

Prueba Técnica Frontend

Introducción

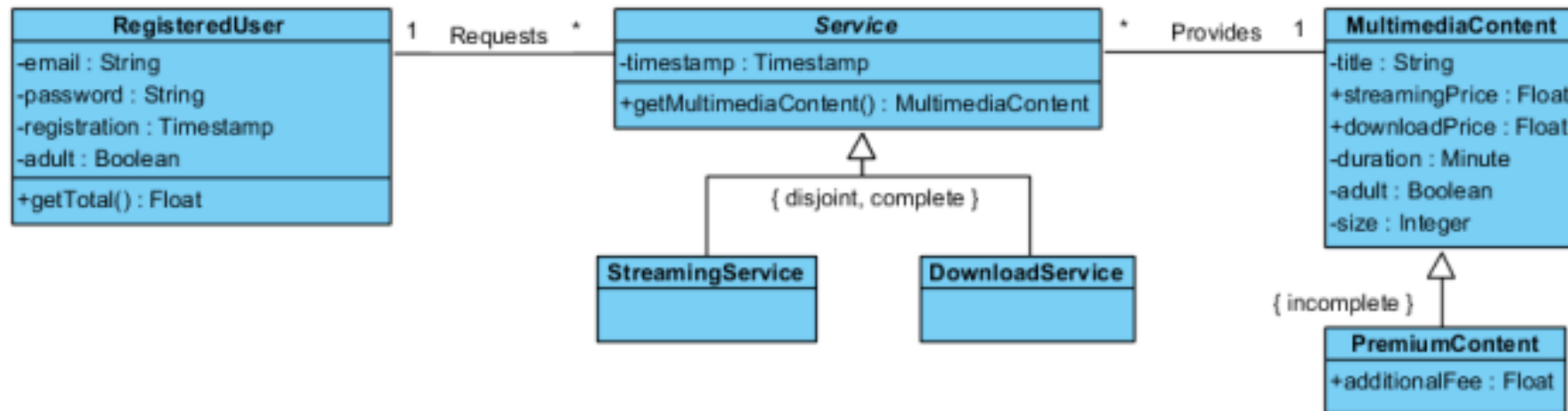
A continuación te proponemos dos ejercicios para evaluar tu nivel técnico.

El primer ejercicio es una parte mas “teórica” y en el segundo, pasamos a una parte mas práctica con un pequeño proyecto.

El código de ejemplo de la diapositiva 5, puedes encontrarlo en las notas de la misma.

Ejercicio 1

Disponemos ya de un análisis previo que podemos utilizar como punto de partida, pero que habrá que corregir y mejorar. Se trata de un software para gestionar el streaming de películas en línea.



Como podemos ver en el diagrama de partida, los usuarios tienen una operación que devuelve el importe total pagado por todos los servicios que han solicitado. El precio de un servicio se calcula de la siguiente manera:

- Para todos los usuarios que quieren ver un contenido multimedia por streaming, se aplica el precio de streaming de este contenido multimedia.
- Para todos los usuarios que quieren descargar un contenido multimedia de la plataforma, aplica el precio de descarga de este contenido multimedia.
- Si el contenido es premium, en cualquiera de los casos anteriores se añade el cargo adicional especificado en el atributo *additionalFee*.

Supongamos que la implementación de este método es como el siguiente pseudocódigo:

```
class RegisteredUser{

    constructor(services = []){
        this.services = services;
    }

    getTotal (){
        let total = 0;

        this.services.forEach(service, index => {
            let multimediaContent = service.getMultimediaContent ();

            if (typeof service == StreamingService) {
                total += multimediaContent.streamingPrice;
            }else if (typeof service == DownloadService) {
                total += multimediaContent.downloadPrice;
            }

            if (typeof multimediaContent == PremiumContent) {
                total += multimediaContent.additionalFee;
            }

        })

        return total;
    }
}
```

