

Módulo 3: Ejercicio de evaluación intermedia

El ejercicio consiste en desarrollar una página web "sencilla" con React.

Antes de empezar, tenéis que crear un nuevo repositorio en GitHub desde GitHub Classroom usando [este enlace](#). Una vez creado, lo clonaremos en nuestro ordenador y en la carpeta creada empezaremos a trabajar en el ejercicio.

Listado de Adalabers

Vamos a llevar la cuenta de las compañeras que somos en esta promo y a apuntar la especialidad de cada una. Queremos crear una aplicación para gestionar alumnas de Adalab y.. ¡¡¡lo vamos a hacer con React!!!

The screenshot shows a web browser window with the title 'Adalabers list' and the address bar showing 'localhost:3001'. The page content includes a header 'Adalabers', a form to add new members, and a table of existing members.

Form to add new members:

Nombre:

Escoge una tutora:

Table of existing members:

Nombre	Tutora	Especialidad	Redes
MariCarmen	Yanelis	Python	
Amparo	Dayana	IA	
Escandia	Iván	3D graphics	

Form to add new members (bottom):

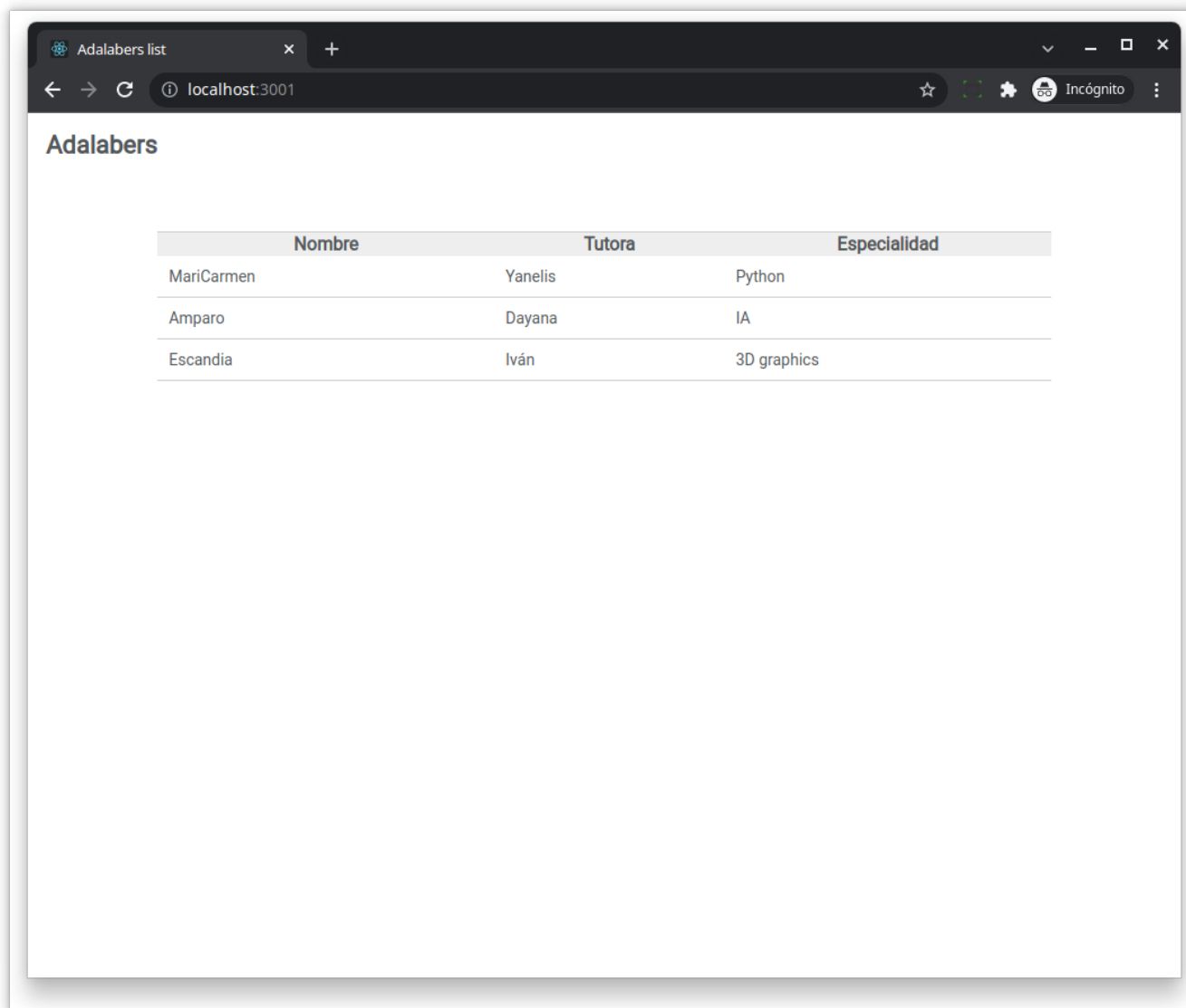
Nombre:

