

# **DESARROLLO DE LA PROPUESTA DE SOLUCIÓN**

## **Objetivo general**

Reducir la cantidad de desechos orgánicos que se generan en los establecimientos de florerías, fruterías y verdulerías ubicados en la región de Teziutlán, mediante el uso de una aplicación móvil que ayude a contactar a los dueños de dichos establecimientos con recolectores de desechos orgánicos que utilizan residuos orgánicos como abono de cultivo.

## **Objetivos específicos**

Valorar el interés general de los posibles futuros usuarios para usar la aplicación mediante el uso de encuestas.

Desarrollar una aplicación móvil para contactar a los dueños de florerías, fruterías y verdulerías en el municipio de Teziutlán con recolectores de desechos orgánicos utilizan residuos orgánicos como abono de cultivo.

Evaluar si el uso de una aplicación móvil reducirá la cantidad de desechos orgánicos que se generan en la región de Teziutlán.

## **Hipótesis**

El desarrollo de una aplicación móvil hará que los dueños de establecimientos como florerías, fruterías y verdulerías de la región de Teziutlán puedan contactarse con agricultores de la zona así ayudando a reducir tales residuos en los contenedores de dicha región y beneficiando a los agricultores, ya que estos desechos les servirán como abono para sus cultivos.

## Casos de Uso.

<b>RF- 0001</b>	<b>Registro de usuarios</b>	
<b>Versión</b>	V1 - 21 de abril de 2021	
<b>Autores</b>	Julio, Karen , Miguel Alejandro	
<b>Objetivos asociados</b>	Que los usuarios se registren	
<b>Descripción</b>	El usuario se registrará mediante dos vistas principales en la primera se ingresará los siguientes datos: nombre de usuario, contraseña, repetir contraseña, numero de teléfono. En la segunda vista solicitará seleccionar si el usuario será donador de residuos orgánicos o recolector de residuos. Al finalizar se redireccionará a la página de inicio	
<b>Precondición</b>	El usuario debe tener instalada la aplicación	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario , inicia la aplicación
	2	Si es la primera vez que la inicia se mostrara la pantalla de presentación en tal caso el usuario terminara de ver la vista de presentación y llegara a la pantalla de iniciar sesión
	3	Si el usuario no ha iniciado sección y no es la primera vez que usa la aplicación se mostrara la vista de iniciar sesión
	4	Si el usuario ha iniciado sección anterior mente deberá de cerrar su sección para visualizar la vista de iniciar sesión
	5	En la vista iniciar sección indicara que desea registrar una nueva cuenta
	6	En la vista Registro se llenaran todos los datos solicitados
	7	Se seleccionará si el usuario será recolector o donador de residuos para finalizar
<b>Importancia</b>	vital	
<b>Comentarios</b>		

<b>RF- 0002</b>	<b>Iniciar de sección</b>	
<b>Versión</b>	V1 - 21 de abril de 2021	
<b>Autores</b>	Julio, Karen , Miguel Alejandro	
<b>Objetivos asociados</b>	Que los usuarios inicien sección	
<b>Descripción</b>	El usuario iniciara sección con los datos de usuario y contraseña antes creados y almacenados en la base de datos	
<b>Precondición</b>	El usuario debe estar de estar registrado	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario , inicia la aplicación
	2	Si es la primera vez que la inicia se mostrara la pantalla de presentación en tal caso el usuario terminara de ver la vista de presentación y llegara a la pantalla de iniciar sesión
	3	Si el usuario no ha iniciado sección y no es la primera vez que usa la aplicación se mostrara la vista de iniciar sesión
	4	Si el usuario ha iniciado sección anterior mente deberá de cerrar su sección para visualizar la vista de iniciar sesión
	5	En la vista iniciar sección ingresara los datos de usuario y contraseña que ya conoce para ser redireccionado a otra vista
	6	Si el usuario selecciono ser recolector se enviará a la vista lista de recolecciones agendadas a recolectar
	7	Si el usuario selecciono ser donador se enviará a la vista lista recolecciones agendadas por el usuario
<b>Importancia</b>	vital	
<b>Comentarios</b>		