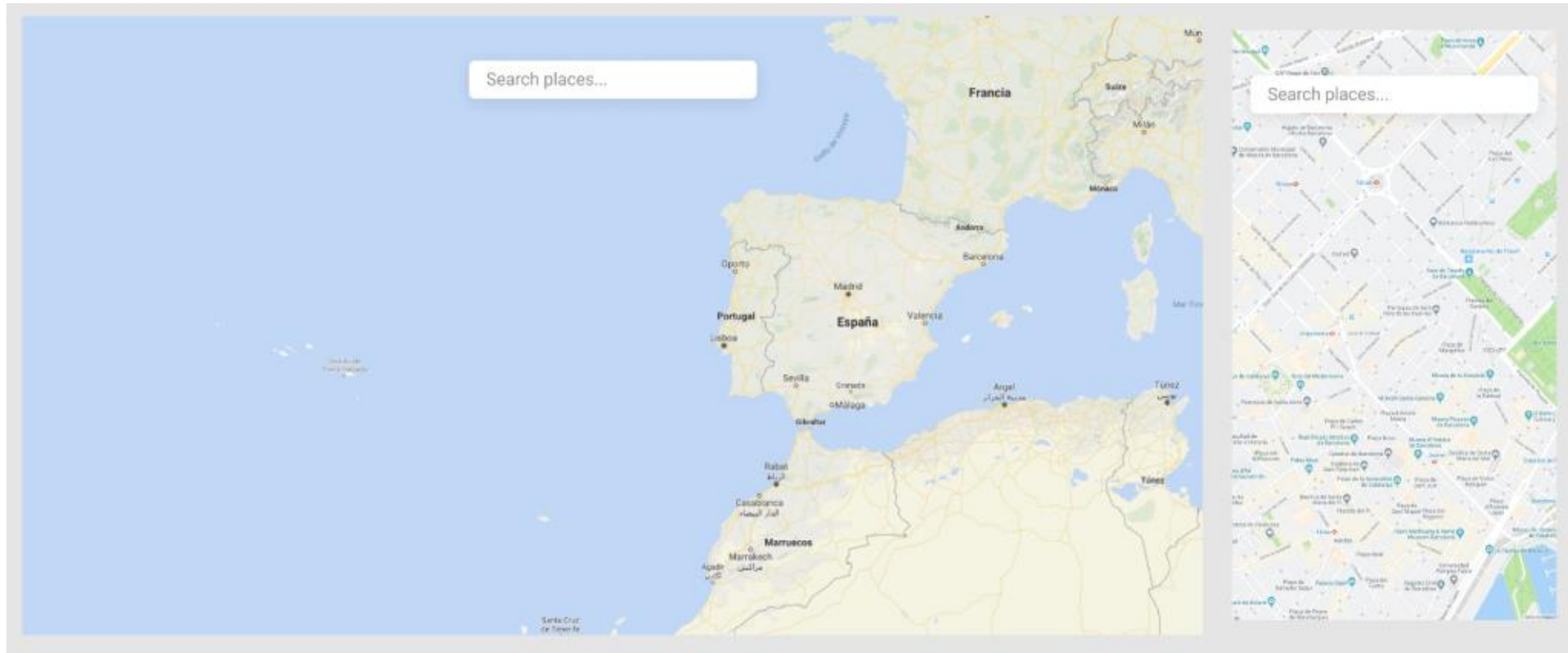
Revisamos el pseudocódigo de la operación *getTotal* de la clase *RegisteredUser* y nos preocupa que el su diseño sea un poco frágil ya que no vemos claro si contempla los posibles escenarios futuros y su impacto:

1. Que problemas detectas en la operación y razona la respuesta
2. Propón una solución alternativa (también en pseudocódigo del mismo estilo) que corrija los problemas de la operación *getTotal* de *RegisteredUser* que has detectado en la pregunta anterior. Realiza todos los cambios que consideres necesarios en cualquiera de las clases del modelo del enunciado.

