UNIVERSIDAD NACIONAL DE TRUJILLO

UNT

APLICACIONES MÓVILES

CRUD DE CONTACTOS CON IMÁGENES USANDO SUPABASE

INTEGRANTES:

PINGO AGUILAR CARLOS ALBERTO

Docente:

Sanchez Ticona, Robert Jerry

Trujillo – Perú

2023

CRUD DE CONTACTOS CON IMÁGENES – GRUPO 1

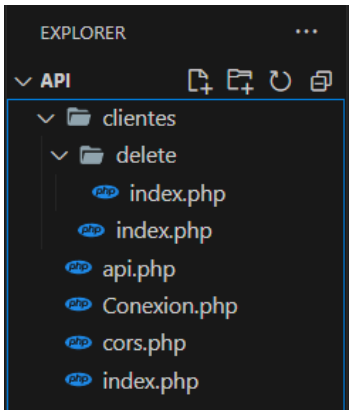
Crear base de datos: bdionic

Crear la tabla clientes y sus campos:

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	cliente_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	ruc_dni	varchar(11)	utf8mb4_general_ci		No	None			Change Drop More
3	nombres	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More
4	apellidos	varchar(80)	utf8mb4_general_ci		No	None			Change Drop More
5	email	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
6	direccion	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
7	url_image	longtext	utf8mb4_general_ci		No	None			Change Drop More
8	estado	tinyint(4)			No	None			Change Drop More

Creamos la carpeta API , donde seguiremos la siguiente estructura :



En el archivo conexion.php se define el método getConexion para poder establecer conexi[on con la base de datos.

```php
<?php
class Conexion {
    public function getConexion(){
        try {
            $host="localhost";
            $db="bdionic";
            $user="root";
            $password="";
            $db = new PDO("mysql:host=$host;dbname=$db;", $user, $password);
            return $db;
        }catch(PDOException $e){
            echo "!Error BD!" . $e->getMessage()."<br/>";
            die();
        }
    }
}
?>
```

Contenido del archivo cors.php

```php
<?php
    header("Access-Control-Allow-Origin: *");
    header("Access-Control-Allow-Headers: X-API-KEY, Origin,Authorization, X-
Requested-With, Content-type, Accept,Access-Control-Request-Method");
    header("Access-Control-Allow-Methods: GET,PUT,POST,DELETE");
    header("Allow: GET,POST,OPTIONS,PUT,DELETE");
?>
```

Contenido de api.php

```php
<?php
class Api {
    public function getClientes(){
        $list = array();
        $conexion = new Conexion();
        $db = $conexion->getConexion();
        $sql = "SELECT * FROM Clientes where estado='1'";
        $query = $db->prepare($sql);
        $query->execute();
        while($row = $query->fetch()){
            $list[]=array(
                "cliente_id" => $row['cliente_id'],
                "ruc_dni" => $row['ruc_dni'],
                "nombres" => $row['nombres'],
                "apellidos" => $row['apellidos'],
                "email" => $row['email'],
                "direccion" => $row['direccion'],
                "url_image" => $row["url_image"]
            );
        }
        return $list;
    }

    public function AddClientes($data){
        $ruc_dni = $data['ruc_dni'];
        $nombres = $data['nombres'];
        $apellidos = $data['apellidos'];
        $email = $data['email'];
        $direccion = $data['direccion'];
        $url_image = $data['url_image'];
        $conexion = new Conexion();
        $db=$conexion->getConexion();

        if (!empty($data['cliente_id'])) {
            $id = $data['cliente_id'];
```

```
            $sql = "update clientes set ruc_dni = :ruc_dni, nombres =
:nombres, apellidos = :apellidos, email = :email, direccion = :direccion,
url_image =:url_image where cliente_id = :id";
            $query=$db->prepare($sql);
            $query->bindParam(':ruc_dni',$ruc_dni);
            $query->bindParam(':nombres',$nombres);
            $query->bindParam(':apellidos',$apellidos);
            $query->bindParam(':email',$email);
            $query->bindParam(':direccion',$direccion);
            $query->bindParam('url_image',$url_image);
            $query->bindParam(':id', $id);
            $query->execute();

            return '{"msg":"Registro Actualizado"}';
        }else{
            $sql = "INSERT INTO clientes
            (ruc_dni,nombres,apellidos,email,direccion,url_image,estado)
values
            (:ruc_dni,:nombres,:apellidos,:email,:direccion,:url_image,'1')";
            $query=$db->prepare($sql);
            $query->bindParam(':ruc_dni',$ruc_dni);
            $query->bindParam(':nombres',$nombres);
            $query->bindParam(':apellidos',$apellidos);
            $query->bindParam(':email',$email);
            $query->bindParam(':direccion',$direccion);
            $query->bindParam('url_image',$url_image);
            $query->execute();
            return '{"msg":"Registro Agregado"}';
        }

    }

    public function getClienteId($data){
        $id=$data['id'];
        $list=array();
        $conexion = new Conexion();
        $db=$conexion->getConexion();
        $sql = "SELECT * FROM Clientes where cliente_id=:id";
        $query = $db->prepare($sql);
        $query->bindParam(':id',$id);
        $query->execute();
        while($row = $query->fetch()){
            $list[]=array(
                "cliente_id" => $row['cliente_id'],
                "ruc_dni" => $row['ruc_dni'],
                "nombres" => $row['nombres'],
                "apellidos" => $row['apellidos'],
                "email" => $row['email'],
                "direccion" => $row['direccion'],
```

```
                "url_image" => $row["url_image"]
            );
        }
        return $list[0];
    }

    public function deleteClient($data){
        $id=$data['cliente_id'];
        $conexion = new Conexion();
        $db=$conexion->getConexion();

        $sql = "update clientes set estado = '0' where cliente_id = :id";
        $query = $db->prepare($sql);
        $query->bindParam(':id', $id);
        $query->execute();

        return '{"msg":"Cliente Eliminado"}';
    }

}
?>
```

Contenido del archivo clientes/index.php.

```
<?php
require_once("../cors.php");
require_once("../api.php");
//require_once("authcontroller.php");
require_once("../conexion.php");

$url=$_SERVER['REQUEST_URI'];
$methodHTTP=$_SERVER['REQUEST_METHOD'];

if($url=='/api/clientes/' && $methodHTTP=='GET'){
    $clientes=array();
    $api= new Api();
    $clientes= $api->getClientes();
    echo json_encode($clientes);
}

if($url == '/api/clientes/' && $methodHTTP == 'POST') {
    $data = $_POST;
    $api= new Api();
    $result = $api->AddClientes($data);
    echo $result;
}
```

```php
if(!empty($_GET['id']) && $methodHTTP=='GET'){
    $data=$_GET;
    $api= new Api();
    $result= $api->getClienteId($data);
    echo json_encode($result);
}

if(!empty($_GET['id']) && $methodHTTP=='POST'){

}
?>
```