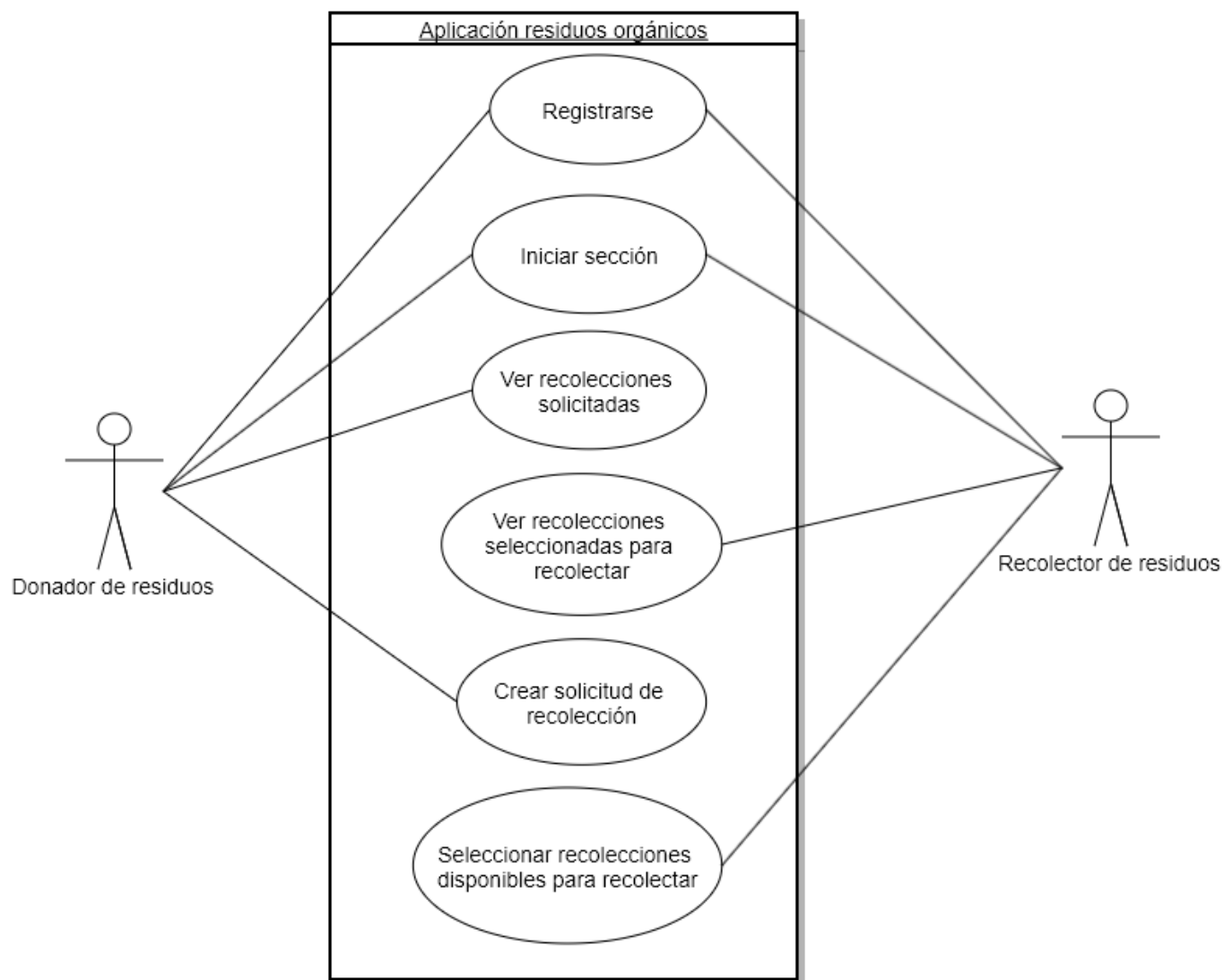
<b>RF- 0003</b>	<b>Ver recolecciones solicitadas</b>	
<b>Versión</b>	V1 - 21 de abril de 2021	
<b>Autores</b>	Julio, Karen , Miguel Alejandro	
<b>Objetivos asociados</b>	Que los usuarios con rol donador visualicen sus recolecciones programadas	
<b>Descripción</b>	Que los usuarios con rol donador visualicen sus recolecciones programadas que se hallan creado con anterioridad las cuales contaran con los datos de dirección, fecha de recolección, límite de recolección, cantidad a recolectar, descripción de la recolección, saber cada que tiempo se repetirá la recolección, botón de eliminar esa recolección	
<b>Precondición</b>	El usuario debe tener su sesión iniciada como donador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que el usuario inicio sección como donador podrá visualizar la información de las recolecciones que ha agendado
	2	Si el usuario lo desea podrá eliminar las recolecciones que ha agendado
<b>Importancia</b>	vital	
<b>Comentarios</b>		

<b>RF- 0004</b>	<b>Ver recolecciones seleccionadas para recolectar</b>	
<b>Versión</b>	V1 - 21 de abril de 2021	
<b>Autores</b>	Julio, Karen , Miguel Alejandro	
<b>Objetivos asociados</b>	Que los usuarios con rol recolector visualicen las recolecciones que seleccionaron para recolectar	
<b>Descripción</b>	Que los usuarios con rol recolector visualicen sus recolecciones que seleccionaron para recolectar de las cuales podrá visualizar la siguiente información: direccion fecha de recolección, límite de recolección, cantidad a recolectar, descripción de la recolección, saber cada que tiempo se repetirá la recolección, botón de eliminar esa recolección de la lista a recolectar, botón para confirmar recolección exitosa	
<b>Precondición</b>	El usuario debe tener su sesión iniciada como recolector	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	Una vez que el usuario inicio sección como recolector podrá visualizar la información de las recolecciones que se a comprometido a recolectar
	2	Si el usuario lo desea podrá deseleccionar las recolecciones que ha agendado
<b>Importancia</b>	vital	
<b>Comentarios</b>		

