

App de Gestión - Firebase (Proyecto)

Este documento contiene la estructura del proyecto, instrucciones de configuración de Firebase y todos los archivos separados (HTML, CSS y JS) listos para copiar/pegar.

Estructura del proyecto

```
/app-gestion-firebase
  /css
    styles.css
  /js
    firebase-config.js
    auth.js
    router.js
    modules.js
    app.js
  index.html
  README.md
```

Pasos rápidos de configuración (Firebase)

1. Ir a <https://console.firebase.google.com> y crear un nuevo proyecto.
 2. En el proyecto, en "Authentication" -> "Sign-in method" habilitar **Email/Password**.
 3. En "Firestore Database" crear una base de datos en modo de prueba (o con reglas que prefieras). Usa la región más cercana.
 4. En "Project settings" -> "Your apps" -> "</> Web" agregar una nueva app web. Copiar el objeto de configuración (apiKey, authDomain, projectId, etc.).
 5. Reemplazar los valores de `firebase-config.js` por los de tu app web.
 6. Servir los archivos (puede ser con `live-server`, `http-server` o subir a un hosting estático). También podés usar GitHub Pages.
-

README: qué hace este proyecto

- Login con Firebase Auth (email/password). Usuario admin puede crearse desde la consola de Firebase o registrarse desde la app.
 - CRUD básico sobre Firestore para: `clientes`, `stock`, `pedidos`, `pagos`.
 - Router simple para navegar entre módulos (Lead, Pedido, Stock, Cobranza).
 - Código modular separado en archivos JS.
-

index.html

```
<!doctype html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>App de Gestión - Firebase</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/
bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <div class="container py-4">
      <h1 class="text-center mb-4">App de Gestión</h1>

      <!-- Login -->
      <div id="login-section">
        <h2>Ingreso de Info</h2>
        <form id="login-form" class="mb-3">
          <div class="mb-3">
            <label for="email" class="form-label">Email</label>
            <input id="email" type="email" class="form-control" required>
          </div>
          <div class="mb-3">
            <label for="password" class="form-label">Contraseña</label>
            <input id="password" type="password" class="form-control"
required>
          </div>
          <div class="d-flex gap-2">
            <button class="btn btn-primary" id="login-btn">Ingresar</button>
            <button class="btn btn-outline-secondary" id="register-btn"
type="button">Registrar</button>
          </div>
        </form>
      </div>

      <!-- Router -->
      <div id="router-section" class="hidden">
        <h2>Router - Selecciona un Módulo</h2>
        <div class="row g-2">
          <div class="col-6 col-md-3"><button class="btn btn-outline-primary
w-100" data-module="lead">Lead</button></div>
          <div class="col-6 col-md-3"><button class="btn btn-outline-primary
w-100" data-module="pedido">Pedido</button></div>
          <div class="col-6 col-md-3"><button class="btn btn-outline-primary
w-100" data-module="stock">Stock</button></div>
          <div class="col-6 col-md-3"><button class="btn btn-outline-primary
w-100" data-module="cobranza">Cobranza</button></div>
        </div>
      </div>
    </div>
  </body>
</html>
```

```

        <div class="mt-3">
            <button class="btn btn-secondary" id="logout-btn">Cerrar sesión</
button>
        </div>
    </div>

    <!-- Modules container -->
    <div id="modules-container" class="mt-3"></div>

</div>

<!-- Scripts -->
<script type="module" src="js/firebase-config.js"></script>
<script type="module" src="js/app.js"></script>
</body>
</html>

```

css/styles.css

```

:root{
    --bg: #f8f9fa;
}
body{background:var(--bg);}
.container{max-width:900px}
.hidden{display:none}
.form-section{margin-top:20px}
.card-module{padding:16px;margin-top:16px}

