEXPRESS.InventoryDB', and 'Flat File Connection Manager 2'.

Lectura del archivo CSV generado para actualizar stock en inventario



Conexión a la base de datos

