

## PRÁCTICA 6.03 Gestionar datos con Firebase

### Normas de entrega

- En cuanto al **código**:
  - en la **presentación interna**, importan los **comentarios**, la claridad del código, la significación de los nombres elegidos; todo esto debe permitir considerar al programa como **autodocumentado**. No será necesario explicar que es un **if** un **for** pero sí su funcionalidad. Hay que comentar las cosas destacadas y, sobre todo, las **funciones** y **clases** empleadas. La ausencia de comentarios será penalizada,
  - en la **presentación externa**, importan las leyendas aclaratorias, información en pantalla y avisos de ejecución que favorezcan el uso de la aplicación,
  - todo el código debe estar situado dentro del evento `window.onload = () => {};` o a través del evento `document.addEventListener("DOMContentLoaded", () => {});`,
  - si no se especifica lo contrario, la información resultante del programa debe aparecer en la consola del navegador `console.log(información)`,
  - los ejercicios deben realizarse usando **JavaScript ES6** y usar el modo estricto (`use strict`) No se podrá utilizar jQuery ni cualquier otra biblioteca (si no se especifica lo contrario en el enunciado),
  - para el nombre de **variables**, **constantes** y **funciones** se utilizará lowerCamelCase,
  - para la asignación de eventos se utilizará `addEventListener()` indicando sus tres parámetros en su definición,
  - debes dividir tu código en **bibliotecas temáticas** de funciones y/o clases a partir de este ejercicio,
  - se usarán las funcionalidades `import` y `export` para crear bibliotecas de funciones temáticas a partir de esta práctica,
  - todo el código que sea susceptible de retrasar su ejecución debe escribirse de forma asíncrona.
- En cuanto a la **entrega** de los archivos que componen los ejercicios:
  - todos los ejercicios en **una carpeta** (creando las **subcarpetas** necesarias para documentación anexa como imágenes o estilos) cuyo nombre queda a discreción del discente,
  - el nombre de los ficheros necesarios para resolver el ejercicio será el número de ejercicio que contenga,
  - el código contendrá ejemplos de ejecución, si procede, y
  - la carpeta será **comprimida** en formato **ZIP** y será subida a **Aules** de forma puntual.

### **Ejercicio 1 - Agenda CRUD con firebase**

Adapta la aplicación web de la práctica anterior permita el mantenimiento de un pequeño listado con el nombre, apellidos, dirección y teléfono de varias personas. Toda la información que contiene debe ser persistente y guardarse en este caso en **Firestore** íntegramente.

Desarrolla una aplicación que maneje la agenda directamente en Firestore, superando las limitaciones de la actividad anterior. Implementarán funciones para añadir, listar, editar, buscar y borrar contactos, con sincronización en tiempo real mediante `onSnapshot`. Esto asegura que cualquier cambio en la base de datos se refleje automáticamente en la interfaz de la aplicación.

El trabajo consiste en crear formularios para las operaciones principales, diseñar consultas eficientes para la búsqueda de contactos por campos específicos, y gestionar los cambios en los datos a través de la integración con Firestore. La funcionalidad aquí es superior a la de la actividad 1 y 2 porque combina persistencia en la nube y actualización dinámica de los datos.

Comprueba trabajar contra la base de datos desde dos terminales para comprobar que permite trabajar de forma colaborativa y que puede ser accedida por múltiples usuarios.