Tutora:

Especialidad:

A continuación detallamos lo que queremos hacer.

1º Pintar el listado de Adalabers



Para empezar el ejercicio queremos:

1. Pintar una cabecera con un título que ponga **Adalabers**.
2. Pintar el listado de Adalabers inicial. Para ello os facilitamos los datos en la siguiente API:
<https://beta.adalab.es/curso-intensivo-fullstack-recursos/apis/adalabers-v1/promo-radsajsd.json>

Os recomendamos que primero copiéis el contenido del API y los peguéis en un fichero `json` en vuestro proyecto. Más adelante os encargaráis del `fetch()`. De momento lo importáis en vuestro componente App y lo guardáis en el estado.

Después, generar el HTML de la página con los datos que habéis importado del json. Recordad que [las etiquetas para hacer tablas](#) en HTML son:

```
<table className="table">
  <!-- Fila de cabecera -->
  <thead><tr>
    <th>Nombre</th>
    <th>Tutora</th>
    <th>Especialidad</th>
  </tr></thead>
```

```
<!-- Fin fila de cabecera -->

<tbody>
  <!-- Primera fila -->
  <tr>
    <td>MariCarmen</td>
    <td>Yanelis</td>
    <td>Python</td>
  </tr>

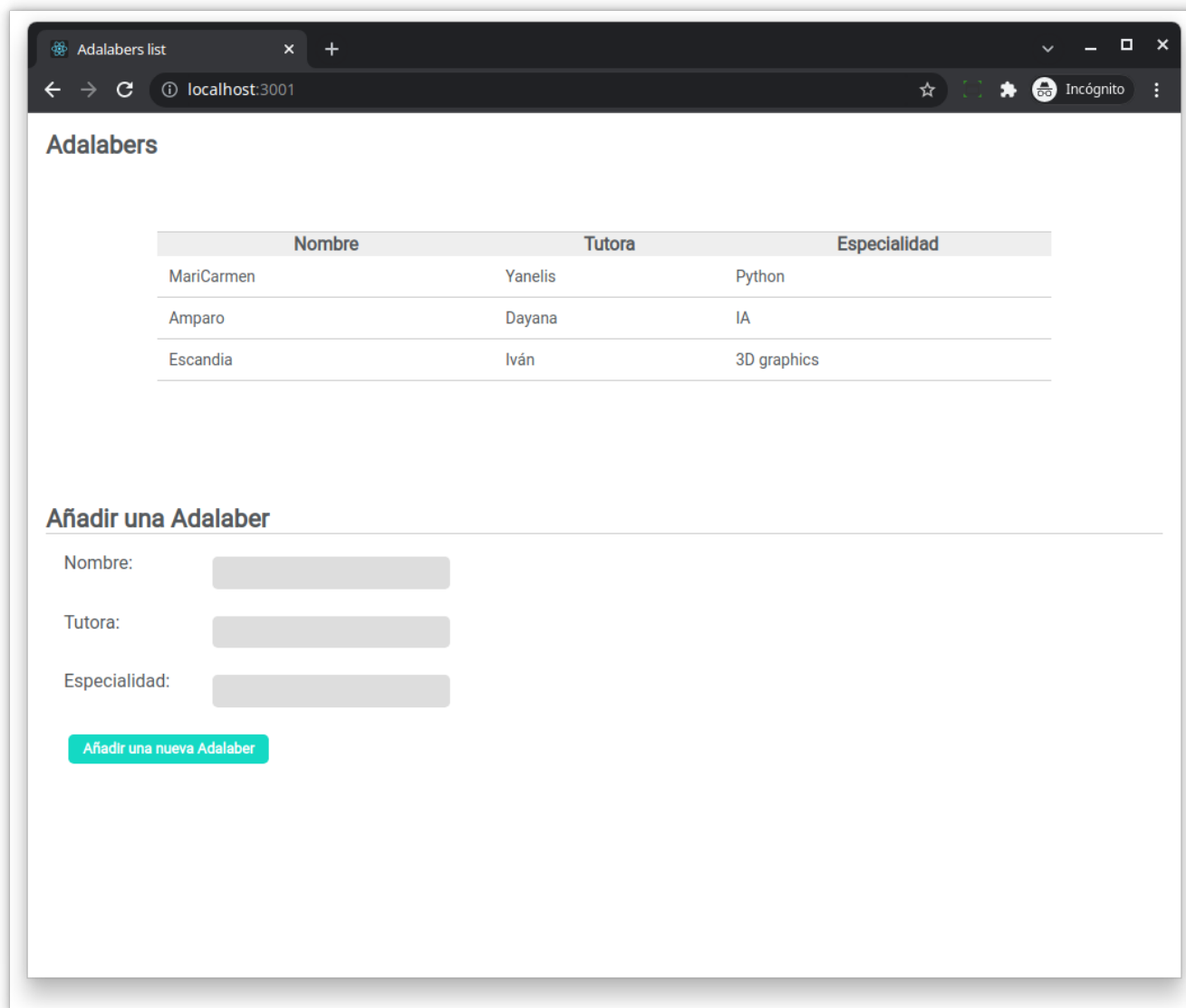
  <!-- Segunda fila -->
  <tr>
    <td>Amparo</td>
    <td>Dayana</td>
    <td>IA</td>
  </tr>

  <!-- Tercera fila -->
  <tr>
    <td>Escandia</td>
    <td>Iván</td>
    <td>3D graphics</td>
  </tr>
</tbody>
</table>
```

