

## Especialización en Front End II

# Examen final

### Trabajamos sobre una aplicación de Los Simpsons

Llegó el momento de poner a prueba los conocimientos adquiridos. Es una oportunidad para validar tu aprendizaje a lo largo de esta cursada. Se trata de una evaluación individual entregable en la cual tendrás la posibilidad de resolver el trabajo por cuenta propia y entregarlo antes de la fecha límite: la **clase 24**.

## Consignas

En esta oportunidad, te proponemos trabajar sobre una aplicación web de Los Simpsons. Dicha aplicación tendrá tres secciones con funcionalidades diferentes, sobre las que deberás resolver distintas tareas aplicando los conocimientos adquiridos durante la cursada. A continuación, detallamos el comportamiento que esperamos obtener una vez finalizado el desarrollo.

## Checklist de evaluación

A continuación, detallamos el listado de requerimientos que serán tenidos en cuenta al momento de evaluar el proyecto.

## Condiciones mínimas de aprobación

Las siguientes condiciones deben cumplirse satisfactoriamente para que un proyecto pueda considerarse como aprobado:

- Cumplir con las tareas que sean indicadas como obligatorias en las consignas.
- El proyecto deberá realizarse sobre la base del template entregado. No se aceptarán proyectos que se hayan realizado sin respetar dicho requerimiento.



- Solo se podrán utilizar las librerías que se detallan en este documento y/o el archivo README que se encuentra dentro del repositorio.
- Cualquier funcionalidad que sea implementada utilizando una librería distinta a las permitidas, no se considerará realizada.
- El proyecto deberá ser desarrollado utilizando TypeScript como lenguaje.
- Deberá contar con tests unitarios escritos para la primera sección, tal como se indica en la consigna número 1. Para ello deberá emplearse Jest y React Testing Library. Además, se deberá utilizar MSW o alguna otra librería para interceptar los requests y mockear una respuesta.
- Se espera que el proyecto cuente con un coverage de al menos 50% como condición mínima de aprobación.
- En los casos en que se requiera dar estilos a componentes, deberá realizarse utilizando la librería Styled Components. Los estilos dinámicos también deberán manejarse mediante esta librería.
- Se espera que al menos puedan aplicar 1 principio SOLID para resolver la primera consigna, explicando en un comentario en qué parte del código podemos ver dicha aplicación.

## Aspectos a tener en cuenta para la valoración final de la nota

Los siguientes aspectos son **extras**, que si bien no constituyen requisitos mínimos para aprobar el trabajo, serán tenidos en cuenta al momento de determinar la nota final de cada estudiante, siempre y cuando su implementación sea correcta:

- **TypeScript**
  - En el caso de que se deba llevar adelante una refactorización del código (consigna número 3), será tenido en cuenta el uso de TypeScript para el tipado de los componentes y demás funciones que desarrollen lógica reutilizable.



- Se valorará la reutilización de tipos comunes que se repiten a lo largo del código, especialmente mediante la reutilización de interfaces.
- **Documentación**
  - Se valorará el correcto uso de la documentación en todas las funciones y componentes en caso de ser necesario.
- **Validaciones**
  - Se valorará el agregado de validaciones de flujos alternativos al normal y el manejo de errores en las distintas funcionalidades implementadas.
- **Testing unitario y coverage**
  - Se valorará el correcto uso del testing unitario y el incremento del porcentaje de cobertura de código (coverage) más allá del 50%.
- **Buenas prácticas**
  - Se prestará especial atención al uso de buenas prácticas, principios SOLID, reutilización de componentes y funcionalidades comunes, y renderizado dinámico.

## Funcionalidades

### Funcionalidades obligatorias

Las siguientes funcionalidades son **requisitos mínimos necesarios** para la aprobación del final y es fundamental que funcionen correctamente.

### Sección 1: Citas de Los Simpsons

En esta sección, el usuario cuenta con la posibilidad de obtener citas de Los Simpsons. Hay dos maneras:

- Primero, haciendo clic en el botón correspondiente sin ingresar ningún nombre en el input. Esto devuelve una cita al azar.