```

js/firebase-config.js

```

// Archivo que inicializa Firebase y exporta utilidades
import { initializeApp } from 'https://www.gstatic.com/firebasejs/9.22.2/
firebase-app.js';
import { getAuth } from 'https://www.gstatic.com/firebasejs/9.22.2/firebase-
auth.js';
import { getFirestore } from 'https://www.gstatic.com/firebasejs/9.22.2/
firebase-firestore.js';

// REEMPLAZA con tu config desde Firebase Console
const firebaseConfig = {
    apiKey: "REPLACE_API_KEY",
    authDomain: "REPLACE_AUTH_DOMAIN",
    projectId: "REPLACE_PROJECT_ID",
    storageBucket: "REPLACE_STORAGE_BUCKET",
    messagingSenderId: "REPLACE_MESSAGING_SENDER_ID",

```

```
    appId: "REPLACE_APP_ID"
  };

  const app = initializeApp(firebaseConfig);
  export const auth = getAuth(app);
  export const db = getFirestore(app);
```

js/app.js

```
// Punto de entrada: monta UI y conecta módulos
import { auth } from './firebase-config.js';
import { initAuth } from './auth.js';
import { initRouter } from './router.js';
import { initModules } from './modules.js';

// Inicializar piezas
initAuth();
initRouter();
initModules();

// Observador de estado de auth (muestra u oculta secciones)
import { onAuthStateChanged } from 'https://www.gstatic.com/firebasejs/9.22.2/firebase-auth.js';

const loginSection = document.getElementById('login-section');
const routerSection = document.getElementById('router-section');

onAuthStateChanged(auth, user => {
  if (user) {
    loginSection.classList.add('hidden');
    routerSection.classList.remove('hidden');
  } else {
    loginSection.classList.remove('hidden');
    routerSection.classList.add('hidden');
    document.getElementById('modules-container').innerHTML = '';
  }
});
```

js/auth.js

```
// Manejo de login y registro
import { auth } from './firebase-config.js';
import { signInWithEmailAndPassword, createUserWithEmailAndPassword } from 'https://www.gstatic.com/firebasejs/9.22.2/firebase-auth.js';
```

```

export function initAuth(){
  const loginForm = document.getElementById('login-form');
  const loginBtn = document.getElementById('login-btn');
  const registerBtn = document.getElementById('register-btn');

  loginForm.addEventListener('submit', async (e)=>{
    e.preventDefault();
    const email = document.getElementById('email').value.trim();
    const password = document.getElementById('password').value;
    try{
      await signInWithEmailAndPassword(auth, email, password);
    }catch(err){
      alert('Error al ingresar: ' + err.message);
    }
  });

  registerBtn.addEventListener('click', async ()=>{
    const email = document.getElementById('email').value.trim();
    const password = document.getElementById('password').value;
    if(!email || !password){ alert('Completa email y contraseña para registrar'); return; }
    try{
      await createUserWithEmailAndPassword(auth, email, password);
      alert('Usuario registrado. ');
    }catch(err){ alert('Error registro: ' + err.message); }
  });

  document.getElementById('logout-btn').addEventListener('click', ()=>
  auth.signOut());
}

```

js/router.js

```

// Router simple: renderiza módulos en #modules-container
export function initRouter(){
  document.querySelectorAll('[data-module]').forEach(btn=>{
    btn.addEventListener('click', ()=>{
      const module = btn.dataset.module;
      renderModule(module);
    });
  });
}

function renderModule(name){
  const container = document.getElementById('modules-container');
  // Limpia
  container.innerHTML = '';
  if(name === 'lead') container.innerHTML = leadTemplate();
}

