

Componente Agregar Cliente – parte 3



Agregar un Cliente – Parte 3

En esta lección, vamos finalmente a guardar el nuevo cliente en la base de datos de **Cloud Firestore**. Para ello vamos a modificar el componente de clientes y también el servicio de clientes.

Código del archivo `clientes.service.ts`

```
import { Injectable } from '@angular/core';
import { Cliente } from '../modelo/cliente.modelo';
import { Firestore, collection, collectionData, addDoc } from '@angular/fire/firestore';
import { Observable } from 'rxjs';
import { query, orderBy } from 'firebase/firestore';

@Injectable({
  providedIn: 'root',
})
export class ClienteServicio {
  private clientesRef: any; // Definir la referencia aquí
  clientes: Observable<Cliente[]>

  constructor(private firestore: Firestore) {
    // Inicializar la referencia a la colección 'clientes' dentro del constructor
  }
}
```

```

this.clientesRef = collection(this.firestore, 'clientes');

// Crear una consulta para ordenar los clientes por nombre ascendente
const consulta = query(this.clientesRef, orderBy('nombre', 'asc'));

// Obtener los datos de la consulta como un Observable
this.clientes = collectionData(consulta, { idField: 'id' }) as Observable<Cliente[]>;
}

// Método para obtener los clientes
getClientes(): Observable<Cliente[]> {
  return this.clientes;
}

// Método para agregar un cliente
agregarCliente(cliente: Cliente) {
  // Reutilizamos la referencia 'clientesRef'
  return addDoc(this.clientesRef, cliente);
}
}

```

Código del archivo `clientes.component.ts`

```

import { Component, ElementRef, ViewChild } from '@angular/core';
import { Cliente } from '../../../../../modelo/cliente.modelo';
import { ClienteServicio } from '../../../../../servicios/cliente.service';
import { CommonModule } from '@angular/common';
import { RouterModule } from '@angular/router';
import { FormsModule, NgForm } from '@angular/forms';

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

```

```
// ¡Usamos '!' para decirle a TypeScript que será inicializada más tarde!
@ViewChild('botonCerrar') botonCerrar!: ElementRef;

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'
  });
}

getSaldoTotal(): number {
  return this.clientes?.reduce((total, cliente) => total + (cliente.saldo ?? 0), 0) ?? 0;
}

agregar(clienteForm: NgForm) {
  const { value, valid } = clienteForm;

  if (valid) {
    // Aquí puedes agregar la lógica para guardar el cliente
    this.clientesServicio.agregarCliente(value);
    // Limpiamos el formulario para agregar un nuevo cliente
    clienteForm.resetForm();
    this.cerrarModal();
  }
}

private cerrarModal(){
  this.botonCerrar.nativeElement.click();
}
}
```

Código del archivo `clientes.component.html`

Se agregó la referencia de `#botonCerrar` en el botón del formulario modal.

```
<!-- 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>
</section>

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

<!-- Tarjetas laterales -->

```

<div class="col-md-3">
    <div class="card text-center bg-danger text-white mb-3">
        <div class="card-body">
            <h3>Saldo Total</h3>
            <h4 class="display-4">
                {{getSaldoTotal() | currency:'MXN':'$'}}
            </h4>
        </div>
    </div>
    <div class="card text-center bg-success text-white mb-3">
        <div class="card-body">
            <h3>Total Clientes</h3>
            <h4 class="display-4">
                <i class="bi bi-people-fill"></i> {{clientes?.length}}
            </h4>
        </div>
    </div>
</div>
</div>
</section>


<div class="modal fade" id="agregarClienteModal" tabindex="-1">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            
            <div class="modal-header bg-info text-white">
                <h5 class="modal-title">Agregar Cliente</h5>
                <button type="button" class="btn-close"
                    data-bs-dismiss="modal" aria-label="Close" #botonCerrar></button>
            </div>

            <form #clienteForm="ngForm" (ngSubmit)="agregar(clienteForm)">
                <div class="modal-body">
                    
                    <div class="row mb-3">
                        <label for="nombre" class="col-form-label col-sm-3">Nombre</label>
                        <div class="col-sm-9">
                            <input type="text" id="nombre" name="nombre"
                                class="form-control" required minlength="2"
                                [(ngModel)]="cliente.nombre" #nombre="ngModel" />
                            @if (nombre.invalid && nombre.touched) {
                                <div class="text-danger">
                                    Nombre es requerido y debe tener al menos 2 caracteres.
                            </div>
                        </div>
                    </div>
                </div>
            </form>
        </div>
    </div>
</div>

```

