

Especialización en Front End I

Examen final

Desarrollo de una aplicación de Rick and Morty

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

Consignas

En esta oportunidad, les proponemos crear una aplicación web de Rick and Morty. Dicha aplicación tendrá dos páginas con funcionalidades diferentes, pero que al mismo tiempo se van a relacionar entre sí. 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 todas las funcionalidades obligatorias.



- 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 el README que se encuentra dentro del repositorio.
- No se considerará realizada cualquier funcionalidad que sea implementada utilizando una librería distinta a las permitidas.
- Deberá ser desarrollado utilizando Typescript como lenguaje. Como mínimo, se espera que se utilice el tipado de props en los componentes de React.
- Toda la información de los elementos deberá provenir de la API de Rick and Morty (<https://rickandmortyapi.com>). No se admiten datos duros (hardcodeados) en el front end. Para ello, se deberá implementar la lógica para manejar el pedido a dicha API.
- Deberá ser desarrollado utilizando **Redux** junto con Thunk o Saga. Se otorga la posibilidad de elegir entre una de las dos. Como mínimo se espera el uso correcto de un Reducer, más el uso de una **ThunkAction** (Thunk) o **función generadora** (Saga).
- Se deberá documentar correctamente al menos siete funciones (pueden o no ser componentes de React).

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**
 - Será tenido en cuenta el uso de Typescript más allá del tipado de las props de componentes. En especial para el caso de funciones que desarrollen lógica reutilizable, de las Actions y de los Reducers.
 - Se valorará la reutilización de tipos comunes que se repiten a lo largo del



proyecto, especialmente mediante la reutilización de interfaces.

- **Documentación**

- Se valorará el correcto uso de la documentación en todas las funciones y componentes.

- **Validaciones**

- Se valorará el agregado de validaciones de flujos alternativos al normal y el manejo de errores en las distintas funcionalidades implementadas.

- **Redux**

- Se valorará el uso correcto de Redux para el almacenamiento del estado completo de la aplicación.

- **Thunk o Saga**

- Se valorará el uso correcto de Thunk o Saga para la utilización de cada función asincrónica dentro de cada funcionalidad.

- **Buenas prácticas**

- Se prestará especial atención al uso de buenas prácticas, 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.

La aplicación deberá contar con dos páginas:

- La página de inicio, que debe contar con:
 - Panel de filtros:



- Contendrá un input en donde el usuario puede ingresar cualquier texto que desee para hacer un filtro sobre los personajes.
- Si el usuario borra el texto ingresado, se deberá mostrar el estado inicial. Es decir, la búsqueda sin filtro.
- Se deberá incluir un botón de "Limpiar filtros", el cual deberá borrar el texto dentro del input y mostrar el estado inicial. Es decir, la búsqueda sin filtro.
- Listado de personajes
 - Deberá mostrar los personajes, que estarán representados por tarjetas con información sobre cada uno de ellos.
 - Como máximo se deben mostrar en pantalla nueve tarjetas de personajes por página.
 - Se deberá poder marcar y desmarcar a cada personaje como favorito desde su tarjeta individual, teniendo que persistir ese estado en la aplicación.
 - Se deberá poder diferenciar si un personaje se encuentra marcado como favorito o no.
- La página de favoritos, que deberá cumplir con los siguientes requisitos:
 - Todos los personajes marcados como favoritos deberán aparecer dentro del listado de favoritos en la segunda página de la aplicación.
 - Si el usuario desmarca un personaje como favorito, el mismo deberá desaparecer del listado de favoritos.
 - Un personaje que haya sido desmarcado de los favoritos, podrá ser marcado nuevamente desde la pantalla de inicio y deberá aparecer en el listado otra vez.

Funcionalidades extras

Los siguientes aspectos son **extras** al requisito mínimo de aprobación que serán tenidos en



cuenta para la nota final, siempre y cuando su funcionamiento sea correcto:

- La página de favoritos:
 - Podrá contar con un botón de "Eliminar todos", que deberá desmarcar todos los personajes del listado de favoritos y estos no deberán aparecer marcados como favoritos en la página principal.
- Realización de una tercera página de vista de cada personaje. Al hacer clic sobre un personaje deberá redirigirnos a esta página.
 - Va a requerir de la invocación de la API de Episodios (<https://rickandmortyapi.com/documentation/#get-multiple-episodes>) de Rick and Morty.
 - Dentro de la vista de cada personaje:
 - Se mostrará un listado de los episodios en los cuales aparece.
 - También podríamos indicar si el personaje está marcado como favorito o no, junto con la posibilidad de marcarlo o desmarcarlo.

Para despejar sus dudas respecto al funcionamiento de la aplicación, dejamos este [video](#) con una pequeña demostración de la aplicación una vez terminada.

¿Listos para comenzar el desafío? ¡Adelante!

¡Iniciamos!

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

Es fundamental que **lean atentamente todas las consignas antes de comenzar a desarrollar sus soluciones**, ya que cada una de ellas representa una porción del puntaje que podrán obtener en la evaluación de su trabajo. Además, mediante esta primera lectura tendrán 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 deberán llevar a cabo en este proyecto, haremos un recorrido por los principales pasos a seguir para poder completar el desarrollo de la aplicación.

En el archivo README que se encuentra en [este link](#) se detallan todos los pasos necesarios para que puedan llevar a cabo este proyecto.

¡A programar!

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