```

```

if(name === 'pedido') container.innerHTML = pedidoTemplate();
if(name === 'stock') container.innerHTML = stockTemplate();
if(name === 'cobranza') container.innerHTML = cobranzaTemplate();

// Inicializa handlers del módulo
import('./modules.js').then(m => m.initModule(name));
}

function leadTemplate(){
  return `
    <div class="card card-module">
      <h3>Lead - Crear Cliente</h3>
      <form id="lead-form">
        <div class="mb-3">
          <label class="form-label">Nombre</label>
          <input id="cliente-nombre" class="form-control" required>
        </div>
        <div class="mb-3">
          <label class="form-label">Email</label>
          <input id="cliente-email" type="email" class="form-control"
required>
        </div>
        <button class="btn btn-success">Crear Cliente</button>
      </form>
      <div id="lead-list" class="mt-3"></div>
    </div>
  `;
}

function pedidoTemplate(){
  return `
    <div class="card card-module">
      <h3>Pedido - Buscar Stock</h3>
      <form id="buscar-stock-form">
        <div class="mb-3">
          <label class="form-label">ID del Producto</label>
          <input id="producto-id" class="form-control" required>
        </div>
        <button class="btn btn-info">Buscar Stock</button>
      </form>
      <div id="stock-result" class="mt-3 hidden">
        <p>Stock disponible: <span id="stock-cantidad"></span></p>
        <button class="btn btn-success" id="btn-create-pedido">Crear Pedido</
button>
      </div>
      <div id="create-pedido-section" class="hidden mt-3">
        <h4>Crear Pedido</h4>
        <form id="create-pedido-form">
          <div class="mb-3"><label class="form-label">Cantidad</label><input
id="pedido-cantidad" type="number" class="form-control" required></div>
          <button class="btn btn-success">Crear Pedido</button>
        </form>
      </div>
    </div>
  `;
}

```

```

        </form>
    </div>
</div>
`;
}

function stockTemplate(){
    return `
        <div class="card card-module">
            <h3>Stock - Crear/Actualizar Stock</h3>
            <form id="stock-form">
                <div class="mb-3"><label class="form-label">ID del Producto</
label><input id="stock-id" class="form-control" required></div>
                <div class="mb-3"><label class="form-label">Cantidad</label><input
id="stock-cantidad" type="number" class="form-control" required></div>
                <button class="btn btn-warning">Crear/Actualizar Stock</button>
            </form>
            <div id="stock-list" class="mt-3"></div>
        </div>
    `;
}

function cobranzaTemplate(){
    return `
        <div class="card card-module">
            <h3>Cobranza - Buscar Pedido</h3>
            <form id="buscar-pedido-form">
                <div class="mb-3"><label class="form-label">ID del Pedido</
label><input id="pedido-id" class="form-control" required></div>
                <button class="btn btn-info">Buscar Pedido</button>
            </form>
            <div id="pedido-result" class="mt-3 hidden">
                <p>Pedido encontrado: <span id="pedido-detalle"></span></p>
                <button class="btn btn-success" id="btn-registrar-pago">Registrar
Pago</button>
            </div>
            <div id="registrar-pago-section" class="hidden mt-3">
                <h4>Registrar Pago</h4>
                <form id="registrar-pago-form">
                    <div class="mb-3"><label class="form-label">Monto</label><input
id="pago-monto" type="number" class="form-control" required></div>
                    <button class="btn btn-success">Registrar Pago</button>
                </form>
            </div>
        </div>
    `;
}

