

# **Documentación aplicación web para FES**

## **1. API**

### **1.1. Definición de API**

La Federación de Escuelas de Samuráis (FES) ha solicitado una aplicación donde cada escuela puede introducir una serie de datos climatológicos de su respectiva ciudad. Los datos para recoger son los siguientes:

- Temperatura
- Humedad
- Presión atmosférica
- Viento
- Precipitaciones
- Radiación solar
- Índices de calidad del aire
- Nubosidad
- Índices UV

Las escuelas han acordado realizar la inserción de datos a la misma hora. El objetivo final es, al finalizar del año, consultar todos los datos para elegir la ciudad que será la sede de un torneo en el que participan las escuelas de la federación.

Debido a la sencillez de los requisitos del cliente, sugiero desarrollar un API Rest para realizar las operaciones CRUD.

### **1.2. Consumo de API desde el Frontend**

Debido a los requisitos expuestos anteriormente, se necesita una interfaz de usuarios que facilite el consumo del API Rest en los siguientes casos:

- Peticiones GET: mostrar registros de una determinada escuela y mostrar registros de un determinado día
- Peticiones POST: insertar registros asegurando que no hay registro de dicha escuela en ese mismo día.
- Peticiones PUT y DELETE: modificar o borrar un registro usando como criterio de búsqueda su fecha y la escuela.

### **1.3. Manejo de respuestas**

A continuación se va a describir qué respuesta debe dar el API Rest para cada una de las peticiones expuestas en el apartado anterior.

- Mostrar registros de una escuela o de una fecha (GET): debe devolver una lista de todos los registros que cumplan los criterios de búsqueda.
- Mostrar un registro usando los criterios de búsqueda de fecha y escuela: debe devolver un registro o un mensaje de error si no se ha encontrado. Esta petición será la base para realizar peticiones PUT y la respuesta tras haber realizado una operación POST.
- Modificar o borrar registro (PUT y DELETE): se debe mostrar los datos del registro en caso de modificación o un mensaje afirmando que “los datos del día ‘x’ se han borrado correctamente” en caso de borrado.

#### **1.4. Actualización de la interfaz de usuario**

En esta sección se va a describir qué debe mostrar la interfaz de usuario después de realizar una operación en el API Rest.

- Peticiones GET: dependiendo de si el API devuelve una lista o un registro las interfaces deben ser distintas. En caso de un listado, debe mostrar una lista con todos los registros ordenados por su fecha, mostrando primeros los elementos más recientes. Para que el API devuelva solo un registro, el usuario tiene que hacer click sobre un registro del listado.
- Peticiones PUT y DELETE: una vez el API ha devuelto el registro seleccionado por el usuario, se deben mostrar unos botones para habilitar la edición del registro o el borrado de este. Después mostrar el registro modificado o un mensaje de borrado exitoso.
- Peticiones POST: tras realizar la inserción, se debe mostrar el registro recién insertado con los botones de modificación y borrado explicados anteriormente.

#### **1.5. Gestión de sesiones**

Un aspecto fundamental de esta aplicación es la gestión de sesiones. Antes de realizar cualquier operación sobre el API, el usuario debe autenticarse. Debe haber dos tipos de usuarios: uno para cada y otro para el presidente de la federación.

#### **1.6. Autenticación y seguridad.**

Respecto a la autenticación, este aspecto es tan importante porque el usuario de tipo escuela solo debe poder realizar operaciones CRUD en los registros que pertenecen a su escuela. Por otra parte el usuario del presidente de la federación tendrá acceso a los registros de todas las escuelas.

Para garantizar la seguridad e integridad de los datos es importante emplear el protocolo HTTPS y asegurar que los datos introducidos por el usuario están en un formato correcto.

## **2. Pruebas**

En esta sección se va a exponer cómo deben realizarse los distintos tipos de pruebas para cada uno de los casos de peticiones al API Rest.

### **2.1 Pruebas unitarias**

Las pruebas unitarias van a consistir en asegurar que cada petición al API devuelve la respuesta deseada ya sea de éxito o del error.

- **Mostrar listado de registros:** se debe asegurar la devolución del listado correspondiente según el criterio de búsqueda (fecha o escuela) o un mensaje de error si no se han encontrado registros.
- **Mostrar un registro según fecha y escuela:** hay que garantizar la devolución de un solo registro de acuerdo con los criterios de búsqueda o mensaje de error en caso no hallarse registro.
- **Insertar, modificar o borrar registro:** el usuario no debe acceder a la fecha y ni la escuela, pues estos valores serán la fecha actual y la escuela con la que se ha iniciado sesión. También se debe evitar que el usuario inserte datos en un formato erróneo. Hay que comprobar que la operación correspondiente se ha realizado con éxito usando la petición GET de obtención de un registro.

En todos los casos se debe mostrar un mensaje de operación no autorizada si el usuario de tipo escuela pretende hacer una operación CRUD sobre un registro que pertenece a otra escuela.

### **2.2. Pruebas de integración**

El aspecto más destacado de las pruebas de integración son los dos siguientes:

- **Listado de registro:** se debe garantizar la correcta transición de la petición del listado a la de mostrar el registro seleccionado por el usuario.
- **Operaciones POST, PUT, DELETE:** en el caso de los dos primeros, se debe mostrar automáticamente los datos del registro insertado o modificado. En cambio, cuando se realiza un borrado, se debe mostrar un mensaje de 'operación exitosa' y luego mostrar el listado de registros de dicha escuela.

### 2.3. Pruebas de aceptación

Finalmente, se deben realizar pruebas de aceptación para mostrar al presidente de la federación que la aplicación cumple con lo demandado. Las pruebas que se van a explicar se deben hacer con los dos tipos de usuarios y estas pruebas son las siguientes:

- Pruebas solo de usuario presidente:
  - Mostrar registro por fecha o por escuela, según lo desee. Debe mostrarse una interfaz clara e intuitiva.
- Pruebas solo usuario escuela:
  - Mostrar los registros de su escuela: debe mostrarse una interfaz clara e intuitiva. En caso de querer acceder a los registros de otra escuela debe mostrar un mensaje de ‘operación no autorizada’.
- Pruebas de ambos usuarios:
  - Mostrar registro seleccionado: debe mostrarse que, al hacer click en un elemento de la lista, se muestra solo dicho registro.
  - Insertar registro: el presidente debe ver un formulario que recoja los datos climatológicos y, después de enviar el formulario, mostrar los datos del registro introducido.
  - Modificar o borrar un registro: tras seleccionar un registro del listado y mostrarse este, debe verse de forma clara e intuitiva los botones de edición y borrado. Tras realizar la acción correspondiente, el presidente debe ver que se muestra en la aplicación una respuesta clara del resultado de la acción, ya sea mostrando el registro modificado o mostrar la listado de registros de la escuela para que el usuario que el registro borrado ya no está.