Contenido del archivo clientes/delete/index.php.

```php
<?php
    require_once("../../api.php");
    require_once("../../conexion.php");
    require_once("../../cors.php");

    header('Content-Type: application/json; charset=utf-8');
    $url=$_SERVER['REQUEST_URI'];
    $methodHTTP=$_SERVER['REQUEST_METHOD'];

    if($url=='/api/clientes/delete/' && $methodHTTP =='GET'){
        echo "delete";
    }

    if($url == '/api/clientes/delete/' && $methodHTTP == 'POST') {
        $data = $_POST;
        if (!empty($data['cliente_id'])) {
            $api = new Api();
            $result = $api->deleteClient($data);
            echo $result;
        } else {
            echo '{"msg":"Cliente ID no proporcionado"}';
        }
    }
?>
```

CREANDO EL PROYECTO EN IONIC

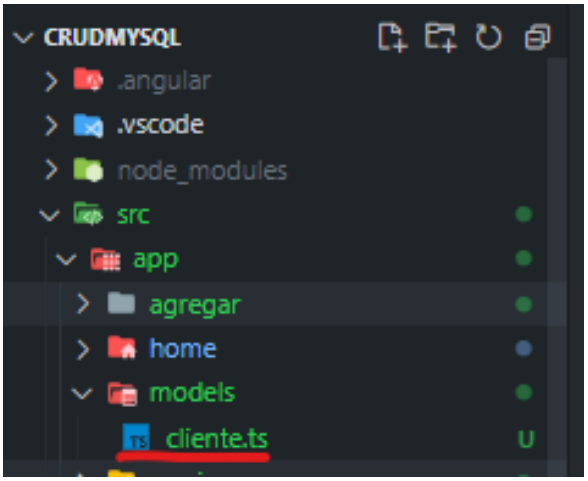
Para crear un proyecto en Ionic se hace uso del comando:

- ionic start crudmysql blank

Para iniciar el proyecto:

- ionic server

En la carpeta app creamos una carpeta models y dentro un archivo cliente.ts



Contenido del archivo cliente.ts

```
export interface Cliente {
    cliente_id: number;
    ruc_dni: string;
    nombres: string;
    apellidos: string;
    email: string;
    direccion: string;
    url_image:string;
}
```

Luego creamos un archivo de services con el siguiente comando:

- ionic g s services\cliente

  services/cliente.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Cliente } from '../models/cliente';

@Injectable({
  providedIn: 'root'
})
export class ClienteService {

  private url = 'http://localhost/api/clientes/'

  constructor(private http:HttpClient) { }

  public getClients(){
    return this.http.get<Cliente[]>(this.url)
  }

  public getClientById(id:any){
    return this.http.get<Cliente>(`${this.url}?id=${id}`)
  }

  public saveCliente(cliente:Cliente){
    const formData = new FormData();
    formData.append('nombres',cliente.nombres);
    formData.append('apellidos',cliente.apellidos);
    formData.append('ruc_dni',cliente.ruc_dni);
    formData.append('direccion',cliente.direccion);
    formData.append('email',cliente.email);
    formData.append('url_image',cliente.url_image);

    return this.http.post(this.url,formData);
  }

  public editCliente(cliente:Cliente){
    const formData = new FormData();
    formData.append('cliente_id', cliente.cliente_id.toString());
    formData.append('nombres',cliente.nombres);
    formData.append('apellidos',cliente.apellidos);
```

```
    formData.append('ruc_dni',cliente.ruc_dni);
    formData.append('direccion',cliente.direccion);
    formData.append('email',cliente.email);
    formData.append('url_image',cliente.url_image);

    return this.http.post(this.url,formData);
  }

  public deleteClient(cliente:Cliente){
    const formData = new FormData();
    formData.append('cliente_id', cliente.cliente_id.toString());
    return this.http.post(`${this.url}delete/`, formData)
  }
}
```

El archivo module.app.ts:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';

import { IonicModule, IonicRouteStrategy } from
'@ionic/angular';

import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { HttpClientModule } from '@angular/common/http';
import { FormBuilder } from '@angular/forms';
import { ReactiveFormsModule } from '@angular/forms';

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule, IonicModule.forRoot(),
AppRoutingModule,HttpClientModule,ReactiveFormsModule],
  providers: [FormBuilder,{ provide: RouteReuseStrategy,
useClass: IonicRouteStrategy }],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

En el archivo home.page.ts

```
import { Component, OnInit } from '@angular/core';
import { Cliente } from '../models/cliente';
import { ClienteService } from '../services/cliente.service';
import { ModalController, ToastController, AlertController } from
'@ionic/angular';
import { AgregarPage } from '../agregar/agregar.page';
import { SupabaseService } from '../services/supabase.service';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage implements OnInit {

  clientes: Cliente[];
  data_cliente: any;


  constructor(private service:ClienteService, private
modalCtrl:ModalController,private toast: ToastController, private
alertController: AlertController, private SupabaseService: SupabaseService) {

  }

  ngOnInit(){
    this.getClientes();
  }

  getClientes(){
    this.service.getClients().subscribe(
      (response) => {
        this.clientes=response;
      },
      (error) => {
        console.log(error);
      }
    );
  }

  async getClientById(clienteId: number){
```

```
  return new Promise((resolve, reject)=>{
    this.service.getClientById(clienteId).subscribe(
      response => {
        resolve(response)
      },
      error => {
        reject(error)
      }
    );
  })
}

async agregar(){
  const modal = await this.modalCtrl.create({
    component: AgregarPage
  });

  modal.onDidDismiss().then(res => {
    const role = res.role;
    if(role === 'guardar'){
      this.getClientes();
    }
  });

  return await modal.present();
}

async editar(cliente: Cliente){

  this.data_cliente = await this.getClientById(cliente.cliente_id);

  const modal = await this.modalCtrl.create({
    component: AgregarPage,
    componentProps:{
      edit: true,
      data: this.data_cliente
    }
  });

  modal.onDidDismiss().then(res => {
    const role = res.role;
    if(role === 'editar'){
```

```
        this.getClientes();
      }
    });

    return await modal.present();

  }

  async delete(cliente:Cliente){
    const alert = await this.alertController.create({
      message: `Desea eliminar al cliente: ${cliente.nombres}
${cliente.apellidos}`,
      buttons: [
        {
          text: 'Cancel',
          role: 'cancel',
          handler: () => {
            console.log('Cancelado')
          }
        },
        {
          text: 'Eliminar',
          role: 'eliminar',
          handler: () => {
            const nameImg = this.getNameImage(cliente.url_image);
            this.SupabaseService.deleteImg(nameImg);
            this.service.deleteClient(cliente).subscribe(
              async (response: any) => {
                this.getClientes();
                const data = response;
                const toasController = await this.toast.create({
                  message: data.msg,
                  duration: 500,
                  position: 'bottom',
                  color: 'success',
                });

