

# Documento de diseño desafío Houm

Por: Julian David Mendoza Ruiz. CC. 1032499557

## Requerimientos:

- Usar API pública que permita obtener lista de recursos, donde cada recurso tenga varios atributos variados.
- Presentar los recursos en formato *hero cards*.
- La API debe permitir filtrado, ordenación y/o paginación de los recursos.
- La aplicación debe tener la paleta de colores de Houm.
- Se deben visualizar al menos 10 recursos simultáneamente.
- Debe haber suficientes recursos para utilizar paginación o infinite scroll.
- Deben implementarse filtros que agreguen valor a la app.
- La página debe ser *responsive*.
- Puede utilizar cualquier librería de componentes de diseño, pero solo puede usarse una.
- La aplicación debe estar desplegada.

## Decisiones de diseño:

### API:

La API elegida para realizar el reto es la API de desarrollador oficial de Marvel. Para poder acceder a ella solo se requiere registrarse y con eso se obtiene la llave pública y privada con la cuál se crea el hash de autenticación en cada petición. Esta API tiene una gran variedad de información, pero se decidió solamente utilizar la parte de obtener personajes, ya que se presta para mostrar mucha más información que otras peticiones y es fácil de parametrizar.

Entre los parámetros hay varios filtros, ordenación, búsqueda exacta y ajustes de la cantidad de recursos enviados por petición, lo cual se ajusta a los requerimientos.

### Librería de componentes de diseño:

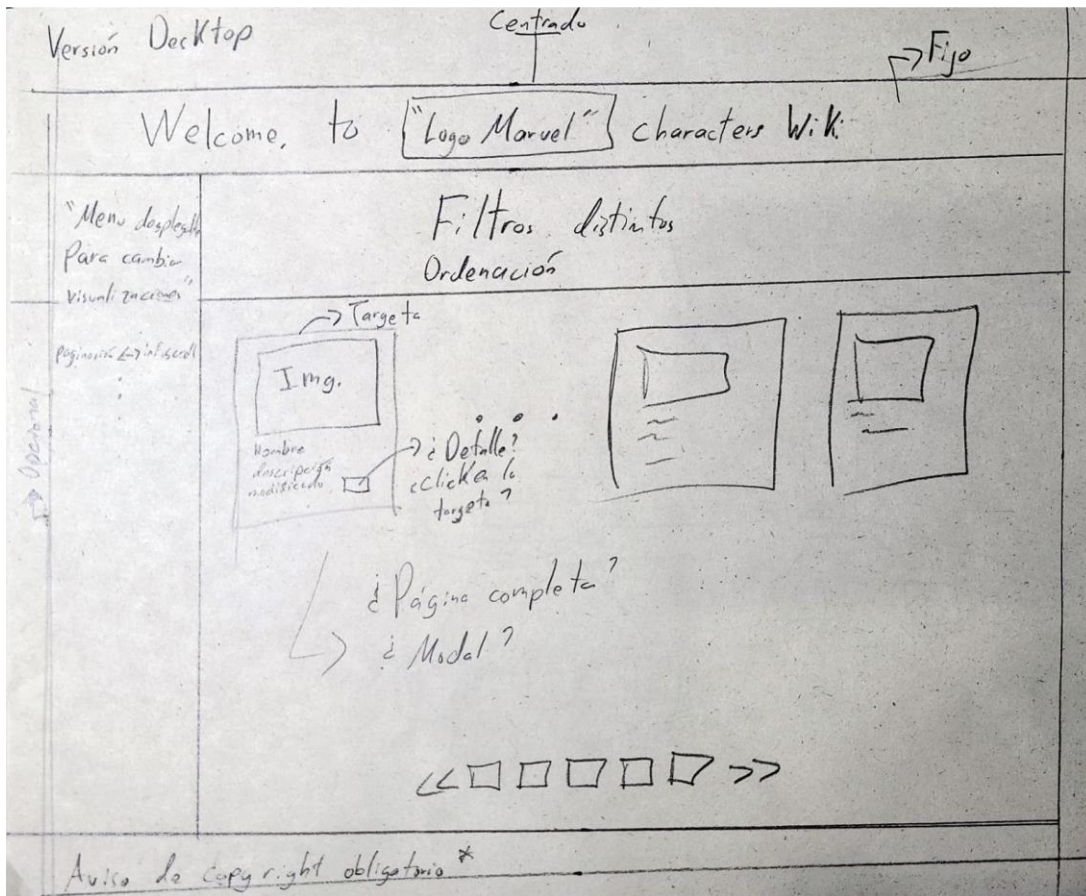
Se decidió utilizar la librería de componentes Material UI. Esta decisión se basó en dos factores principales:

1. En la página principal de Houm mediante un análisis usando las herramientas de desarrollador del navegador se descubrió que es la librería utilizada por la empresa. Tener un buen uso de la librería sería útil para reducir los tiempos de adecuación.