<b>RF- 0005</b>	<b>Crear solicitud de recolección</b>	
<b>Versión</b>	V1 - 21 de abril de 2021	
<b>Autores</b>	Julio, Karen , Miguel Alejandro	
<b>Objetivos asociados</b>	Que los usuarios inicien como donador	
<b>Descripción</b>	El usuario iniciara sección con los datos de usuario y contraseña antes creados y almacenados en la base de datos	
<b>Precondición</b>	El usuario debe estar de estar registrado, el usuario debe de tener su sección activada teniendo este un rol de donador	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario se encontrará en la vista para visualizar las recolecciones agendadas anteriormente
	2	Presionara el botón para agendar nueva recolección
	3	Se abrirá una nueva vista con un formulario para agendar una nueva recolección
	4	El usuario ingresara su dirección, la fecha de recolección inicial y final ,la cantidad aproximada de desechos que generara , una pequeña descripción opcional sobre la recolección y seleccionara si las recolecciones serán realizadas en un cierto periodo de tiempo
	5	Si los datos ingresados son validos el usuario presionara el botón de agendar recolección
	6	Al presionar el botón de agendar recolección se redireccionará a la pantalla para visualizar las recolecciones agendadas donde podrá ver que se a añadido a la lista su recolección
<b>Importancia</b>	vital	
<b>Comentarios</b>		

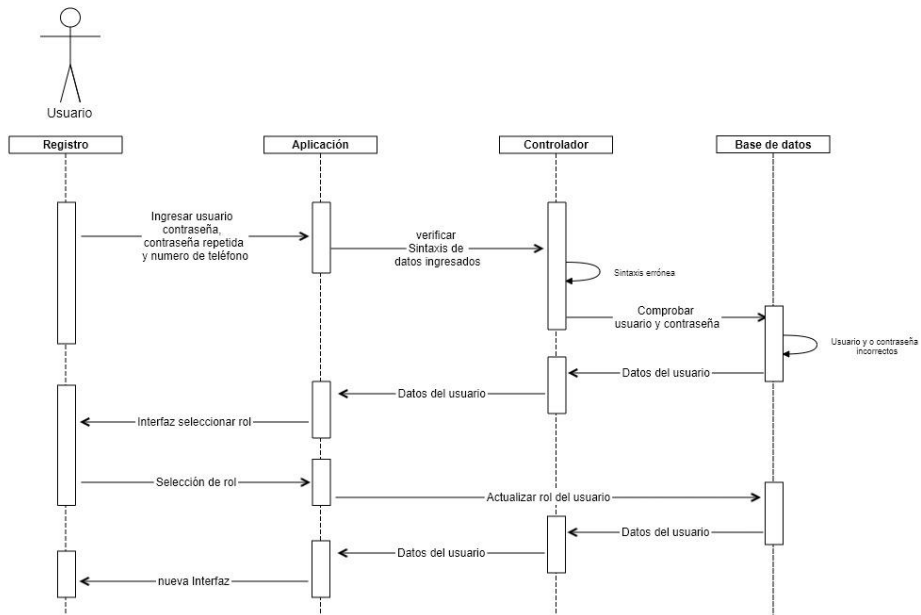
<b>RF- 0006</b>	<b>Seleccionar recolecciones disponibles para recolectar</b>	
<b>Versión</b>	V1 - 21 de abril de 2021	
<b>Autores</b>	Julio, Karen , Miguel Alejandro	
<b>Objetivos asociados</b>	Que los usuarios inicien como recolector	
<b>Descripción</b>	El usuario iniciara sección con los datos de usuario y contraseña antes creados y almacenados en la base de datos	
<b>Precondición</b>	El usuario debe estar de estar registrado, el usuario debe de tener su sección activada teniendo este un rol de recolector	
<b>Secuencia Normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El usuario se encontrará en la vista para visualizar las recolecciones añadidas para recolectar anteriormente
	2	Presionara el botón para agendar nueva recolección
	3	Se abrirá una nueva vista con un formulario para seleccionar las recolecciones para recolectar
	4	En pantalla se mostrará los datos de todas las recolecciones solicitadas por los usuarios que no hallas sido tomadas por otro recolector
	5	El usuario podrá seleccionar una o múltiples recolecciones a la vez de acuerdo a como lo considere
	6	Al presionar el botón de agendar recolección se redireccionará a la pantalla para visualizar las recolecciones agendadas donde podrá ver que se a añadido a la lista su recolección
<b>Importancia</b>	vital	
<b>Comentarios</b>		