Ejercicio 2

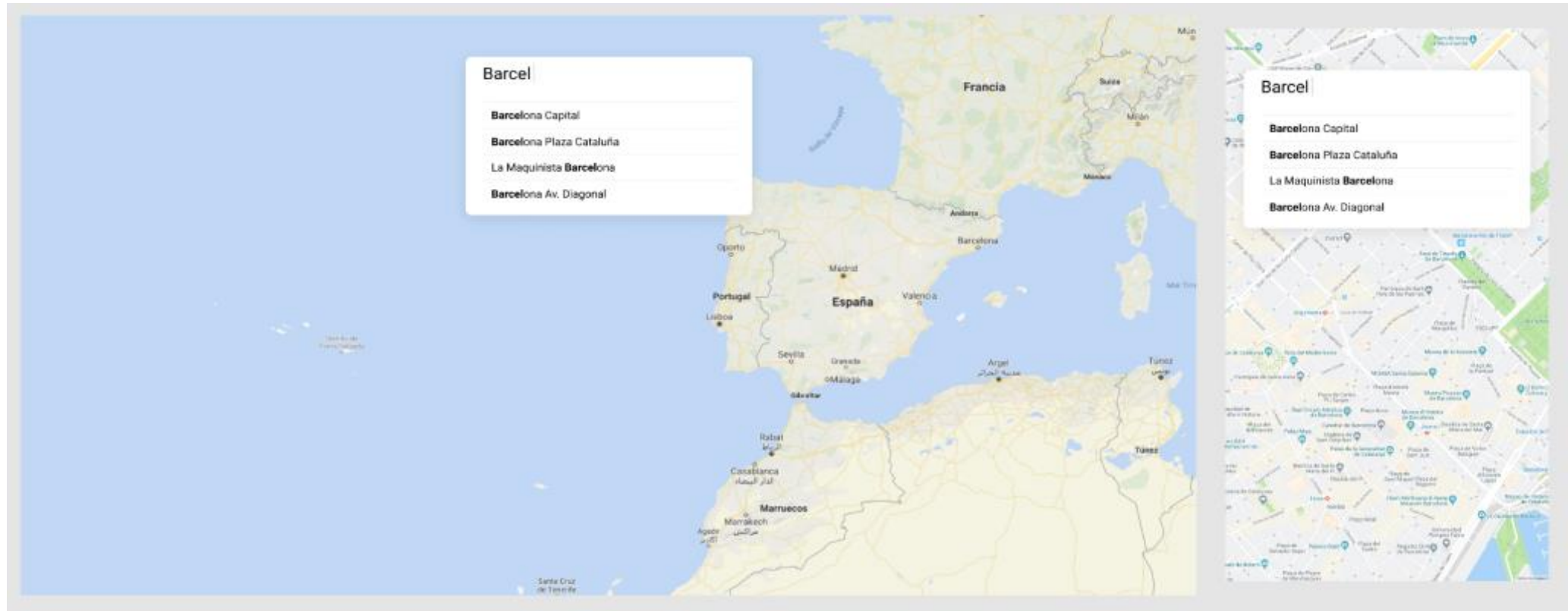
Utilizando React y la API JavaScript de Google Maps, implementa una solución responsive, sencilla pero eficaz para los siguientes diseños.

Nota: necesitarás una API KEY de Google, puedes crearte una fácilmente siguiendo [la documentación](#).