Los elementos `<table>`, `<thead>` y `<tbody>` son fijos. La parte variable de la tabla son los `<tr>` que se crean por cada elemento del array de datos dentro de la etiqueta `<tbody>` de la tabla. Esos `<tr>` los debemos generar desde React.

Y cuando se vean todos los datos, nos ocuparemos del fetch para obtener los datos directamente del API y no tenerlos "grabados a fuego" en el código fuente de nuestra aplicación.

2º Añadir a una nueva Adalaber



A continuación, queremos añadir los datos de una compañera nueva. Para ello seguiremos los siguientes pasos:

1. Crear un formulario con los campos:
 - Nombre de la compañera
 - Nombre de su tutora
 - Especialidad en la que destaca
 - Un botón para **Añadir una nueva Adalaber**.
2. Cuando la usuaria pulse en el botón, hay que añadir la Adalaber al listado de alumnas para que ésta aparezca en la tabla.

Te queremos dar unas pistas en forma de preguntas. Antes de que la usuaria pulse en el botón de "Añadir una nueva Adalaber" debes pensar:

- ¿Vas a guardar los datos del formulario en algún sitio?
- ¿En qué **tipo** de datos vas a guardar los valores de los `<input>`?
- ¿Qué **valores iniciales** tienen estos datos?
- ¿Qué **eventos** tendrá la página? ¿Qué hay que cambiar en las variables estado?

3º Filtrar el listado de Adalabers (BONUS)

Adalabers list

localhost:3001

Adalabers

Nombre:

Escoge una tutora:

Nombre	Tutora	Especialidad
MariCarmen	Yanelis	Python
Amparo	Dayana	IA
Escandia	Iván	3D graphics

Añadir una Adalaber

Nombre:

Tutora:

Especialidad:

A continuación queremos filtrar las alumnas por nombre y por tutora. Intenta hacer uno de los dos y, cuando lo tengas controlado, implementa el restante (no hace falta que estén los dos filtros).

Para filtrar por nombre:

1. Añade un recuadro sobre la tabla del listado de alumnas donde la usuaria pueda escribir texto.
2. Añade la funcionalidad para que, cuando la usuaria escriba en ese `<input>`, el listado de Adalabers se re-pinte mostrando solo las que su nombre incluya el texto que ha escrito la usuaria (aplicando también el filtrado por tutora, si ya lo tuvieras implementado).