## Casos de Uso Diagrama

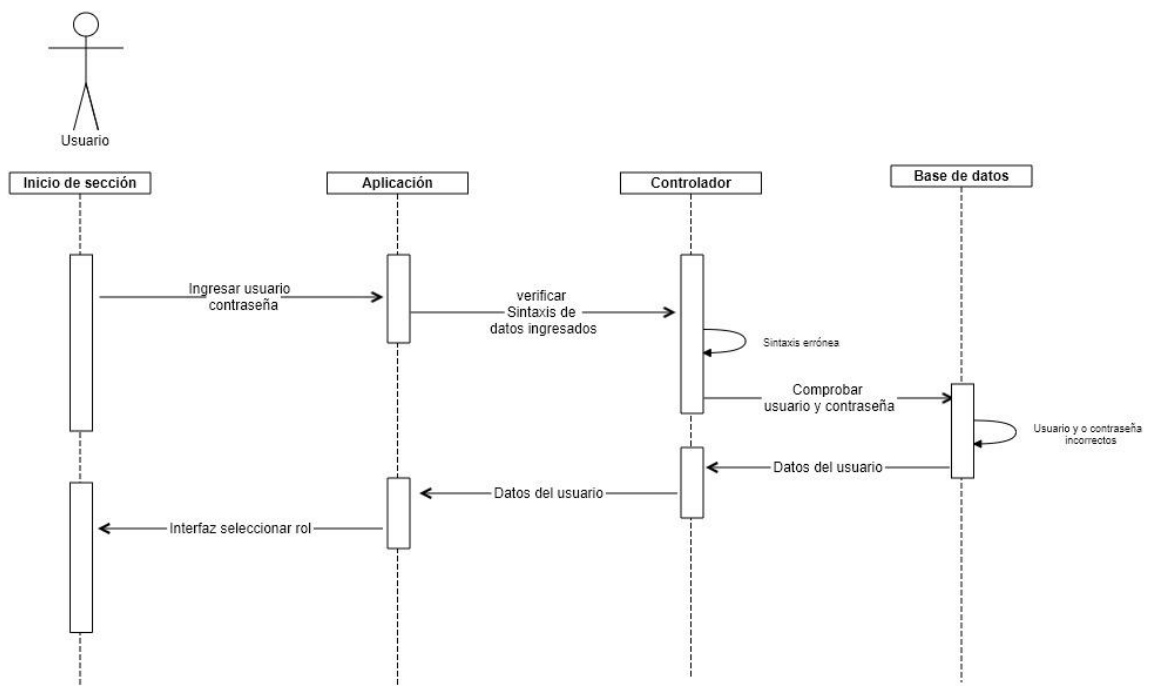


# Diagramas de secuencia

## Registro

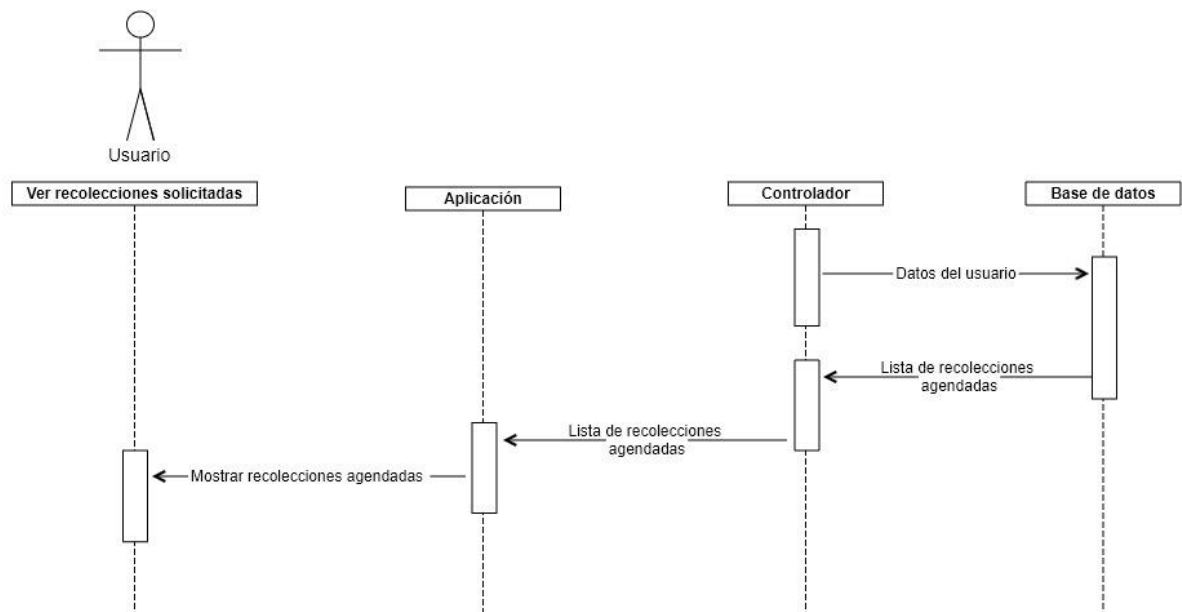


## Inicio de sección

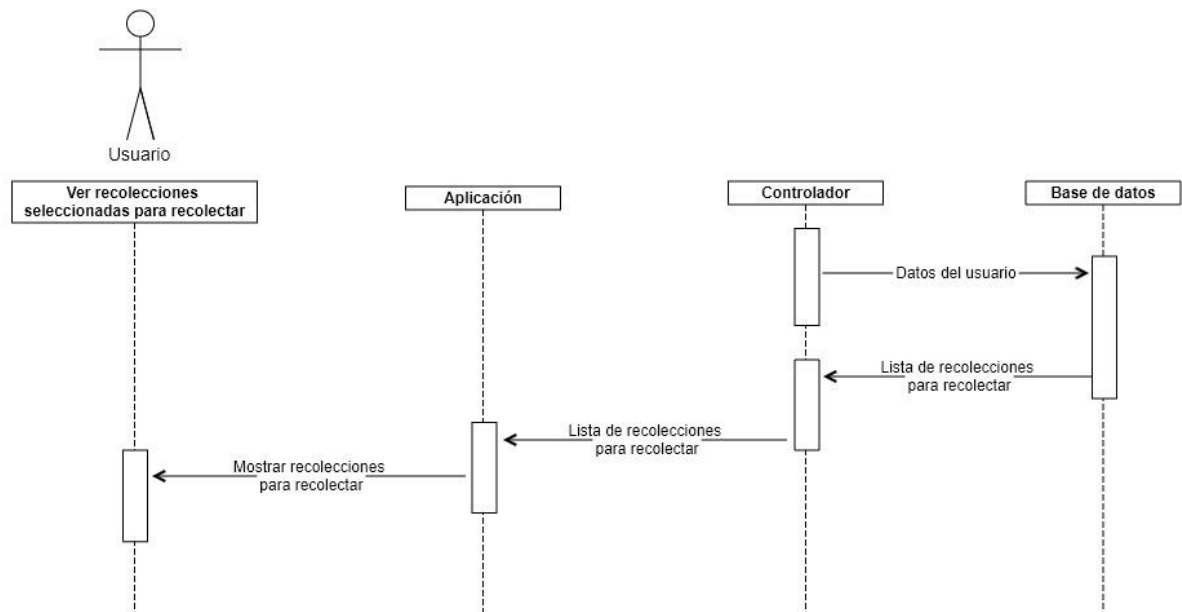




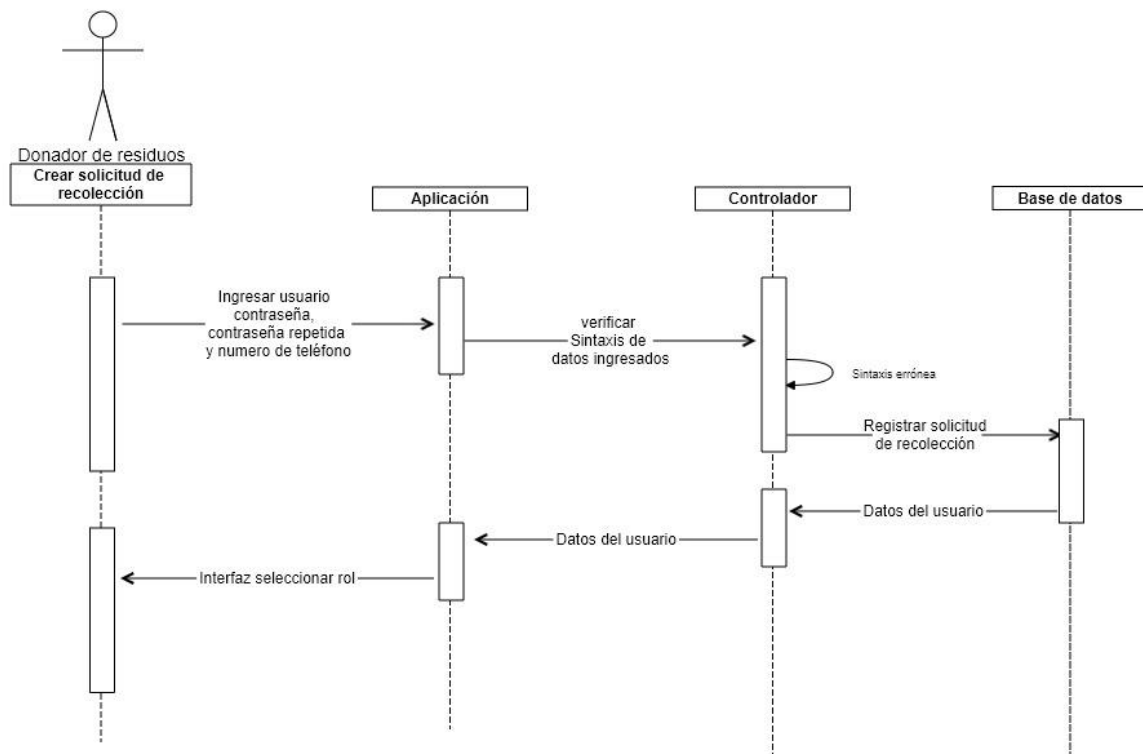
## Ver recolecciones solicitadas



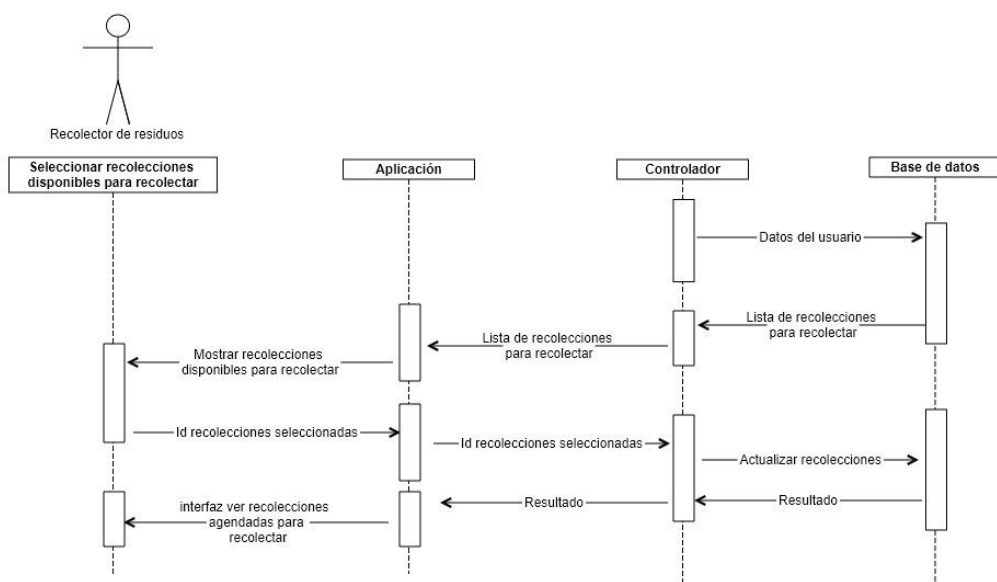
## Ver recolecciones seleccionadas para recolectar



## Crear solicitud de recolección



## Seleccionar recolecciones disponibles para recolectar



## Base de datos

```