                await toasController.present();
              }
            );
          }
        },
```

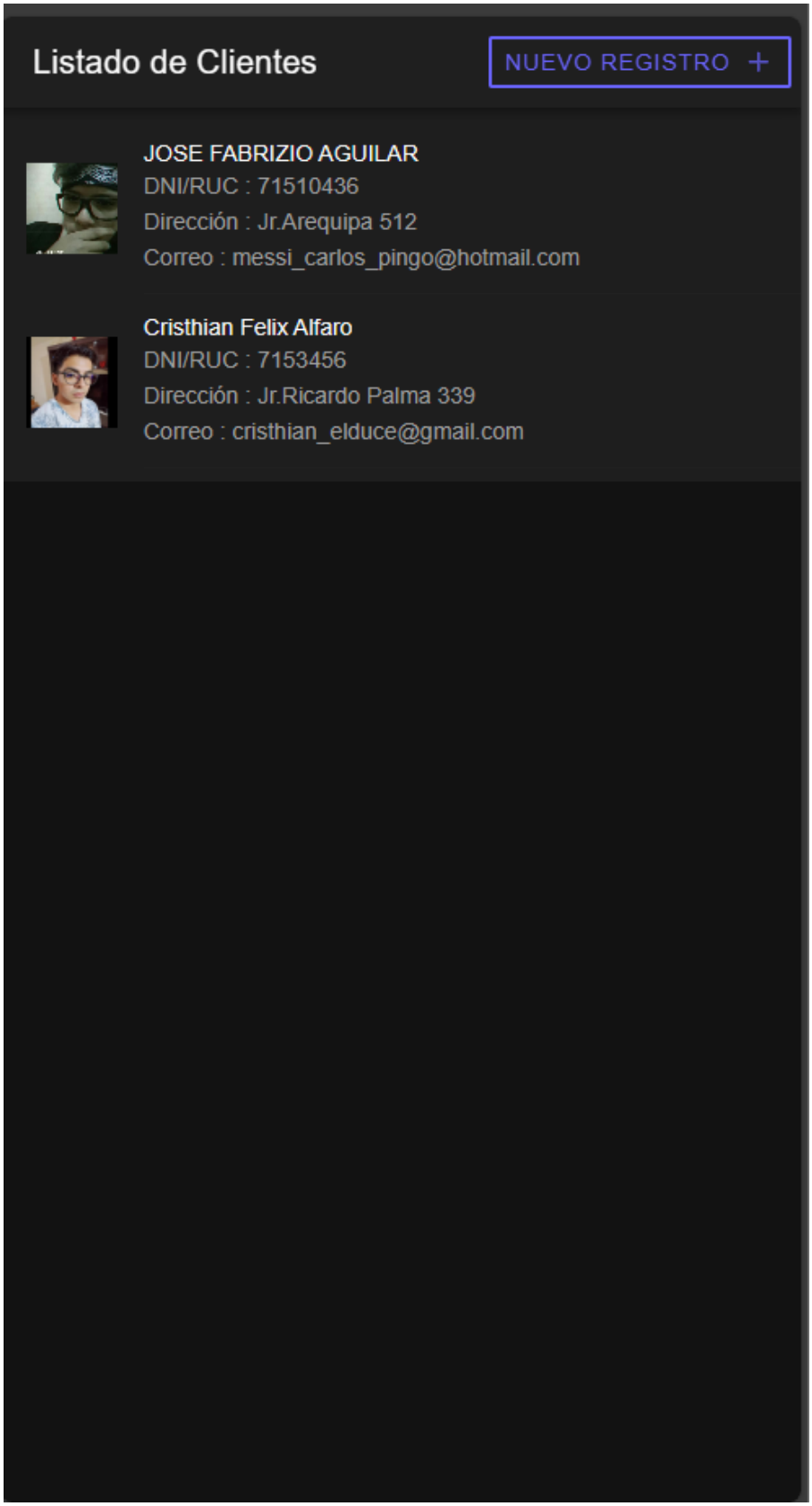
```
      ]
    });

    await alert.present();
  }

  getNameImage(urlImage:String){
    const partesUrl = urlImage.split('/');
    const nombreArchivo = partesUrl[partesUrl.length - 1];
    return nombreArchivo;
  }

}
```

En el archivo home.page.html

```
<ion-header [translucent]="true">
  <ion-toolbar>
    <ion-title>Listado de Clientes</ion-title>
    <ion-buttons slot="primary" (click)="agregar()">
      <ion-button color="tertiary" fill="outline">
      Nuevo Registro
      <ion-icon slot="end" name="add-outline"></ion-icon>
      </ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>

<ion-content [fullscreen]="true">
  <ion-header collapse="condense">
    <ion-toolbar>
      <ion-title size="large">Blank</ion-title>
    </ion-toolbar>
  </ion-header>

  <div>
    <ion-list>
      <ion-item-sliding *ngFor="let cliente of clientes">
        <ion-item>
          <ion-thumbnail slot="start">
```

```
            <img alt="Silhouette of mountains"
src="{{cliente.url_image}}" />
          </ion-thumbnail>
          <ion-label> <h3 class="title">{{ cliente.nombres }} {{
cliente.apellidos }} </h3>
          <p class="subtext">{{'DNI/RUC : ' +
cliente.ruc_dni}}</p>
          <p>{{'Dirección : ' + cliente.direccion}}</p>
          <p>{{'Correo : ' + cliente.email}}</p>
          </ion-label>
        </ion-item>
        <ion-item-options side="end">
          <ion-button color="success" (click)="editar(cliente)">
            <ion-icon name="create"></ion-icon>
            Editar
          </ion-button>
          <ion-button color="danger" (click)="delete(cliente)" >
            <ion-icon name="trash"></ion-icon>
            Eliminar
          </ion-button>
        </ion-item-options>
      </ion-item-sliding>
    </ion-list>
  </div>
</ion-content>
```

Vista de la page home, mostrando el listado de los clientes:

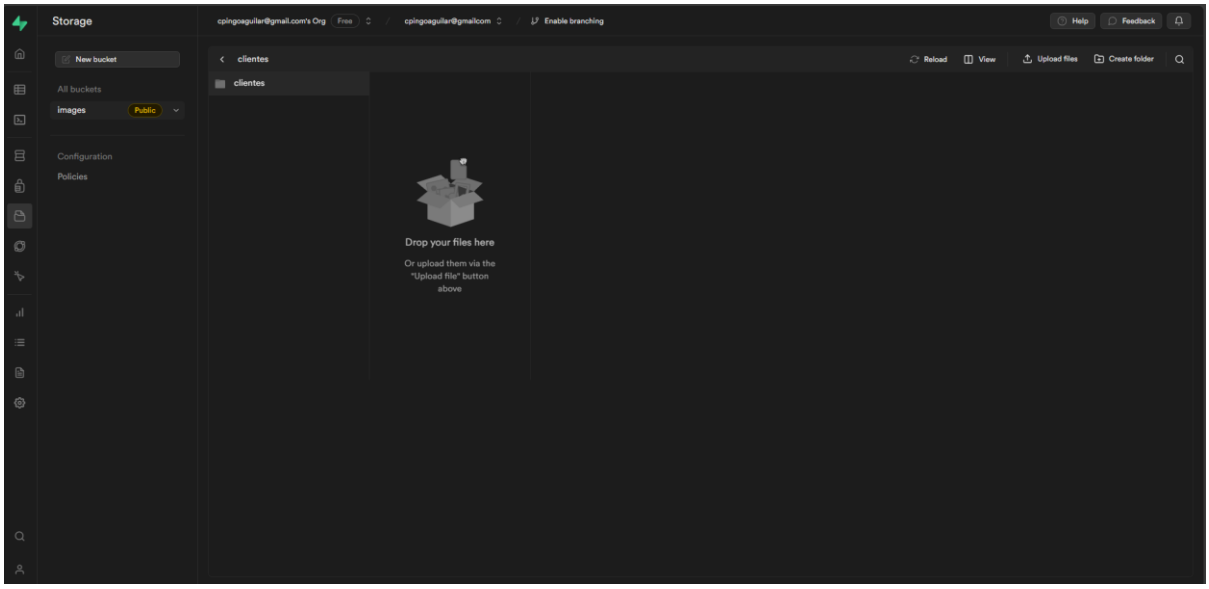


HACIENDO USO DE SUPABASE

Instalamos Supabase en nuestro proyecto ionic

npm install @supabase/supabase-js

Antes de esto tenemos que crear un nuevo proyecto en SupaBase y crear un bucket para poder guardar imágenes



Creamos un nuevo servicio para poder hacer uso de supabase:
ionic g s services\supabase

Para poder usar supabase necesitamos la url de SUPABASE y nuesta API_KEY.

Para subir archivos a tu bucket de almacenamiento en Supabase, puedes utilizar el método upload proporcionado por Supabase.