Para filtrar por tutora:

1. Añade un `<select>` a la cabecera que tenga las siguientes opciones:
 - Cualquiera (es la opción por defecto)
 - Yanelis
 - Dayana
 - Iván
2. Añade la funcionalidad para que, cuando la usuaria cambie ese `<select>`, el listado de Adalabers se re-pinte mostrando solo las que coincidan con la opción seleccionada (aplicando también el filtrado por nombre, si procede).

Más pistas, esta vez sobre ese `<input>` y ese `<select>`:

- ¿Vas a guardar los datos del select en algún sitio? ¿y del input?
- ¿Qué **tipo** de datos vas a guardar?
- ¿Qué **valor inicial** le vas a poner?

Y siempre que filtramos algo debemos hacernos las siguientes preguntas:

- ¿Qué datos de cada Adalaber debo usar en cada filtro?
- ¿Qué datos de cada Adalaber tienes que comparar con la opción seleccionada por la usuaria para saber si una Adalaber cumple o no con el filtro?
- ¿Hay alguna opción del select con la que deba mostrar todas las Adalabers?
- ¿Hay alguna opción del select con la que no deba mostrar ninguna Adalaber?
- ¿Debo guardar las Adalabers filtradas en algún sitio o puedo calcularlas con JavaScript antes de ponerlas en el HTML?

4º Mostrar redes sociales (BONUS)

Nombre	Tutora	Especialidad	Redes
MariCarmen	Yanelis	Python	GitHub LinkedIn Twitter
Amparo	Dayana	IA	GitHub
Escandia	Iván	3D graphics	GitHub LinkedIn

Añadir una Adalaber

Nombre:

Tutora:

Especialidad:

[Añadir una nueva Adalaber](#)

Después de pintar el listado, podemos intentar pintar una columna más en nuestra tabla con enlaces a las redes sociales que tiene cada Adalaber. Puedes guiarte con los siguientes pasos:

1. Añade una columna más en la cabecera de la tabla.
2. Añade una celda más a cada fila.
3. El contenido de la celda extra de cada fila serán los nombres de las redes sociales de cada Adalaber. Puede ser que no tenga ninguna o puede ser que tenga hasta 3.
4. Añade un enlace `<a>` al nombre de la red social que abra la URL (o dirección) de la misma.

Y ya que estamos también te damos unas pistas:

- ¿Cómo están almacenados los datos de las redes sociales de cada Adalaber?
- ¿Cómo podemos recorrer todos los datos de las redes sociales para generar su HTML? ¿Hemos hecho algo parecido en este proyecto antes?
- ¿Dónde debo incluir ese código que genera el HTML de la lista de redes sociales? Recuerda que este código hay que usarlo para cada Adalaber.

5º Obtén los datos desde el API con un fetch (BONUS)

Implementa la funcionalidad de traer los datos desde el servidor cuando se cargue la página. Sólo debe pedir los datos al cargar/refrescar la página y no cuando creamos una nueva adalaber o cuando usamos algún filtro.

6º Maqueta a tu gusto (BONUS)

No te vamos a decir cómo debes poner bonita tu web porque tienes más estilo y flow que nosotras.

Entrega

En este enunciado **os proponemos dos ejercicios básicos** y cuatro ejercicios avanzados marcados como **BONUS**. Hemos calculado que hacer los 6 ejercicios durante el tiempo de la evaluación es científicamente imposible, aspiramos a que hagas solo los dos primeros ejercicios y el resto se queden para practicar los próximos días sobre un ejercicio que ya conoces.

Cuando te dispongas a continuar con los bonus (o completar el ejercicio con lo que se te ocurra) después del tiempo de evaluación, puedes hacerlo pero en otra rama diferente a master o main, ya que no debes hacer cambios en la rama principal de tu repo hasta que tu profe no termine de corregir tu evaluación.

La evaluación solo se considerará terminada cuando:

- **Esté publicada en GitHub Pages** (apuntando a la rama main / master) y esté funcionando.
- El enlace a GitHub Pages esté en la página principal del repositorio, en la parte superior, al lado de la descripción.
- Debéis hacer commits y merges a la rama main / master solo hasta las **11:10**. No hagáis commit a la rama main / master después de esa hora.
- **Recuerda** que GitHub Pages a veces tarda unos minutos en actualizar tu página.

¡A por el Ferrari!