```

js/modules.js

```
// Funciones CRUD que interactúan con Firestore
import { db } from './firebase-config.js';
import {
  collection, addDoc, getDocs, doc, getDoc, updateDoc, setDoc, query, where
} from 'https://www.gstatic.com/firebasejs/9.22.2/firebase-firestore.js';

export function initModule(name){
  if(name === 'lead') initLead();
  if(name === 'pedido') initPedido();
  if(name === 'stock') initStock();
  if(name === 'cobranza') initCobranza();
}

// LEAD
async function initLead(){
  const form = document.getElementById('lead-form');
  const list = document.getElementById('lead-list');
  form.addEventListener('submit', async (e)=>{
    e.preventDefault();
    const nombre = document.getElementById('cliente-nombre').value.trim();
    const email = document.getElementById('cliente-email').value.trim();
    await addDoc(collection(db, 'clientes'), {nombre, email, createdAt:
Date.now()});
    alert('Cliente creado');
    form.reset();
    renderClientes();
  });
  renderClientes();
}

async function renderClientes(){
  const q = await getDocs(collection(db, 'clientes'));
  const list = document.getElementById('lead-list');
  list.innerHTML = '';
  q.forEach(docSnap=>{
    const d = docSnap.data();
    const el = document.createElement('div');
    el.textContent = `${d.nombre} - ${d.email}`;
    list.appendChild(el);
  });
}

// PEDIDO
async function initPedido(){
  const buscarForm = document.getElementById('buscar-stock-form');
  const stockResult = document.getElementById('stock-result');
  buscarForm.addEventListener('submit', async (e)=>{
    e.preventDefault();
  });
}
```



```

    const id = document.getElementById('producto-id').value.trim();
    const stockDoc = await getDoc(doc(db, 'stock', id));
    if(stockDoc.exists()){
        document.getElementById('stock-cantidad').textContent =
stockDoc.data().cantidad;
        stockResult.classList.remove('hidden');
        window.currentStockId = id;
    } else { alert('Producto no encontrado'); }
});

document.getElementById('btn-create-pedido').addEventListener('click',
()=>{
    document.getElementById('create-pedido-
section').classList.remove('hidden');
});

document.getElementById('create-pedido-form').addEventListener('submit',
async (e)=>{
    e.preventDefault();
    const cantidad = Number(document.getElementById('pedido-
cantidad').value);
    const pedidosRef = collection(db, 'pedidos');
    const id = 'ped' + Date.now();
    await setDoc(doc(db, 'pedidos', id), {id, cantidad, stockId:
window.currentStockId || null, createdAt: Date.now()});
    alert('Pedido creado');
    document.getElementById('create-pedido-form').reset();
});
}

// STOCK
async function initStock(){
    const form = document.getElementById('stock-form');
    const list = document.getElementById('stock-list');
    form.addEventListener('submit', async (e)=>{
        e.preventDefault();
        const id = document.getElementById('stock-id').value.trim();
        const cantidad = Number(document.getElementById('stock-cantidad').value);
        await setDoc(doc(db, 'stock', id), {cantidad, updatedAt: Date.now()});
        alert('Stock actualizado');
        form.reset();
        renderStock();
    });
    renderStock();
}

async function renderStock(){
    const q = await getDocs(collection(db, 'stock'));
    const list = document.getElementById('stock-list');
    list.innerHTML = '';
    q.forEach(snap=>{

```

```

    const d = snap.data();
    const el = document.createElement('div');
    el.textContent = `${snap.id} - ${d.cantidad}`;
    list.appendChild(el);
  });
}

// COBRANZA
async function initCobranza(){
  const buscarForm = document.getElementById('buscar-pedido-form');
  buscarForm.addEventListener('submit', async (e)=>{
    e.preventDefault();
    const id = document.getElementById('pedido-id').value.trim();
    const pedidoDoc = await getDoc(doc(db, 'pedidos', id));
    if(pedidoDoc.exists()){
      document.getElementById('pedido-detalle').textContent = `Cantidad: $
{pedidoDoc.data().cantidad}`;
      document.getElementById('pedido-result').classList.remove('hidden');
      window.currentPedidoId = id;
    } else { alert('Pedido no encontrado'); }
  });

  document.getElementById('btn-registrar-pago').addEventListener('click',
  ()=>{
    document.getElementById('registrar-pago-
section').classList.remove('hidden');
  });

  document.getElementById('registrar-pago-form').addEventListener('submit',
  async (e)=>{
    e.preventDefault();
    const monto = Number(document.getElementById('pago-monto').value);
    await setDoc(doc(db, 'pagos', 'pago' + Date.now()), {monto, pedidoId:
window.currentPedidoId || null, createdAt: Date.now()});
    alert('Pago registrado');
    document.getElementById('registrar-pago-form').reset();
  });
}

```

Notas y mejoras sugeridas

- Reglas de seguridad: al empezar en modo desarrollo Firestore estará en modo de pruebas. Cambialas antes de producción.
- Indicar roles (admin) con claims personalizados si necesitás controles más finos.
- Validaciones y UX: agregar loaders y validaciones en formularios.
- Considerar paginación si tendrás miles de documentos.
- Para integraciones (Make, Zapier): podés usar Cloud Functions o conectar Firestore directamente.

Si querés que te entregue estos archivos en un ZIP descargable o que los suba a un repo GitHub, decímelo y lo preparo.