```
import { Injectable } from '@angular/core';
import { createClient } from '@supabase/supabase-js';

@Injectable({
  providedIn: 'root'
})
export class SupabaseService {

  supabaseUrl = 'https://ajpcvlbafpnwwpdfnmpi.supabase.co';
  supabaseKey =
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsIn
JlZiI6ImFqcGN2bGJhZnBud3dwZGZubXBpIiwicm9sZSI6InNlcnZpY2Vfcm9sZS
```

```
IsImlhdCI6MTcwMzk2OTAyOSwiZXhwIjoyMDE5NTQ1MDI5fQ.PnpbOT2UA4qRNzC
krSdEtI8LxIaShg_lVR79DP3xVEo';

  supabase = createClient(this.supabaseUrl,this.supabaseKey);

  constructor() { }

  async subirArchivo(file: File): Promise<string | null | any> {

    try {
      const { data, error } = await this.supabase.storage.
        from('images/clientes')
        .upload(file.name, file);

      if (error) {
        throw error;
      }
      if (data) {// O puede ser otro campo que contenga la URL
de la imagen
        return data;
      }
      return null;
    } catch (error) {
      console.error('Error al subir la imagen:', error);
      return null;
    }
  }

  async deleteImg(nombreArchivo: string) {
      const { data, error } = await
this.supabase.storage.from('images').remove([`clientes/${nombreA
rchivo}`]);

      if (error) {
          console.error('Error al subir el archivo:', error);
          return;
      }
      console.log('Archivo eliminado exitosamente:', data);
  }
}
```

Creamos la page agregar con el siguiente comando:

ionic g page agregar

El contenido archivo agregar.page.ts , que contiene lo siguiente:

```
import { Component, OnInit, Input, ViewChild } from
'@angular/core';
import { NgForm, FormControl } from '@angular/forms';
import { ClienteService} from '../services/cliente.service';
import { Cliente} from '../models/cliente';
import { ModalController } from '@ionic/angular';
import {Validators, FormBuilder, FormGroup } from
'@angular/forms';
import { NavParams } from '@ionic/angular';
import { SupabaseService } from '../services/supabase.service';


@Component({
  selector: 'app-agregar',
  templateUrl: './agregar.page.html',
  styleUrls: ['./agregar.page.scss'],
})
export class AgregarPage implements OnInit {

  @Input() cliente:Cliente;
  edit: Boolean;
  datos={
    nombres:'',
    apellidos:'',
    ruc_dni:'',
    direccion:'',
    email:''
  }
  previsualizacion: String
  @ViewChild('fileInput') fileInput: any;
  fileProfile: File

  createFormGroup(){
    return new FormGroup({
      cliente_id: new FormControl(''),
      nombres: new
FormControl('',[Validators.required,Validators.minLength(5)]),
```

```
      apellidos: new
FormControl('',[Validators.required,Validators.minLength(5)]),
      ruc_dni: new
FormControl('',[Validators.required,Validators.maxLength(8)]),
      direccion: new
FormControl('',[Validators.required,Validators.maxLength(100)]),
      email: new FormControl('',[Validators.required,
Validators.pattern("^[a-zA-Z0-9._%-]+@[a-zA-Z0-9*-]+.[a-zA-
Z]{2,4}$")]),
      url_image: new FormControl('')
    });

  }

  validation_messages = {
    'nombres': [
      { type: 'required', message: 'Escriba Nombre.' },
      { type: 'minlength', message: 'Nombre maximo de 5
caracteres' }
    ],
    'apellidos': [
      { type: 'required', message: 'Escriba Apellido.' },
      { type: 'minlength', message: 'Apellido maximo de 5
caracteres' }
    ],
    'ruc_dni': [
      { type: 'required', message: 'Escriba RUC/DNI' },
      { type: 'maxlength', message: 'RUC/DNI es de 8 caracteres'
}
    ],
    'direccion': [
      { type: 'required', message: 'Escriba direccion' },
      { type: 'maxlength', message: 'No puede escribir mas de
100 caracteres' }
    ],
    'email': [
      { type: 'required', message: 'Escribir correo' },
      { type: 'pattern', message: 'No es un formato de correo' }
    ]
  }

  get nombres() {
```

```
    return this.registrarForm.get('nombres');
  }
  get apellidos() {
    return this.registrarForm.get('apellidos');
  }
  get ruc_dni(){
    return this.registrarForm.get('ruc_dni');
  }
  get direccion(){
    return this.registrarForm.get('direccion');
  }
  get email(){
    return this.registrarForm.get('email');
  }
  get cliente_id(){
    return this.registrarForm.get('cliente_id');
  }
  get url_image(){
    return this.registrarForm.get('url_image');
  }

  registrarForm : FormGroup;

  constructor(private service:ClienteService, private
modalCtrl:ModalController, public
formBuilder:FormBuilder,private navParams:NavParams, private
supabaseService:SupabaseService) {
    this.registrarForm = this.createFormGroup();
    this.edit = this.navParams.get('edit') || false;
  }

  ngOnInit() {
    this.cliente = this.navParams.get('data');
    if (this.cliente){
      this.edit=true;
      this.registrarForm.setValue({
        cliente_id: this.cliente.cliente_id,
        nombres: this.cliente.nombres,
        apellidos: this.cliente.apellidos,
        ruc_dni: this.cliente.ruc_dni,
        direccion: this.cliente.direccion,
        email: this.cliente.email,
```