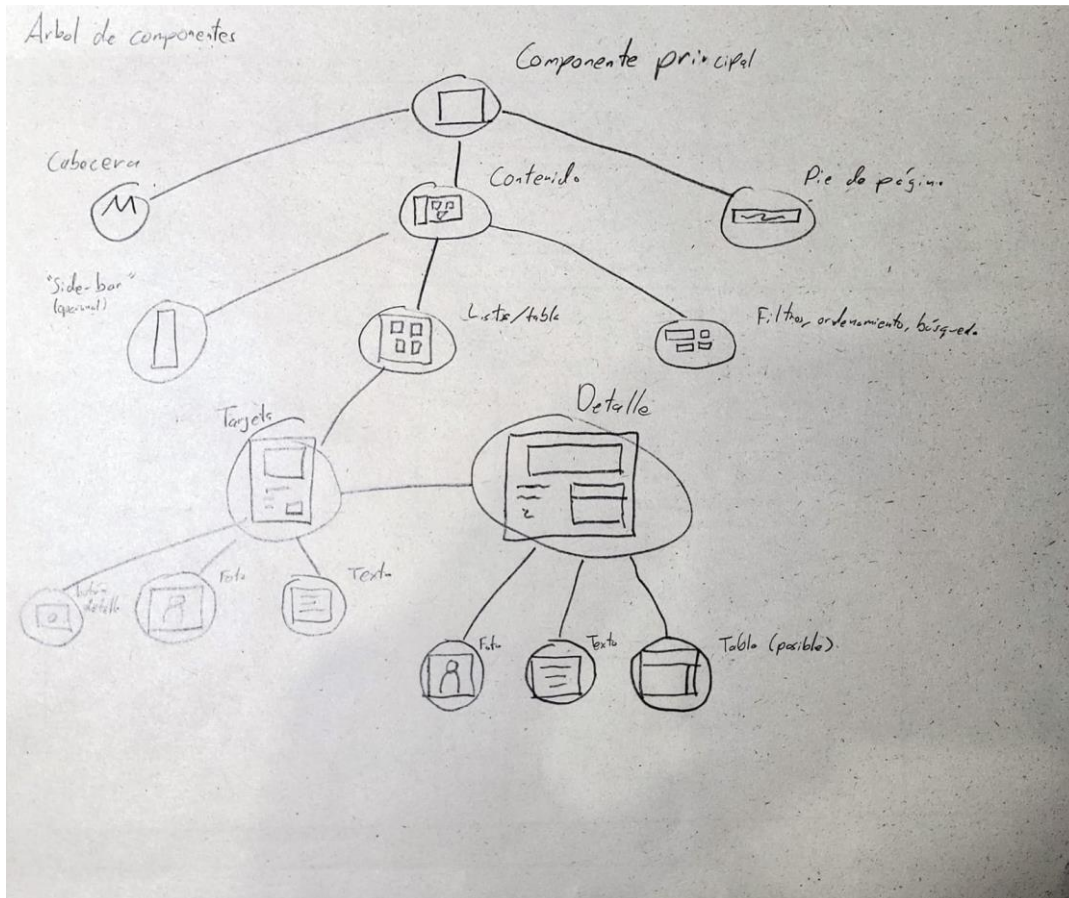
2. Material UI a diferencia de otras librerías es flexible en su configuración desde el inicio. Esto significa que cualquier componente que es agregado no tiene estilos por defecto y queda a juicio del desarrollador como configurarlo.  
En otras librerías como Primefaces los componentes vienen estilizados desde el inicio lo cual facilita el prototipado rápido pero genera retrasos al estilizar cuando se trata de sobrescribir los estilos por defecto.

## Diseño inicial de la aplicación:

Antes de empezar el desarrollo de la aplicación era necesario realizar un boceto o prototipo que pudiera representar el cómo se debe ver la aplicación. Esto debido a que es más fácil visualizar la organización de los distintos componentes y el árbol de componentes que eventualmente reflejaran decisiones de arquitectura.



En este boceto a mano se puede ver de forma básica la organización de los distintos componentes que hay en la aplicación. A su vez se utilizó el boceto para proponer algunas funcionalidades extra que podrían añadir valor a la solución, no obstante, debido a la capacidad de desarrollo actual no todas podían ser implementadas. Se priorizó que la aplicación cumpliera con los requerimientos y además cumpliera requerimientos no funcionales que caben dentro de la definición de "código de producción".