CREATE DATABASE desechosorganicosapp;
use desechosorganicosapp;

CREATE TABLE IF NOT EXISTS usuarios (
  usuario_id int(6) unsigned zerofill NOT NULL AUTO_INCREMENT,
  usrName varchar(50) NOT NULL,
  pwd varchar(50) NOT NULL,
  telefono varchar(30) NOT NULL,
  rol tinyint, /*1:Usuario 2:Recolectos*/
  puntuacionUsr int,
  puntuacionRec int,
  PRIMARY KEY (usuario_id),
  UNIQUE KEY unique_id (usrName)
);

CREATE TABLE IF NOT EXISTS recolecciones (
  recoleccion_id int(6) unsigned zerofill NOT NULL AUTO_INCREMENT,
  direccion varchar(500) NOT NULL,
  fechade datetime NOT NULL,
  fechahasta datetime NOT NULL,
  cantidad varchar(50) ,
  descripcion varchar(500),
  repetir varchar(10) NOT NULL,
  usuario_id int(6) unsigned zerofill NOT NULL,
  recolector_id int(6) unsigned zerofill ,

  PRIMARY KEY (recoleccion_id),
  FOREIGN KEY (usuario_id) REFERENCES usuarios(usuario_id),
  FOREIGN KEY (recolector_id) REFERENCES usuarios(usuario_id)
);
```

