

Componente de Cliente



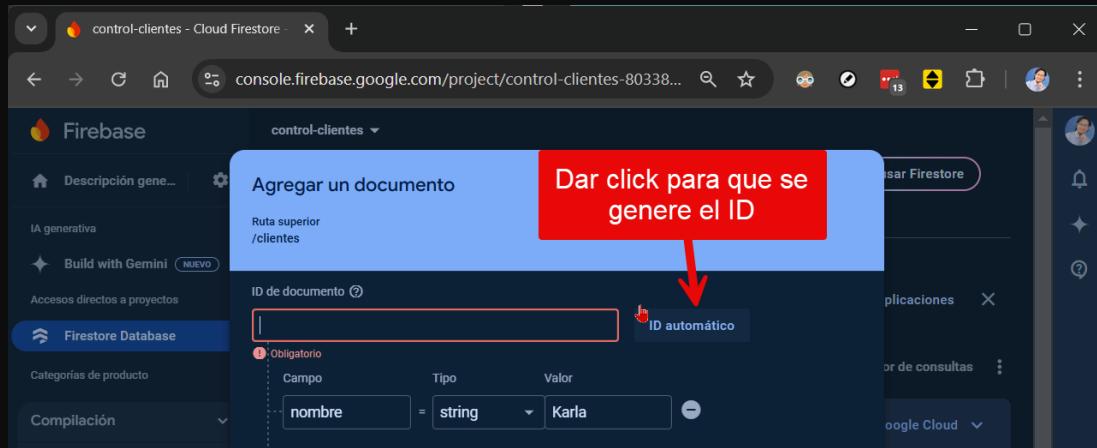
Listado de Clientes en Angular con Firestore

En esta lección, vamos a trabajar con el **listado de clientes**. Aprenderemos a agregar nuevos clientes y a consumir el servicio para recuperar la lista de clientes almacenada en la base de datos de **Cloud Firestore**.

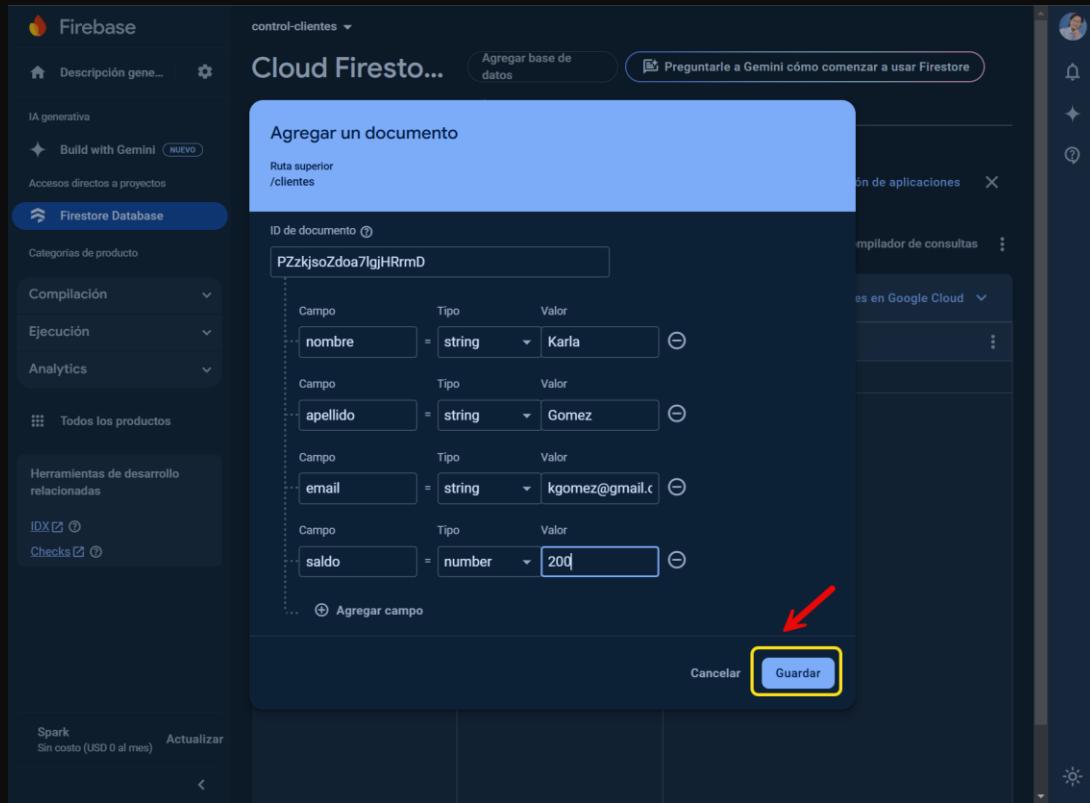
En primer lugar, vamos a agregar un nuevo a la colección de clientes.