El siguiente boceto representa el árbol de componentes resultante del prototipo a papel. Esto permitió tener una mejor organización del proyecto para implementar los inicios de una arquitectura MVVM, aunque aún hay posibilidades de mejor. Si bien el bocetó fue una guía del proyecto no se siguió completamente, se fueron haciendo cambios sobre el proyecto según aparecían posibilidades de mejora o limitaciones técnicas.

Parámetros de búsqueda posibles: [Personajes Marvel]

- Nombre (exact) → Búsqueda ✓
- Empieza por ] → Filtrado ✓
- Modificado desde ]
- Comics (id) ]
- Series (id) ]
- Eventos (id) ]
- Historial (id) ]
- Ordenar por ]
  - Nombre
  - Modificado (fecha)
- Límite de resultados ]
- Offset (A de resultados a saltar) ]

Buscar / Filtrar / Ordenación

Buscar por nombre exacta

Nombre  Buscar

Empieza por:  Modificado desde:

Ordenar por:

Estructura de campos:

Tipo de búsqueda:

☐ Exacto ☐ Filtrado

Tipo de búsqueda:

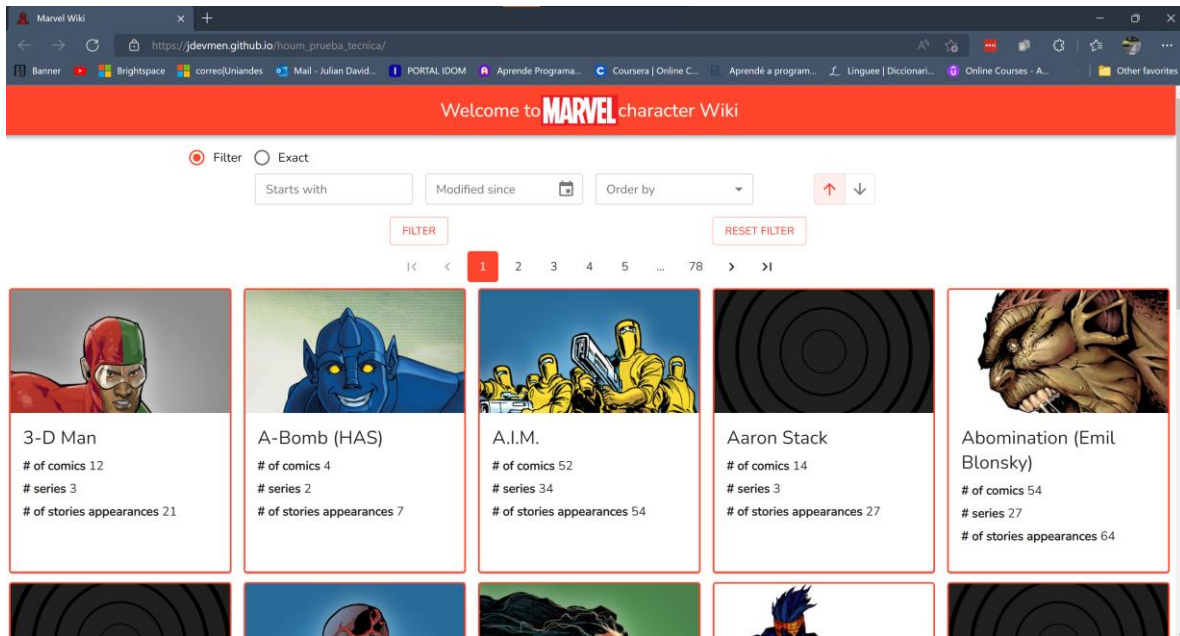
☐ Exacto ☐ Filtrado

Paginación:

...

Finalmente se hizo un listado de los parámetros que permitía la API configurar en cada petición para determinar cuáles eran relevantes y qué requerimientos cumplían. Teniendo como referencia los componentes de Material UI se hizo un boceto de los controles de la interfaz que estarían encargados de modificar los parámetros de la petición. Esto se hizo porque se consideró los requerimientos de paginación, filtrado y ordenado como una prioridad en el desarrollo de la aplicación.

## Resultado final:



En esta imagen se puede apreciar el resultado final del desarrollo. Varias de las decisiones de diseño previas pueden ver implementadas con ciertos ajustes o cambios según las limitaciones técnicas y posibilidades de mejora que se encontraron a lo largo del desarrollo.