## Código parcial web service

```
const express = require('express');
const router = express.Router();
const mysqlConnection = require('../database.js');

// GET all recolecciones
router.get('/api/recolecciones', (req, res) => {
  mysqlConnection.query('select * from recolecciones order by fechade;', (err, rows, fields) => {
    if(!err) {
      res.json(rows);
    } else {
      console.log(err);
    }
  });
});

// GET recoleccion
router.get('/api/recolecciones/:id', (req, res) => {
  const { id } = req.params;
  mysqlConnection.query('SELECT * FROM recolecciones WHERE recoleccion_id = ? order by fechade', [id], (err, rows, fields)
=> {
    if (!err) {
      res.json(rows[0]);
    } else {
      console.log(err);
    }
  });
});

// DELETE An recolecciones
router.delete('/api/recolecciones/:id', (req, res) => {
  const { id } = req.params;
  mysqlConnection.query('DELETE FROM recolecciones WHERE recoleccion_id = ?', [id], (err, rows, fields) => {
    if(!err) {
      console.log("Usuario eliminado");
      res.json({status: 200});
    } else {
      console.log("Error usr eliminado");
      console.log(err);
    }
  });
});
```

```

router.post('/api/recoleccion', (req, res) => {
  const {recoleccion_id, direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuarioid, recolectorid} = req.body;
  console.log(recoleccion_id, direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuarioid, recolectorid);
  console.log("usrid="+usuarioid);
  const query = `
    insert into recolecciones (direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuario_id, recolector_id) values
    (?, ?, ?, ?, ?, ?, ?, ?);
  `;
  mysqlConnection.query(query, [direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuarioid, recolectorid], (err,
rows, fields) => {
    if(!err) {
      res.json({status: 200});
    } else {
      console.log(err);
    }
  });
});
//Actualizar
router.put('/api/recoleccion/:id', (req, res) => {
  const {recoleccionid, direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuarioid, recolectorid} = req.body;
  console.log(recoleccionid, direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuarioid, recolectorid);
  const query = `
    update recolecciones set direccion=?, fechade=?, fechahasta=?, cantidad=?, descripcion=?, repetir=?, usuario_id=?, recol
ector_id=? where recoleccion_id=?;
  `;
  mysqlConnection.query(query, [direccion, fechade, fechahasta, cantidad, descripcion, repetir, usuarioid, recolectorid, recolecc
ionid], (err, rows, fields) => {
    if(!err) {
      res.json({status: 200});
    } else {
      console.log(err);
    }
  });
});
/* /api/recoleccion/usuario/ */
router.get('/api/recoleccion/usuario/:usrId', (req, res) => {
  const { usrId } = req.params;
  mysqlConnection.query('select * from recolecciones where usuario_id=? order by fechade;', [usrId], (err, rows, fields) =
> {
    if (!err) {
      console.log("Consulta de recolecciones del usuario");
      res.json(rows);
    } else {
      console.log(err);
    }
  });
});

```

## Código parcial aplicación consumir web service

```
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:residuos/src/models/recolecciones.dart';
import 'package:residuos/src/models/shared_preferences.dart';
import 'package:residuos/src/models/usuarios.dart';
import 'package:residuos/src/variables/variables.dart';

class FuturosUsr {
  static Future<http.Response> createUser(Usuario usr) {
    return http.post(
      Uri.http(ApiEndPointData.endPoint, "/api/usuarios/"),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
      body: jsonEncode(<String, String>{
        "usrName": usr.usrName,
        "pwd": usr.pwd,
        "telefono": usr.telefono,
        "rol": usr.rol.toString(),
        "puntuacionUsr": usr.puntuacionUsr.toString(),
        "puntuacionRec": usr.puntuacionRec.toString(),
      })),
    );
  }

  static Future<http.Response> updateUser(Usuario usr) {
    return http.put(
      // '${ApiEndPointData.usuarios}/${usr.usuarioId}'
      Uri.http(ApiEndPointData.endPoint, '/api/usuarios/${usr.usuarioId}'),
      headers: <String, String>{
        'Content-Type': 'application/json; charset=UTF-8',
      },
      body: jsonEncode(<String, String>{
        "usrName": usr.usrName,
        "pwd": usr.pwd,
        "telefono": usr.telefono,
        "rol": usr.rol.toString(),
        "puntuacionUsr": usr.puntuacionUsr.toString(),
        "puntuacionRec": usr.puntuacionRec.toString(),
      })),
    );
  }
}
```

```

static Future<Usuario> getUsuario(String usr, String pwd) async {
    var url =
        Uri.http(ApiEndPointData.endPoint, '/api/usuarios/validar/$usr/$pwd');
    final response = await http.get(url);
    if (response.statusCode == 200) {
        String body = utf8.decode(response.bodyBytes);
        if (body == "" || body.isEmpty) {
            print("Usr vacio");
            return null;
        }
        final jsonData = jsonDecode(body);
        int usuarioId = int.parse(jsonData["usuario_id"].toString());
        String usrName = jsonData["usrName"];
        String pwd = jsonData["pwd"];
        String telefono = jsonData["telefono"];
        int rol = int.parse(jsonData["rol"].toString());
        int puntuacionUsr = int.parse(jsonData["puntuacionUsr"].toString());
        int puntuacionRec = int.parse(jsonData["puntuacionRec"].toString());
        Usuario usr = Usuario(
            usuarioId, usrName, pwd, telefono, rol, puntuacionUsr, puntuacionRec);
        return usr;
    } else {
        throw Exception("Fallo la conexion");
    }
}

class FuturosRecolec {
    static Future<http.Response> createRecol(Recolecciones rec) {
        return http.post(
            Uri.http(ApiEndPointData.endPoint, "/api/recolecciones/"),
            headers: <String, String>{
                'Content-Type': 'application/json; charset=UTF-8',
            },
            body: jsonEncode(<String, String>{
                "direccion": rec.direccion,
                "fechade": rec.fechade,
                "fechahasta": rec.fechahasta,
                "cantidad": rec.cantidad,
                "descripcion": rec.descripcion,
                "repetir": rec.repetir,
            }
        );
    }
}

