

**Apellidos y Nombre:** \_\_\_\_\_ **Fecha:** \_\_\_\_\_

Se proporciona fichero Json con los datos de la aplicación. Crear un proyecto en IONIC cuyo nombre sea “apellidosNombreRecuperacionMarzo2019”. Ejemplo: “blanchLeivaPedroRecuperacionMarzo2019”. Se deberá utilizar OBLIGATORIAMENTE el interface que se indica en la página 2.

**Ejercicio 1.- (1,5 puntos)** La pantalla principal de la aplicación mostrará la pantalla dividida en dos RecyclerViews, cada uno ocupando el 40% y 60% respectivamente de la pantalla. En el RecyclerView de la parte superior aparecerá una fila por cada carrera existente en el servidor web. Se mostrará la descripción de la carrera, distancia y fecha. Las carreras se mostrarán ordenadas por fecha de forma descendente.

**Ejercicio 2.- (1,5 puntos)** Al hacer click en cualquier carrera se buscarán en el servidor web los participantes en dicha carrera, mostrándose en el RecyclerView ubicado en la zona inferior de la pantalla. En cada fila se mostrará: posición, apellidos y nombre, tiempo y categoría. Los usuarios de cada carrera se mostrarán ordenados por tiempo de forma ascendente.

**Ejercicio 3.- (3 puntos)** Añadir en el action bar de la App un botón “+” que al ser pulsado permita añadir una nueva carrera al servicio web. Se deberán cumplir los siguientes criterios:

**fecha** en formato YYYY-MM-DD, **descripción** de la carrera solo debe admitir letras en mayúsculas o minúsculas, **distancia** numérico positivo sin decimales, latitud y longitud numéricos positivos o negativos con dos decimales como mínimo y un dígito como mínimo en la parte entera, debiendo tener el punto como separación decimal. El botón de validación estará desactivado mientras falte algún campo por rellenar, o bien no se cumpla alguno de los requisitos anteriores.

Al pulsar el botón aceptar se añadirá la carrera en el servidor web JSON.

**Se deberá realizar OBLIGATORIAMENTE con un formulario.**

**Ejercicio 4.- (2,5 puntos)** Al iniciarse la App se mostrará una pantalla con una caja de texto donde el usuario podrá escribir sus apellidos. Al pulsar un botón “buscar” del action bar se hará una petición al servidor web de los usuarios cuyos apellidos concuerdan con los caracteres introducidos. Se mostrarán los usuarios en una lista ubicada debajo de la caja de texto.

Al seleccionar uno de los usuarios de la lista se pasará a la pantalla principal de la aplicación, donde se mostrarán sus apellidos y nombre en el título del action bar.

**Ejercicio 5.- (1,5 puntos)** Añadir un botón en el action bar que al ser pulsado busque en la lista de clasificaciones de usuarios al usuario logueado en la aplicación (tenga igual ID). Si lo encuentra mostrará en una ventana emergente el nombre de la carrera, su posición y tiempo.

Notas de ayuda:

- Clasificaciones de una carrera por orden ascendente de tiempos:

[http://localhost:8080/clasificaciones?idCarrera=4&\\_sort=tiempo&\\_order=ASC](http://localhost:8080/clasificaciones?idCarrera=4&_sort=tiempo&_order=ASC)

- Carreras por orden descendente de fecha:

[http://localhost:8080/carreras?\\_sort=fecha&\\_order=DESC](http://localhost:8080/carreras?_sort=fecha&_order=DESC)

- Usuarios cuyos apellidos empiezan por Blan: (CUIDADO: SE DISTINGUEN MAYÚSCULAS Y MINÚSCULAS)

[http://localhost:8080/usuarios?apellidos\\_gte=Blan&apellidos\\_lte=Blanz](http://localhost:8080/usuarios?apellidos_gte=Blan&apellidos_lte=Blanz) (se añade una z al final)

- Obtener un usuario a partir de su id: (CIUDADO: DEVUELVE UN JSONObject)

<http://localhost:8080/usuarios/10>

```
export interface UserServiceProviderListener {  
    onGetUsuarioResponse(usuario: Usuario, errorString);  
    onGetUsuariosPorCadenaBusquedaResponse(usuarios:Usuario[], error:string);  
    onGetUsuariosResponse(usuarios: Map<number,Usuario>, error:string);  
    onGetCarrerasResponse(carreras:Carrera[], error:String);  
    onGetClasificacionesCarreraResponse(clasificaciones:Clasificacion[],  
    error:string);  
    onPostCarreraResponse(carrera:Carrera, error:string);  
}
```