```
      url_image:this.cliente.url_image
    });
    this.previsualizacion= this.cliente.url_image;
  }
}

closeModal(){
  this.modalCtrl.dismiss(null,'cerrado');
}

async onFileSelected(oEvent:Event){
  const fileInput = oEvent.target as HTMLInputElement;
  if (fileInput.files && fileInput.files.length > 0) {
    const file = fileInput.files[0];
    this.fileProfile = file
    try{
      const base64String = await this.getBase64(file);
      this.previsualizacion = base64String;
      console.log(this.previsualizacion)
    }catch(error){
      console.error('Error al cargar la imagen:', error);
    }
  }
}


getBase64(file: File): Promise<string> {
  return new Promise((resolve, reject) => {
    const reader = new FileReader();

    reader.onload = (e: any) => {
      const base64String = e.target.result as string;
      resolve(base64String);
    };

    reader.onerror = (error) => {
      reject(error);
    };

    reader.readAsDataURL(file);
  });
}
```

```
  getUrlImage(fullPath:String){
    const urlImage =
`https://ajpcvlbafpnwwpdfnmpi.supabase.co/storage/v1/object/publ
ic/${fullPath}`;
    return urlImage;
  }


  async onSubmit(){
    if(!this.edit){
      const cliente=this.registrarForm.value;
      const url = await
this.supabaseService.subirArchivo(this.fileProfile);
      if(url){
        console.log(url);
        cliente.url_image = this.getUrlImage(url.fullPath);
        this.service.saveCliente(cliente).subscribe(
          response => {
            this.modalCtrl.dismiss(response,'guardar');
            console.log(response);
          }
        );
      }
    }else{
      const cliente=this.registrarForm.value;
      const url = await
this.supabaseService.subirArchivo(this.fileProfile);
      if(url){
        cliente.url_image = this.getUrlImage(url.fullPath);
        this.service.editCliente(cliente).subscribe(
          response => {
            this.modalCtrl.dismiss(response,'editar');
            console.log(response);
          }
        );
      }
    }
  }
}
```

Contenido del archivo agregar.page.html

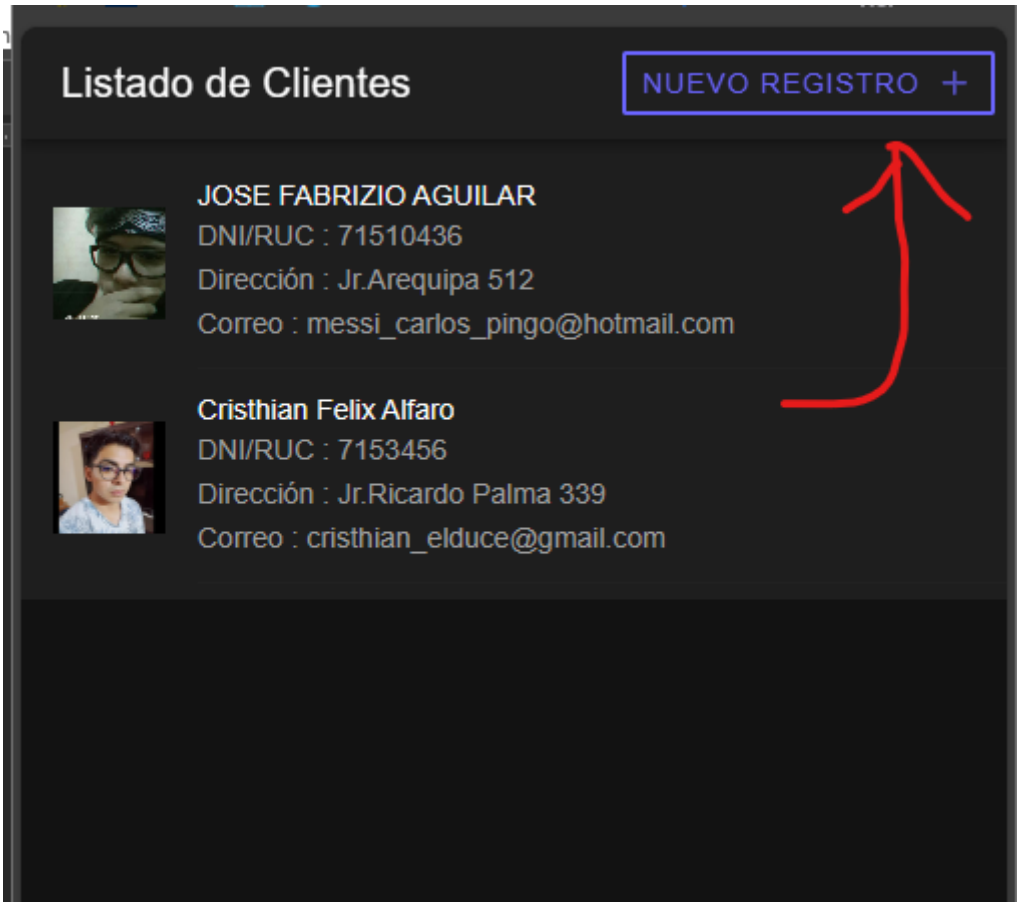
```
<ion-header >
  <ion-toolbar >
    <ion-icon name="close" slot="start" size="medium"
color="medium" class="ion-padding" (click)="closeModal()"></ion-
icon>
    <!-- <ion-button (click) = "cancel()" fill="clear"><ion-icon
name="close"></ion-icon> </ion-button> -->
    <ion-title slot="start">{{ edit ? 'Editar Cliente' :
'Agregar Cliente' }}</ion-title>
    <!-- <ion-icon name="close" slot="end" size="large"
class="ion-padding" (click)="dismiss()"></ion-icon> -->
    <!-- <ion-button slot="end" (click)="onSubmit()"
fill="clear">Guardar</ion-button> -->
  </ion-toolbar>
</ion-header>


<ion-content class="ion-padding">
  <form [formGroup]="registrarForm" (ngSubmit)="onSubmit()">

    <ion-list lines="full" class="ion-no-margin ion-no-padding">

      <!-- Cliente_id -->
      <ion-item style="display: none;">
        <ion-input label="Cliente" labelPlacement="floating"
type="text" formControlName="cliente_id"></ion-input>
      </ion-item>

      <!-- Nombres -->
      <ion-item>
        <ion-input label="Nombres:" labelPlacement="floating"
formControlName="nombres" placeholder="Ingrese nombres"></ion-
input>
        <ion-text *ngIf="nombres?.invalid && (nombres?.dirty ||
nombres?.touched)" color="danger">
          <p *ngFor="let error of validation_messages.nombres"
[hidden]="!(nombres && nombres.hasError(error.type))">{{
error.message }}</p>
        </ion-text>
      </ion-item>
```

```
      <!-- Apellidos -->
      <ion-item>
        <ion-input label="Apellidos:" labelPlacement="floating"
formControlName="apellidos" placeholder="Ingrese
apellidos"></ion-input>
        <ion-text *ngIf="apellidos?.invalid && (apellidos?.dirty
|| apellidos?.touched)" color="danger">
          <p *ngFor="let error of validation_messages.apellidos"
[hidden]="!(apellidos && apellidos.hasError(error.type))">{{
error.message }}</p>
        </ion-text>
      </ion-item>

      <!-- RUC/DNI -->
      <ion-item>
        <ion-input label="RUC/DNI:" labelPlacement="floating"
formControlName="ruc_dni" placeholder="Ingrese RUC/DNI"></ion-
input>
        <ion-text *ngIf="ruc_dni?.invalid && (ruc_dni?.dirty ||
ruc_dni?.touched)" color="danger">
          <p *ngFor="let error of validation_messages.ruc_dni"
[hidden]="!(ruc_dni && ruc_dni.hasError(error.type))">{{
error.message }}</p>
        </ion-text>
      </ion-item>