The screenshot shows the Google Cloud Firestore console. On the left, a sidebar lists a collection named 'clientes'. In the main area, a table shows a single document with the ID 'Ow546ztseGV7zizHoYil'. A red arrow points to the 'Agregar documento' button in the top right corner of the table row. To the right of the table, the document details are shown: 'apellido: "Perez"', 'email: "jperez@mail.com"', 'nombre: "Juan"', and 'saldo: 100'.

Recuerden que debemos dar click en el botón de ID automático para que se genere el ID y se pueda guardar el nuevo registro, de lo contrario no permite guardarlo.



Registro completo:



Una vez que hayas agregado un cliente, modificaremos el componente para consumir el servicio de clientes y mostrar el listado actualizado en la vista.

Modificar el Componente de Clientes

Vamos a modificar el componente `ClientsComponent` para recuperar el listado de clientes almacenado en la base de datos de **Cloud Firestore**.

Código: `clients.component.ts`

```

import { Component } from '@angular/core';
import { Cliente } from '../../modelo/cliente.modelo';
import { ClienteServicio } from '../../servicios/cliente.service';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-clientes',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './clientes.component.html',
  styleUrls: ['./clientes.component.css']
})
export class ClientesComponent {
  clientes: Cliente[] | null = null;

  constructor(private clientesServicio: ClienteServicio) {}

  ngOnInit() {
    // Al inicializar el componente, se suscribe al servicio para obtener la lista de
    clientes
    this.clientesServicio.getClientes().subscribe(clientes => {
      this.clientes = clientes; // Almacena el listado en la propiedad 'clientes'
    });
  }
}

```

Comentarios importantes del código:

1. **ngOnInit**: Cuando el componente se inicializa, se llama al servicio `ClienteServicio` para obtener la lista de clientes desde Firestore. Se usa `subscribe` para recibir la lista y almacenarla en la propiedad `clientes`.
2. **Standalone Component**: Este componente es **standalone**, lo que significa que no depende de un módulo para ser utilizado. Pero además, por ello debemos importar los módulos que necesitemos como es el módulo de `CommonModule` para poder utilizar el pipe de currency en la plantilla de este componente como sigue.

Código: `clientes.component.html`

Este archivo define la estructura visual del listado de clientes y los botones de acciones.

```

<!-- Botones -->
<section id="actions" class="py-4 mb-4">
  <div class="container">
    <div class="row">
      <div class="col-md-3">

```

```
<a href="#" class="btn btn-primary w-100"
    data-bs-toggle="modal"
    data-bs-target="#agregarClienteModal">
    <i class="bi bi-plus-circle"></i> Agregar Cliente
</a>
</div>
</div>
</div>
</div>
</section>

<!-- Listado Clientes -->
<section id="clientes">
    <div class="container">
        <div class="row">
            <div class="col-md-9">
                <div class="card">
                    <div class="card-header">
                        <h4>Listado de Clientes</h4>
                    </div>
                    <table class="table table-striped">
                        <thead class="table-dark">
                            <tr>
                                <th>Nombre</th>
                                <th>Saldo</th>
                            </tr>
                        </thead>
                        <tbody>
                            <!-- Iteración sobre la lista de clientes -->
                            @for (cliente of clientes; track cliente) {
                                <tr>
                                    <td>{{cliente.nombre}} {{cliente.apellido}}</td>
                                    <td>{{cliente.saldo | currency: 'MXN': '$'}}</td>
                                </tr>
                            }
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</section>
```

Comentarios importantes del código:

1. **Botón "Agregar Cliente"**: Abrirá una ventana modal para agregar un nuevo cliente. Posteriormente agregaremos la ventana modal con el ID `agregarClienteModal`, de momento aún no la hemos definido.
2. **Tabla del Listado de Clientes**: Se utiliza un bucle `@for` para iterar sobre la lista de clientes y mostrar sus nombres, apellidos y saldo. El pipe `currency` se emplea para formatear el saldo en moneda **MXN** (pesos mexicanos), sin embargo puedes adaptarla a la moneda que necesites.
3. **Icono de Bootstrap**: El icono `bi-plus-circle` es un ícono de **Bootstrap Icons**, que mejora la interfaz visual.

Resultado Final hasta el momento

The screenshot shows a dark-themed web application. At the top, there's a header bar with 'Control Clientes' on the left, 'Inicio' and 'Bienvenido' on the right, and a user icon. Below the header is a bright blue navigation bar with the title 'Control de Clientes' and a gear icon. A blue button labeled '+ Agregar Cliente' is positioned below the title. The main content area has a dark background and features a table titled 'Listado de Clientes'. The table has two columns: 'Nombre' and 'Saldo'. It contains three rows with data: 'Juan Perez' and '\$100.00'; 'Karla Gomez' and '\$200.00'. At the bottom of the page, a yellow footer bar displays the copyright notice '© GlobalMentoring.com.mx'.

Conclusión

Con esto ya tenemos el listado de clientes de nuestra aplicación.

¡Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)