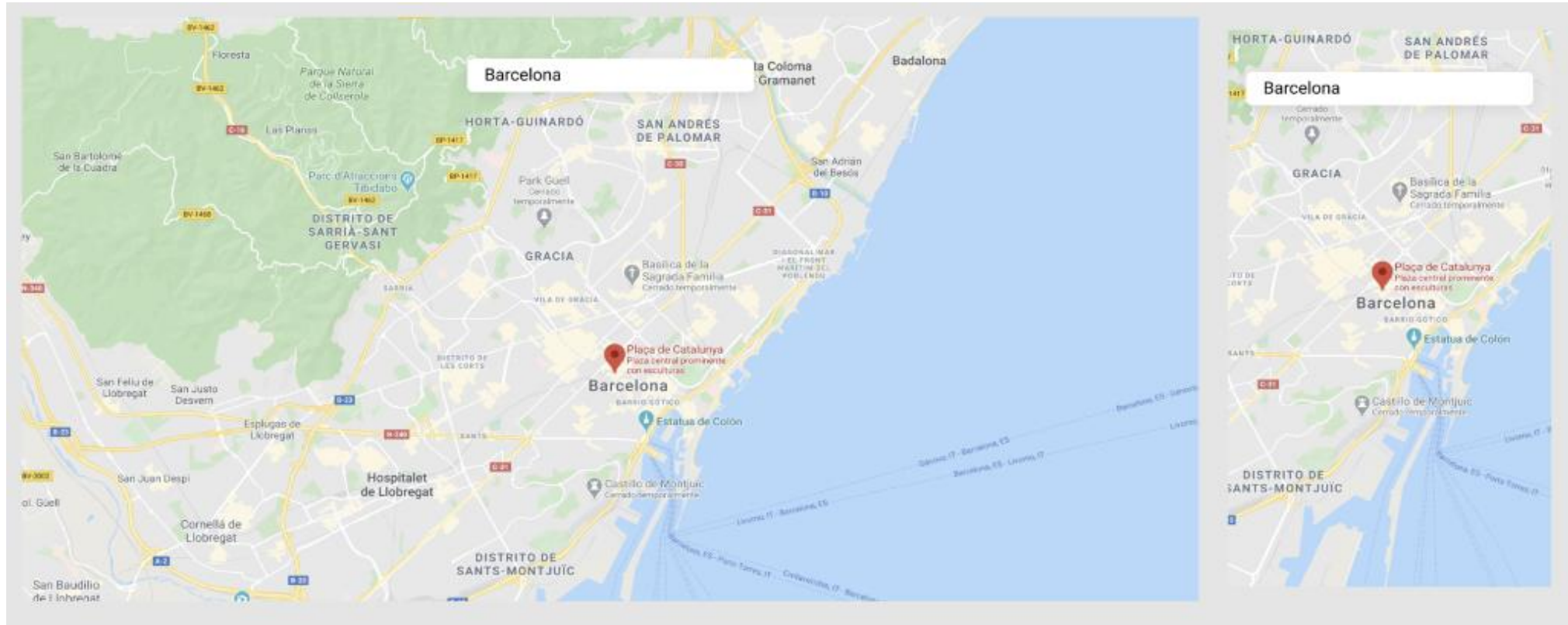
Vista principal: Mapa de Google + Buscador con sugerencias

Prueba Técnica Frontend



Buscador: Mostrar las sugerencias con estilos customizados, también, *highlight* del texto que coincida.

Prueba Técnica Frontend



Selección de sugerencia: Añadir un marker en el mapa por cada sugerencia seleccionada.

Condiciones:

- Tiene que ser con React. Eres libre de utilizar CRA y cualquier librería de React.
- Los componentes tienen que estar desacoplados, aunque utilicen la misma api de Google.
- Por cada búsqueda que el usuario realice, se tiene que guardar un “Marker” con la información de éste en un estado gestionado por Redux.
- Los estilos en general, los dejamos a tu elección.
- Implementación de algún test. Puede ser e2e, unitario o funcional.
- Optimiza la app de la mejor manera que puedas para que sea:
 1. Bundle de tamaño reducido.
 2. Poca carga de procesamiento del cliente (renders controlados).
 3. Reducir costes de llamadas a la API de Google.

El proyecto tiene que estar disponible en un repositorio git accesible, para que podamos ver el control de versiones. Además, el *README* deberá contener una explicación breve de la solución implementada y de como podemos ejecutar el proyecto.