

ZARAWEBCHALLENGE

INTRODUCCIÓN/RESUMEN

Esta prueba consiste en la creación de una pequeña aplicación para obtener información sobre diferentes personajes de Marvel.

VISTAS

La aplicación deberá contener dos vistas:

1. Vista principal

- Mostrará un listado de 50 personajes o el resultado de los personajes introducidos en el buscador.
- Al hacer clic en el icono superior de favoritos, se deberán mostrar los personajes favoritos almacenados.

2. Detalle de personaje

- Mostrará información relativa al personaje y los cómics en los que aparece.

DISEÑO

El diseño de las vistas deberá ser responsive y ceñirse a los diseños definidos en Figma:

- [figma con los diseños](#)

MODO DESARROLLO Y MODO PRODUCCIÓN

La aplicación deberá tener un modo desarrollo en el que se sirvan los assets sin minimizar (pueden estar concatenados) y otro modo producción donde se deben servir los assets concatenados y minimizados.

PRESENTACIÓN

El objetivo final de la prueba es presentar un repositorio de código público (Github o Bitbucket) con la solución desarrollada.

En el repositorio deberá existir un archivo nombrado README donde se explicará cómo ejecutar la aplicación, la arquitectura y estructura de esta, así como toda la información relevante acerca del proyecto.

DOCUMENTACIÓN Y UTILIDADES

API-REST

- Las peticiones se tienen que realizar a la siguiente URL: <http://gateway.marvel.com/v1/>.
- La documentación se puede consultar en este [enlace](#).
- Los diseños e iconos/logotipos se podrán consultar y descargar desde el siguiente [enlace](#).

DESCRIPCIÓN DE LAS VISTAS

Vista 1 - Listado de personajes

- Desarrolla una interfaz que siga el diseño propuesto en Figma.
- La búsqueda de personajes debe apoyarse en el filtrado de la API.

Funcionalidad

- Inicialmente la página debe mostrar los primeros 50 personajes.
- La vista debe contener:
 - o Un icono con el logotipo. o Un icono que mostrará el número de personajes favoritos.
 - o Una barra de búsqueda en la que se podrán buscar personajes por su nombre. (Ejemplo: Si se busca por “Spider” se deberían mostrar todos los nombres que contengan dicha palabra).
 - o Un contador con los resultados obtenidos tras cada búsqueda que se actualizará en tiempo real.
 - o Un listado de resultados, cada uno de los cuales contendrá la imagen, nombre del personaje y una opción para añadir el personaje a favoritos.
- Al hacer clic en un resultado, se deberá de redirigir a la vista de detalle del personaje.
- Al hacer clic en el icono de favoritos en cada resultado, deberá cambiar de color y se deberá añadir una unidad al contador de la zona superior, como se indica en los diseños. Debe de existir la posibilidad de eliminar los personajes favoritos, modificando el contador. La información de favoritos debe persistir entre las diferentes vistas.

Filtro de favoritos

- Al hacer clic en el icono superior de favoritos, se deberán mostrar únicamente las tarjetas con los personajes favoritos almacenados por el usuario.
- Para volver al listado de personajes de nuevo, se deberá hacer clic en el logotipo Marvel situado en la barra de navegación.

VISTA 2. - Detalles de personaje

- Desarrolla una interfaz que siga el diseño propuesto en Figma.

Funcionalidad

La vista debe de contener:

- Un icono con el logotipo. Al pulsar en dicho enlace se redirigirá a la vista principal (Página de listado de personajes).
- Un icono que mostrará el número de personajes favoritos. Al pulsar en dicho icono se redirigirá a la vista principal, mostrando el listado de personajes favoritos almacenados.
- La imagen, título y descripción del personaje, así como la opción de añadir al personaje como favorito.
- Un listado de los cómics del personaje ordenados por fecha de salida.
- Solo se deberán mostrar los primeros 20 cómics de cada personaje.

STACK

- React \geq 17.
- Node \geq 18.
- CSS, SASS o StyledComponents.
- No usar librerías de componentes como antd, reactstrap, materialui etc...
 - o Los componentes deben ser creados desde cero por ti mismo.
- Para la gestión de estado, usar ContextAPI de React.

REQUISITOS:

- Implementación de testing
- La aplicación tiene que ser responsive.
- Correcta accesibilidad.
- Uso de *Linters* y *Formatters*.
- La consola del navegador debe de estar limpia de errores y advertencias.
- README donde se explicará cómo ejecutar la aplicación, la arquitectura y estructura, así como toda la información relevante acerca del proyecto.

Opcional:

- La aplicación puede estar desplegada.
- Uso de SSR (Es posible usar Next.js).
- Se valorará el uso de variables CSS.

CONTACTO:

Si durante la realización de la prueba encuentras alguna duda o necesitas aclaraciones, no dudes en contactarnos en el siguiente correo electrónico: **ddfrendzara.com@inditex.com**.