```
</div>
}
</div>
</div>

<!-- Campo de Apellido -->
<div class="row mb-3">
    <label for="apellido" class="col-form-label col-sm-3">Apellido</label>
    <div class="col-sm-9">
        <input type="text" id="apellido" name="apellido"
            class="form-control" required minlength="2"
            [(ngModel)]="cliente.apellido" #apellido="ngModel" />
        @if (apellido.invalid && apellido.touched) {
            <div class="text-danger">
                Apellido es requerido y debe tener al menos 2 caracteres.
            </div>
        }
    </div>
</div>

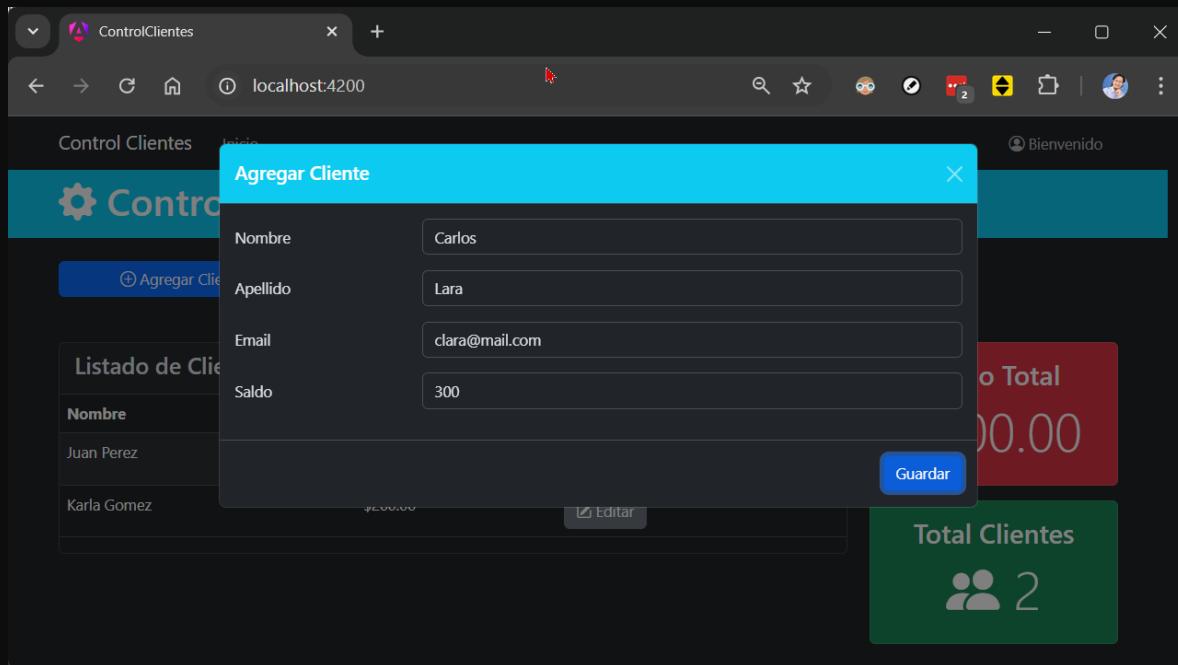
<!-- Campo de Email -->
<div class="row mb-3">
    <label for="email" class="col-form-label col-sm-3">Email</label>
    <div class="col-sm-9">
        <input type="email" id="email" name="email"
            class="form-control" required email
            [(ngModel)]="cliente.email" #email="ngModel" />
        @if (email.invalid && email.touched) {
            <div class="text-danger">
                Por favor ingresa un email válido.
            </div>
        }
    </div>
</div>

<!-- Campo de Saldo -->
<div class="row mb-3">
    <label for="saldo" class="col-form-label col-sm-3">Saldo</label>
    <div class="col-sm-9">
        <input type="number" id="saldo" name="saldo"
            class="form-control" required min="0.01"
            step="0.01" [(ngModel)]="cliente.saldo" #saldo="ngModel" />
        @if (saldo.invalid && saldo.touched) {
            <div class="text-danger">
                El saldo es requerido y debe ser un número positivo.
            </div>
        }
    </div>
</div>
```

```
        </div>
    }
</div>
</div>

<!-- Footer del modal con el botón de guardar --&gt;
&lt;div class="modal-footer"&gt;
    &lt;button type="submit" class="btn btn-primary" [disabled]="clienteForm.invalid"&gt;
        Guardar
    &lt;/button&gt;
&lt;/div&gt;
&lt;/form&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;</pre>
```

Resultado Final hasta el momento



Conclusión

Con estos cambios ya hemos terminado el caso de agregar un nuevo cliente por completo. Ya estamos listos para editar un cliente ya existente.

Recuerda que tienes disponible el proyecto con todo el código completamente funcional de esta lección en caso de que lo necesites.

¡Sigue adelante con tu aprendizaje  , el esfuerzo vale la pena!

Saludos 

Ing. Ubaldo Acosta

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