      <!-- Dirección -->
      <ion-item>
        <ion-input label="Dirección:" labelPlacement="floating"
formControlName="direccion" placeholder="Ingrese
dirección"></ion-input>
        <ion-text *ngIf="direccion?.invalid && (direccion?.dirty
|| direccion?.touched)" color="danger">
          <p *ngFor="let error of validation_messages.direccion"
[hidden]="!(direccion && direccion.hasError(error.type))">{{
error.message }}</p>
        </ion-text>
      </ion-item>

      <!-- Email -->
      <ion-item>
```

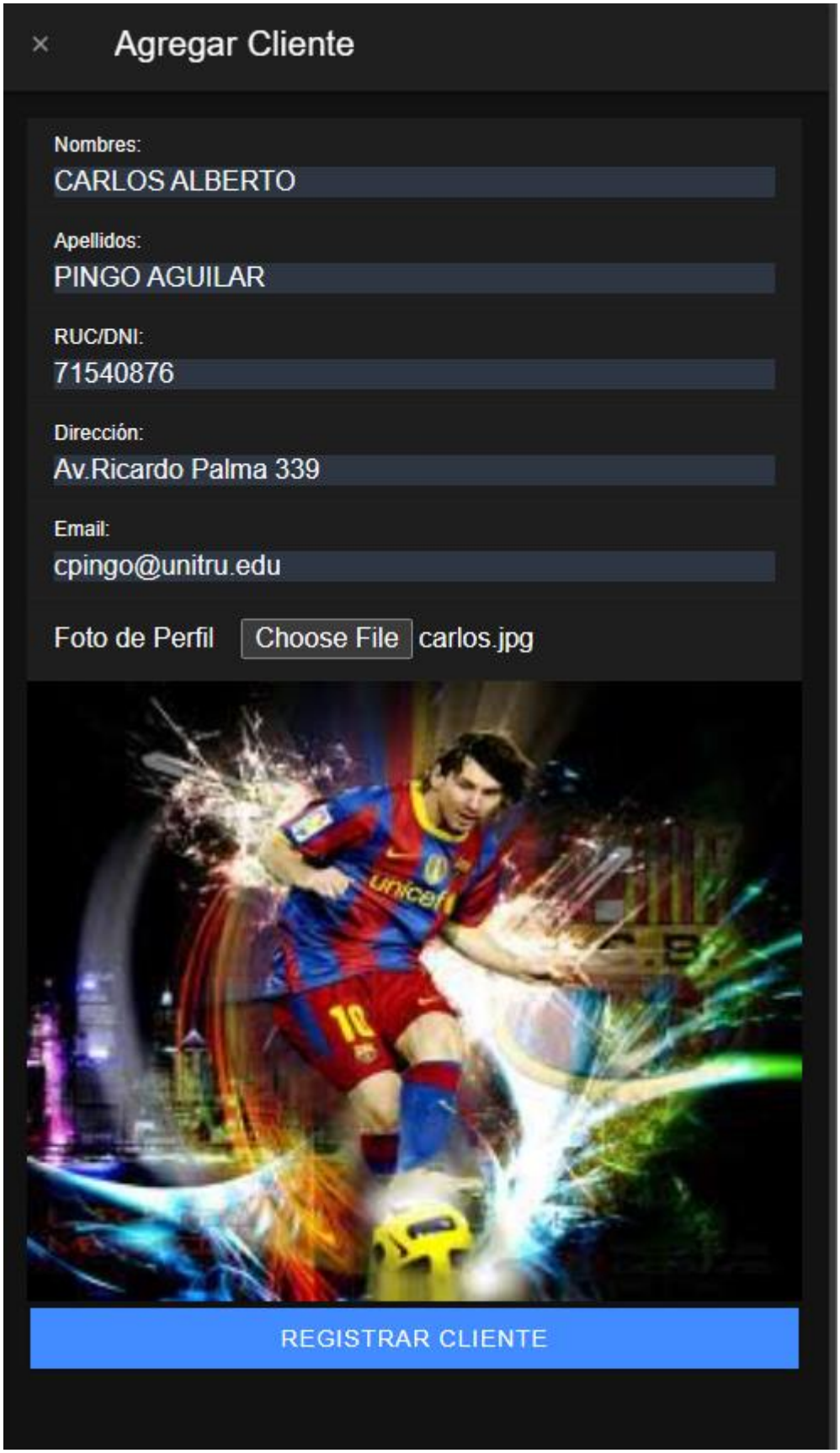
```
        <ion-input label="Email:" labelPlacement="floating"
formControlName="email" placeholder="Ingrese correo
electrónico"></ion-input>
        <ion-text *ngIf="email?.invalid && (email?.dirty ||
email?.touched)" color="danger">
          <p *ngFor="let error of validation_messages.email"
[hidden]="!(email && email.hasError(error.type))">{{
error.message }}</p>
        </ion-text>
      </ion-item>

      <!-- Image -->
      <ion-item>
        <ion-input #fileInput label="Foto de Perfil" type="file"
(change)="onFileSelected($event)"
formControlName="url_image"  accept="image/*"></ion-input>
      </ion-item>

      <ion-img *ngIf="previsualizacion"
        [src]="previsualizacion" alt="imagen no cargada"
class="responsive-image"
      ></ion-img>

    </ion-list>

    <!-- Botones de acción -->
    <ion-button expand="full" type="submit"
[disabled]="registrarForm.invalid">
      {{ edit ? 'Guardar Cambios' : 'Registrar Cliente' }}
    </ion-button>

