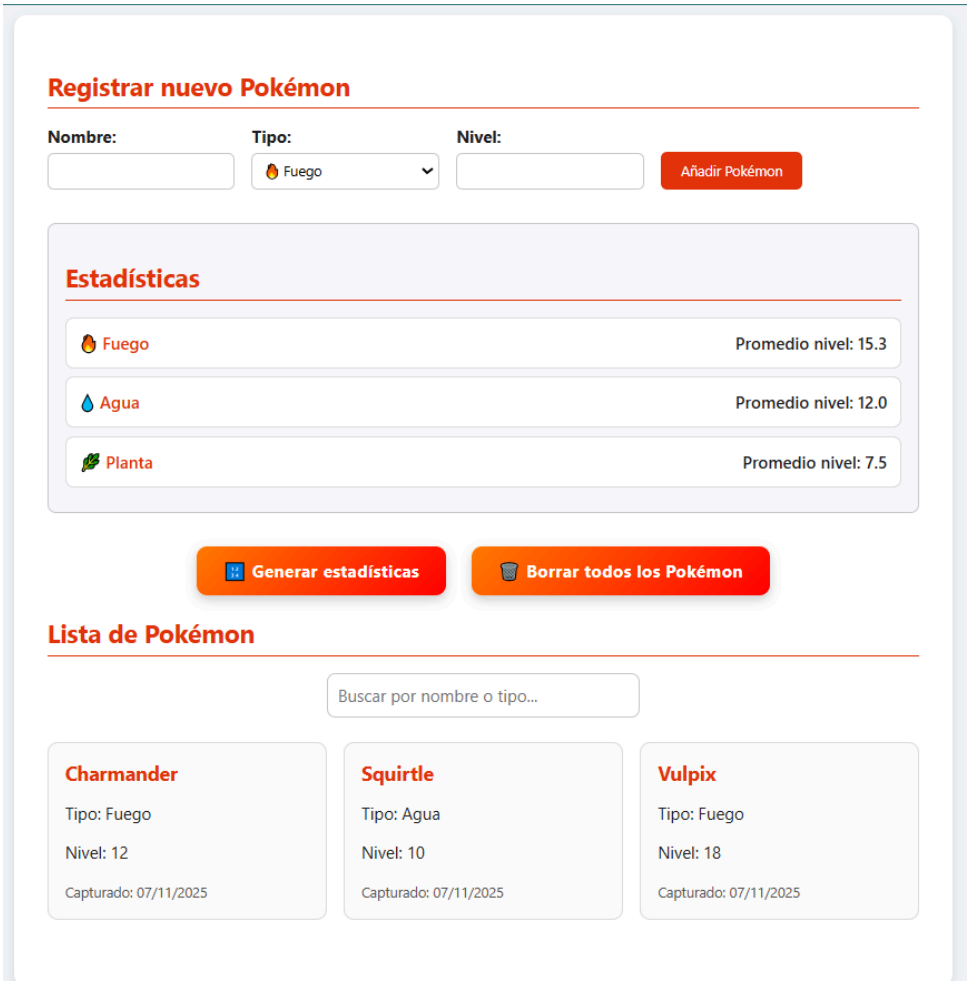
Pokédex Comercio

El IES Comercio quiere crear una pequeña aplicación web llamada **Pokédex Comercio**. La aplicación permitirá registrar Pokémon con su nombre, tipo y nivel, consultarlos desde un listado, filtrarlos mediante búsqueda y visualizar estadísticas básicas.

Tu tarea consiste en **diseñar e implementar la aplicación web** utilizando **HTML, CSS y JavaScript**, siguiendo las especificaciones que se indican a continuación.

Además, el estudiante **deberá implementar 3 patrones de diseño** en JavaScript:

- **1 Patrón Creacional**
- **1 Patrón Estructural**
- **1 Patrón de Comportamiento**



Estructura general

La web constará de **una sola página (index.html)** con tres secciones principales:

1. **Registro de Pokémon**
   Permite añadir nuevos Pokémon introduciendo su nombre, tipo y nivel.
   - Campos obligatorios:
     - Nombre (texto)
     - Tipo (selector con opciones)
     - Nivel (número entre 1 y 100)
   - Al pulsar el botón **"Añadir Pokémon"**, se agrega un nuevo Pokémon al listado general.
   - La fecha de captura se añadirá automáticamente con la fecha actual (por ejemplo: "07/11/2025").
   - Los Pokémon se guardarán en **LocalStorage**, de manera que al cerrar la pestaña no se perderán.
2. **Estadísticas (parte superior de la página)**
   Muestra el promedio de nivel de todos los Pokémon agrupados por tipo.
   - Este bloque se actualizará cuando se pulse el botón **"Generar estadísticas"**.
   - Al entrar en la página se generan con los datos que haya en el LocalStorage.
3. **Listado de Pokémon**
   - Muestra todos los Pokémon almacenados en una cuadrícula.
   - Cada tarjeta incluirá:
     - Nombre
     - Tipo
     - Nivel
     - Fecha de captura
   - En la parte superior del listado habrá un **campo de búsqueda** para filtrar Pokémon por nombre o tipo.
   - Si no hay resultados, se mostrará un mensaje del tipo "No se han encontrado Pokémon".

Funcionalidades a implementar

Tu aplicación debe cumplir los siguientes requisitos funcionales:

Añadir Pokémon

Cuando el usuario complete el formulario y pulse el botón “Añadir Pokémon”:

Se creará un objeto con la siguiente estructura:

```
{
 nombre: "Charmander",
 tipo: "Fuego",
 nivel: 12,
 fecha: "07/11/2025"
}
```

Este objeto se añadirá a un **array.** El array se guardará en **LocalStorage** para conservar los datos mientras la pestaña esté abierta. El listado de Pokémon debe actualizarse automáticamente.

Buscar Pokémon

En el campo de búsqueda (situado **encima del listado**) el usuario podrá escribir un texto. El filtrado se realizará por **nombre** o **tipo** (no sensible a mayúsculas/minúsculas).

Solo se mostrarán los Pokémon que coincidan con el criterio de búsqueda.

Generar estadísticas

El botón **“Generar estadísticas”** calculará los **promedios de nivel por tipo**.

Borrar todos los Pokémon

El botón **“Borrar todos los Pokémon”** eliminará el contenido del listado y del **LocalStorage**.

Al recargar la página, la lista debe aparecer vacía.

Requisitos de diseño

- El estilo de la web debe ser **similar al mockup entregado**, los colores no son importantes, pero sí que los elementos están organizados de la misma manera.
- No se permite el uso de frameworks CSS (Bootstrap, Tailwind, etc.).
- Se debe usar **flexbox o grid** para distribuir los elementos.

Pistas / Sugerencias

- Crea una función actualizarLista() que redibuje los Pokémon cada vez que cambie el array.
- Crea una función guardarEnSession() y otra cargarDeSession() para simplificar el guardado.
- Para calcular promedios, puedes usar reduce().
- Para la fecha de captura, puedes usar: new Date().toLocaleDateString("es-ES");

Patrones de diseño

La aplicación debe incluir **tres patrones (1 de cada tipo)**.

Singleton

Para el gestor principal de Pokémon o para el gestor de LocalStorage.

Patrón Estructural

Facade Pattern

Un módulo `DOMFacade` que centralice:

```
DOMFacade.renderPokemonList(lista)
DOMFacade.renderStats(stats)
DOMFacade.clearForm()
DOMFacade.showMessage("No se han encontrado Pokémon")
```

Oculta la manipulación directa del DOM.

Patrón de Comportamiento

Strategy Pattern

Para el filtro de Pokémon:

```
FilterStrategy.byName(lista, texto)
FilterStrategy.byType(lista, texto)
```

Observer Pattern

Para que el listado se actualice automáticamente cuando cambia el array.

Command Pattern

Para acciones como:

- Añadir Pokémon
- Filtrar
- Borrar todos
- Generar estadísticas

Estructura recomendada

```
/js
   /patterns
       Taskfactory.js
       DOMFacade.js
       FilterStrategy.js
   script.js
```

Evaluación

Criterio	Descripción	Puntuación
HTML semántico, Estilos CSS y diseño temático	Uso correcto de etiquetas, formularios, secciones. Organización de los elementos.	0.5 pts
Uso de objetos y arrays	Estructura de datos y manipulación con JS	1,5 pts
LocalStorage	Persistencia correcta durante la sesión	2 pt
Filtrado de Pokémon	Correcta implementación del buscador	2 pt
Generación de estadísticas	Implementación correcta de la función nivelPromedioPorTipo()	2 pts
Patrón creacional	Singleton	0,5pts
Patrón estructural	Facade	0,5pts
Patrón de comportamiento	Strategy / Observer / Command	1pts

Total: 10 puntos

Entrega

Un único archivo comprimido con los archivos HTML, CSS, JS.