

Guia lo que he hecho en la Práctica Global:

-APIS elegidas([ValorantAPI](https://valorant-api.com/)(no la que da RIOT)):

- <https://valorant-api.com/v1/agents> (recordar que hay un personaje que se repite dos veces y hay que usar: "isPlayableCharacter=true" para evitar que se duplique).
- <https://valorant-api.com/v1/competitivetiery>(no se hizo uso de esta ruta al final por temas de tiempo).
- <https://valorant-api.com/v1/bundles>

-Cosas por hacer(Las que no estén marcadas son las que no se han implementado):

- ☒ ~~Uso de estructuras jerárquicas de componentes dónde existan al menos una relación de abuelo-bisnieto.~~
(Se usa la jerarquía de componentes en Layout, AgentDetails y AgentCard).
- ☒ ~~Uso del hook **useState**.~~
(El useState se usa en useFetch, BundlesList y en AgentList).
- ☒ ~~Realiza alguna consulta a una api. (Puedes usar si quieres useFetch)~~
(Las consultas a las api se hacen en AgentList y BundleList además de que se hace uso del useFetch).
- ☒ ~~Uso del hook **useContext**.~~
(Se hace uso del useContext en AgentDetails y Search).
- ☒ ~~Uso de la biblioteca **React Route**:~~
 - ☒ ~~Crear un **Layout** personalizado.~~
 - ☒ ~~Creación de rutas y rutas anidadas.~~
(Hay rutas pero no rutas anidadas en App.jsx).
 - ☒ ~~Uso de **useParams** y **useSearchParams**.~~
(Se hace uso de useParams y useSearchParams en AgentDetails y en Search).
- ☒ ~~Consigue que puedas realizar búsquedas en tiempo real y que el resultado de la búsqueda se vaya mostrando en la url de tu web.~~
(Se adaptó el buscador a tiempo real de la tarea de Rick&Morty en Search).
- ☒ ~~Uso del hook **useReducer**.~~
(Se hace uso en useReducer en SearchContext).
- ☒ ~~Se valorará uso de almacenamiento local.~~
- ☐ Se valorará adicionalmente el uso de useMemo y/o useCallback.

Yeray García 2ºDAW.