  </form>
</ion-content>
```

Ahora vamos a agregar un nuevo cliente a nuestra lista:

Hacemos un click en el botón **“NUEVO REGISTRO”**



Llenamos el formulario y hacemos clic en REGISTRA CLIENTE:



Listado de Clientes

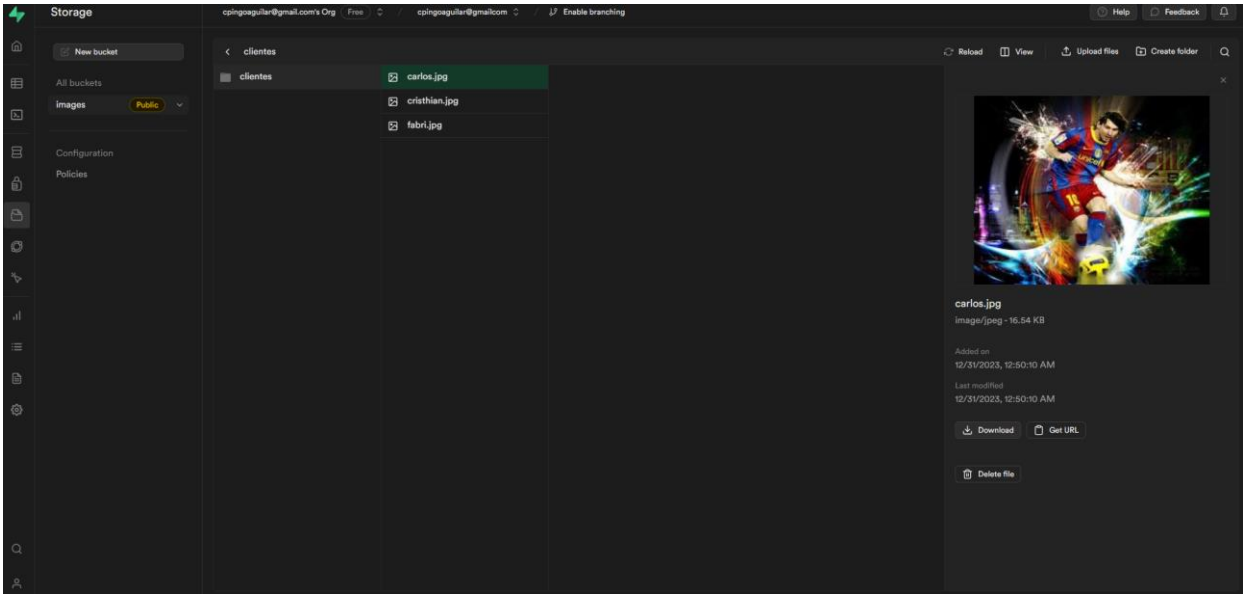
NUEVO REGISTRO +

JOSE FABRIZIO AGUILAR
DNI/RUC : 71510436
Dirección : Jr.Arequipa 512
Correo : messi_carlos_pingo@hotmail.com

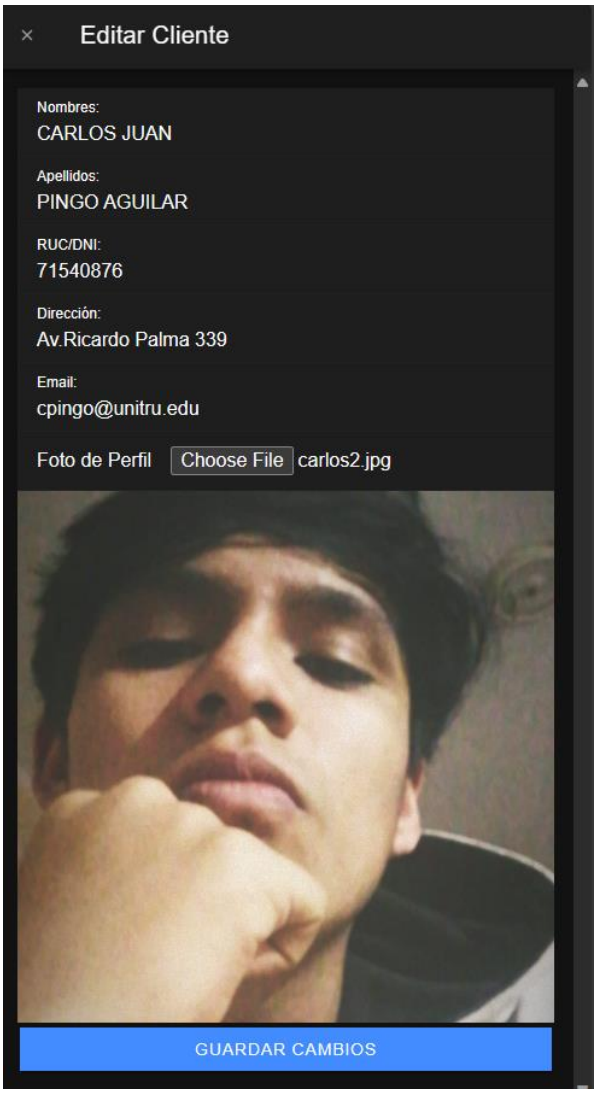
Cristhian Felix Alfaro
DNI/RUC : 7153456
Dirección : Jr.Ricardo Palma 339
Correo : cristhian_elduce@gmail.com

CARLOS ALBERTO PINGO AGUILAR
DNI/RUC : 71540876
Dirección : Av.Ricardo Palma 339