- Segundo, si se ingresa el nombre de un personaje antes de presionar el botón, la API devuelve una cita de ese personaje.
- Por otra parte, si se ingresa un valor numérico, se muestra un mensaje de error.

La funcionalidad está implementada con Redux Toolkit usando `thunk` para el request a la API.

Deberás escribir los tests de esta sección. Para ello deberás:

- Crear un custom render agregando el provider de Redux para poder disparar las acciones.
- Utilizar MSW o alguna otra librería para interceptar los requests y mockear una respuesta. No se permite mockear el método `fetch`.
- Desarrollar tests de integración sobre el componente “Quotes”, evaluando los distintos test cases que contemplen los distintos flujos de comportamiento.

El código base de esta sección se encuentra dentro de **src/features/quotes**. Allí encontrarás el componente y demás utilidades a tener en cuenta para escribir los tests.

## Sección 2: Bio

En esta parte se presenta una sección que tiene una botonera que permite alternar entre la biografía de cada personaje. En este caso, lo que pretendemos es maquetar la sección usando **Styled Components**.

Además, para el caso de los botones, se requiere aplicar estilos basados en props para poder resaltar el botón del personaje cuya bio se está mostrando en la pantalla.

El código base para trabajar en esta sección se encuentra en **src/features/bio**. El componente se encuentra funcionando según lo esperado, por lo que es importante verificar que dicha funcionalidad no haya sido afectada luego de la refactorización.

## Sección 3: Noticias

Esta sección presenta un listado de noticias acerca de Los Simpsons. Cada noticia tiene una prop **esPremium** que determina si el usuario puede acceder a la misma o no.

- Si puede, al hacer clic en el botón “Ver más” se abre un modal con el detalle completo de la noticia.



- Si no, se abre un modal que invita al usuario a suscribirse.

El código base de esta sección se encuentra dentro de la carpeta **src/features/news**. Allí encontrarás el archivo **Noticias.jsx**, dentro del cual puede verse que hay múltiples componentes y funciones que realizan tareas sobre la información. En este caso, el requerimiento consiste en refactorizar el código aplicando los principios y buenas prácticas que hemos visto a lo largo de la cursada. En especial, se espera que puedas aplicar los principios SOLID durante el proceso de refactorización. A tal fin, está permitido crear nuevos archivos dentro de la carpeta “news”, en caso de que lo consideres necesario para extraer cierta lógica del código. Sin embargo, debe tenerse en cuenta que solo debe trabajarse sobre el archivo **Noticias.jsx**. No deberá modificarse el contenido de los archivos **fakeRest.ts** y **styled.ts**, ya que ello no forma parte de la consigna.

**¿Listos para comenzar el desafío? ¡Adelante!**

## ¡Iniciamos!

Es indispensable realizar un **fork** de este proyecto para poder trabajar de forma individual. Luego, deberás clonar el repositorio que se encuentra en el siguiente [link](#). Allí encontrarás el setup necesario para poder comenzar a trabajar. No se aceptarán proyectos que se hayan realizado sin partir de este template.

Es fundamental que **leas atentamente todas las consignas antes de comenzar a desarrollar tus soluciones**, ya que cada una de ellas representa una porción del puntaje que podrás obtener en la evaluación de tu trabajo. Además, mediante esta primera lectura tendrás una idea general del resultado buscado, lo cual es clave para planificar el desarrollo del proyecto y cumplir con los requerimientos.



## Paso a paso

Ahora que ya vimos cuáles son los requerimientos y funcionalidades que deberás realizar en este proyecto, haremos un recorrido por los principales pasos a seguir para poder completar el desarrollo de la aplicación.

En el archivo README, se detallan todos los pasos necesarios para que puedas llevar a cabo este proyecto.

## ¡A programar!

Una vez cumplidos los pasos anteriores, tu aplicación debería funcionar conforme lo explicado en esta consigna. ¡Felicitaciones!