

TP3: Directorio Alumnos (React + Vite)

Objetivo

Construir una aplicación web para mostrar el directorio de alumnos (solo lectura) que:

- Cargue los datos desde `alumnos.vcf` (vCard 3.0) incluido en el proyecto.
- Muestre por alumno: nombre completo, teléfono, legajo y, si está disponible, usuario de GitHub con avatar.
- Permita buscar por nombre, teléfono o legajo (ignorando mayúsculas/minúsculas y acentos).
- Permita marcar/desmarcar favoritos.
- Agrupe los resultados mostrando primero los favoritos y luego el resto; ambos grupos ordenados alfabéticamente por nombre normalizado.

Restricciones y Alcance

- Sin CRUD: no se puede crear, editar ni eliminar alumnos.
- Sin formularios ni modales.
- Sin persistencia: el estado de favoritos no se guarda entre recargas.

Requisitos Funcionales

1. Carga de datos

- Importar `alumnos.vcf` como texto y parsearlo en el cliente.
- Asumir un patrón fijo por tarjeta con los campos: `FN`, `TEL;TYPE=CELL`, `NOTE` (contiene `Legajo: <número> ... Github: <usuario>`).
- Generar objetos: `{ id, nombre, telefono, legajo, github, favorito }` con `favorito: false` por defecto.
- `id`: usar `legajo`.
- `github`: usar el usuario de GitHub si está presente; si no, `""`.

2. Búsqueda y orden

- Campo de búsqueda en la barra superior.
- Filtra por coincidencia en `nombre`, `telefono` o `legajo` (normalización sin acentos y sin distinción de mayúsculas/minúsculas).
- Orden alfabético por nombre normalizado dentro de cada grupo.

3. Favoritos y agrupación

- Interacción para alternar la propiedad `favorito` de un alumno en memoria.

- Listar en dos grupos: favoritos y no favoritos; ambos ordenados.
- Si no hay resultados, mostrar un mensaje informativo de lista vacía.

4. Avatar de GitHub

- Si `github` está presente, usar la imagen `https://github.com/<usuario>.png?size=100` como avatar del alumno; si no, proveer una alternativa textual (por ejemplo, iniciales).

Requisitos Técnicos

- Stack: React + Vite.
- Importar el VCF como texto (ej.: `import alumnosVcf from '../alumnos.vcf?raw'`).
- Parseo del VCF con expresiones regulares asumiendo el formato fijo de las tarjetas.
- Estado en memoria con `useState`.
- Separación en módulos:
 - Servicio de datos: `src/services/alumnos.js` con `parseVcf` y `loadAlumnos`.
 - Utilidades de texto: `src/utls/text.js` con `norm`, `cmpNombre`, `includesContacto`.

Componentes sugeridos

- `Topbar` : título y campo de búsqueda.
- `ContactSection` : recibe `title` y `contacts` y renderiza la lista.
- `ContactCard` : muestra nombre, teléfono, legajo y avatar (si hay GitHub), además de la interacción de favorito.

Referencia visual

El diseño es libre, pero se sugiere una estructura similar a la siguiente:

Alumnos Programación 4

Q Buscar por nombre, teléfono o legajo

Favoritos (3)



**Ferreyra Appas, Santiago
Nicolás**

Tel: 3816484455
Legajo: 61198



Giacobbe, Franco

Tel: 3815880371
Legajo: 61167



**Jiménez Paz, Patricio
Agustín**

Tel: 3813882674
Legajo: 61450

Contactos (61)



**Acevedo Costello, Juan
Ignacio**

Tel: 3813887804
Legajo: 61203



Acosta, Maira

Tel: 3815628162
Legajo: 61667



**Agostino Colombres,
Juan Manuel**

Tel: 3816848399
Legajo: 61131



**Ahumada, Aiquen
Osvaldo**



Amoedo, Pablo Isaías



Carabajal, José Gabriel

Entrega

[!IMPORTANT]

El trabajo se deben entregar hasta el **lunes 15 de septiembre a las 21:00 hs.**

Como realizar el trabajo.

Recuerde

1. Pasar a la rama main de repositorio, actualizar el contenido.
2. Crear su propia rama resolver y publicar la solución.
3. Realizar un Pull Request a la rama main del repositorio original.

Nota: Recuerde que el titulo del PR debe ser: "TP3 - - <Apellido, Nombre>"