Correo : cpingo@unitru.edu

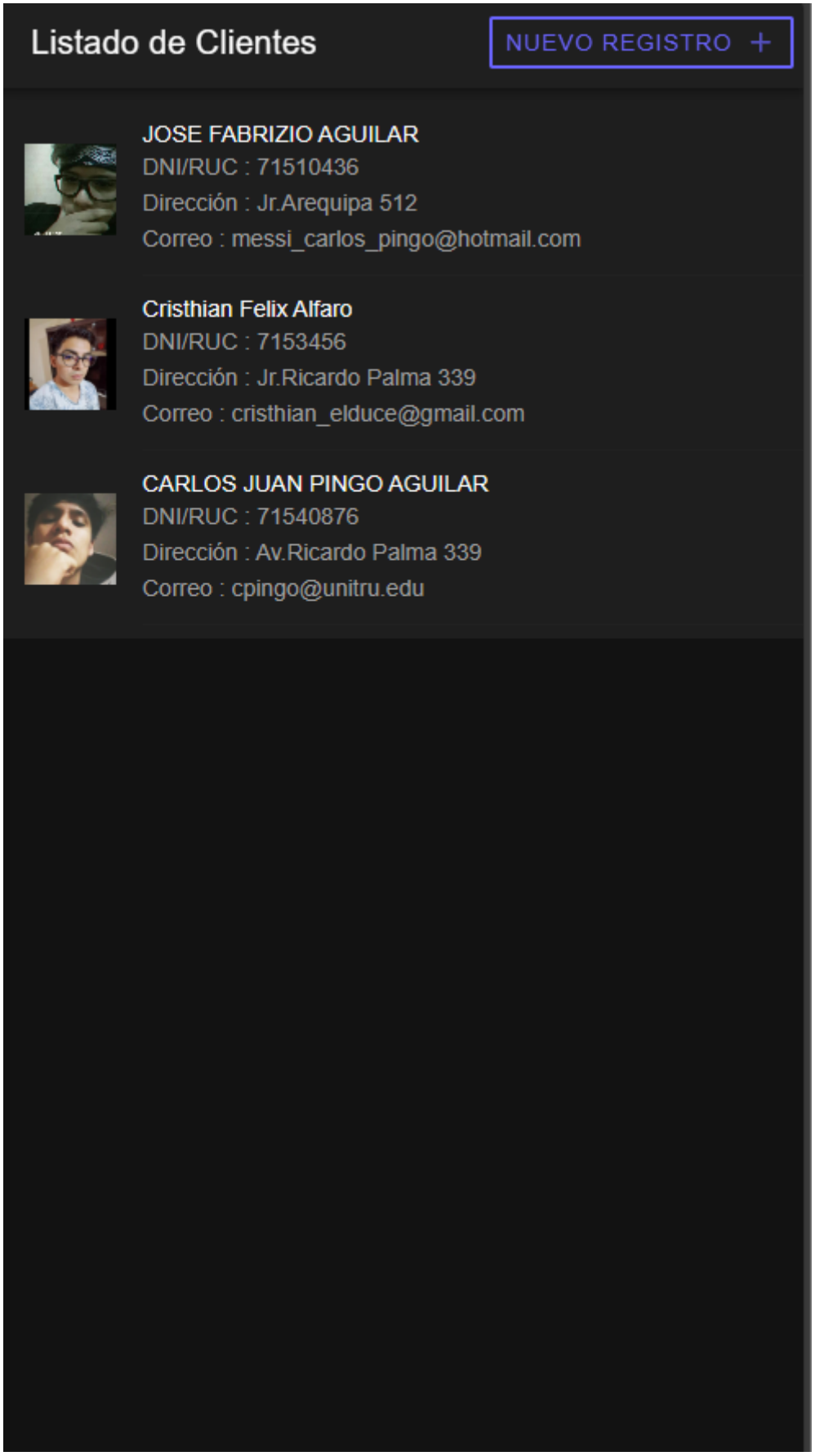
En nuestro bucket, se puede visualizar que se están guardando las imágenes:



Ahora editamos el nombre y la foto de un cliente :

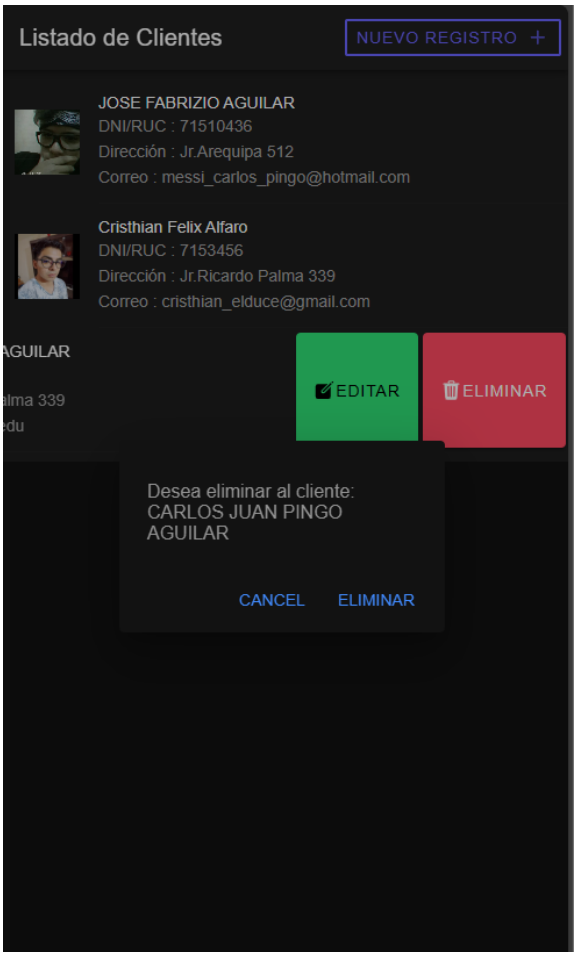


Se muestra el listado de los usuarios actualizados.

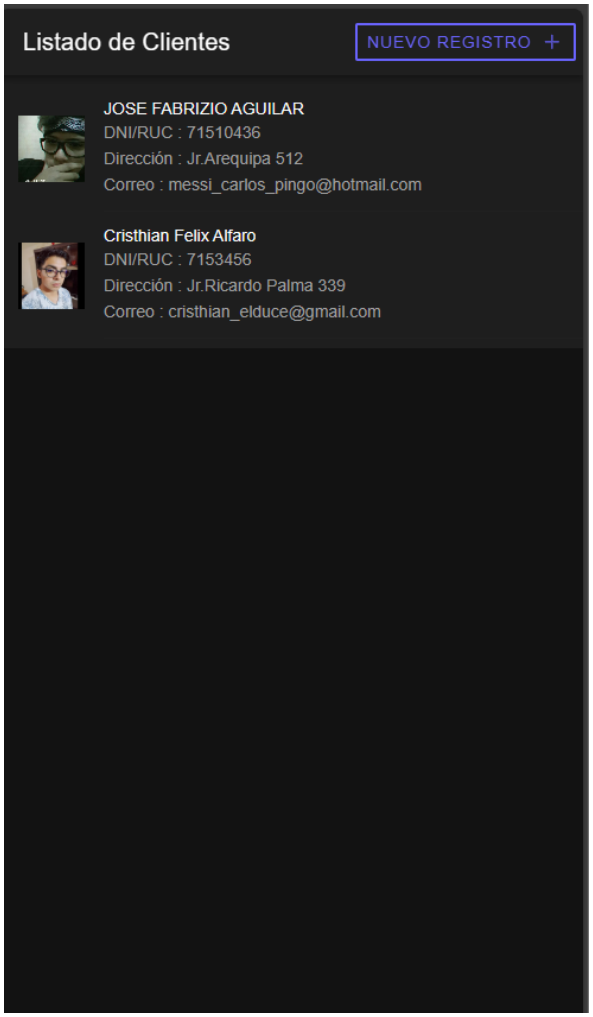
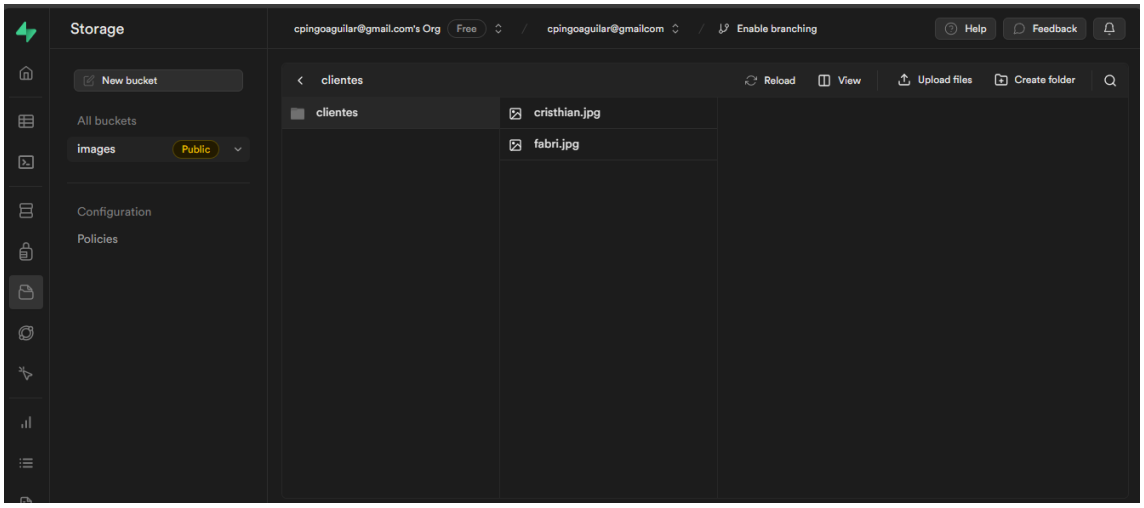


Ahora eliminamos el ultimo usuario agregado :

Se muestra un mensaje para confirmar la eliminación del cliente.



Se muestra la lista de usuarios actualizados.



En Supabase, se elimino la imagen subida por el usuario