```

```

        "usuarioid": rec.usuarioid.toString(),
        "recolectorid": rec.recolectorid.toString(),
    }}, );}

static Future<http.Response> updateRecol(Recolecciones rec) {
    return http.put(
        Uri.http(
            ApiEndPointData.endPoint, '/api/recolecciones/${rec.recoleccionid}'),
        headers: <String, String>{
            'Content-Type': 'application/json; charset=UTF-8',
        },
        body: jsonEncode(<String, String>{
            "recoleccionid": rec.recoleccionid.toString(),
            "direccion": rec.direccion,
            "fechade": rec.fechade,
            "fechahasta": rec.fechahasta,
            "cantidad": rec.cantidad,
            "descripcion": rec.descripcion,
            "repetir": rec.repetir,
            "usuarioid": rec.usuarioid.toString(),
            "recolectorid": rec.recolectorid.toString(), })), ); }

static Future<http.Response> deleteRecoleccionById(int id) async {
    var url = Uri.http(ApiEndPointData.endPoint, '/api/recolecciones/$id');
    return await http.delete(url); }

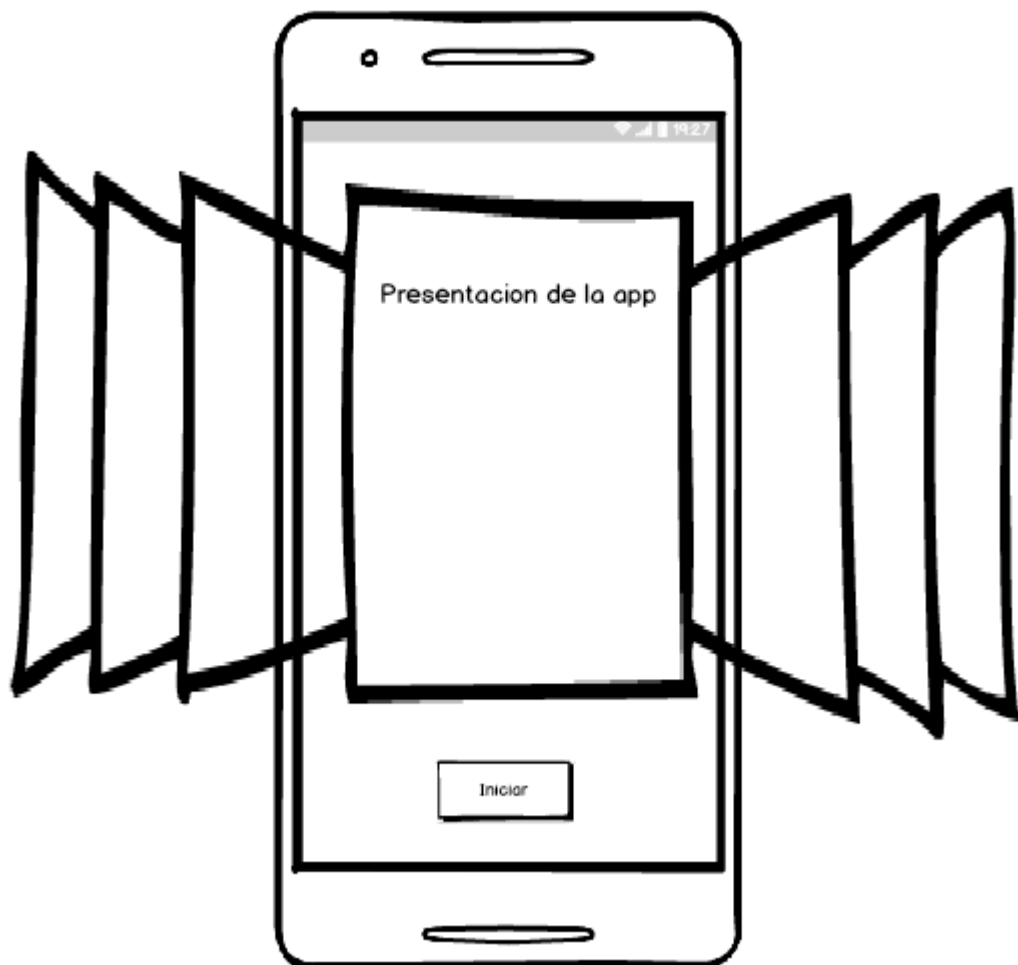
static Future<List<Recolecciones>> getRecolecciones() async {
    print("Inicio futuro");
    var url = Uri.http(ApiEndPointData.endPoint, '/api/recolecciones');
    final response = await http.get(url);
    if (response.statusCode == 200) {
        print("Estado 200 get recolecciones");
        String body = utf8.decode(response.bodyBytes);
        if (body == "" || body.isEmpty) {
            print("No existen recolecciones");
            return null;
        }
        final jsonData = jsonDecode(body);
        List<Recolecciones> listRecol = [];
        for (var item in jsonData) {
            int recoleccionid = int.parse(item["recoleccion_id"]);
            String direccion = item["direccion"];
            String fechade = item["fechade"];
            String fechahasta = item["fechahasta"];
            String cantidad = item["cantidad"];
            String descripcion = item["descripcion"];

```

**Prototipo de la solución.**



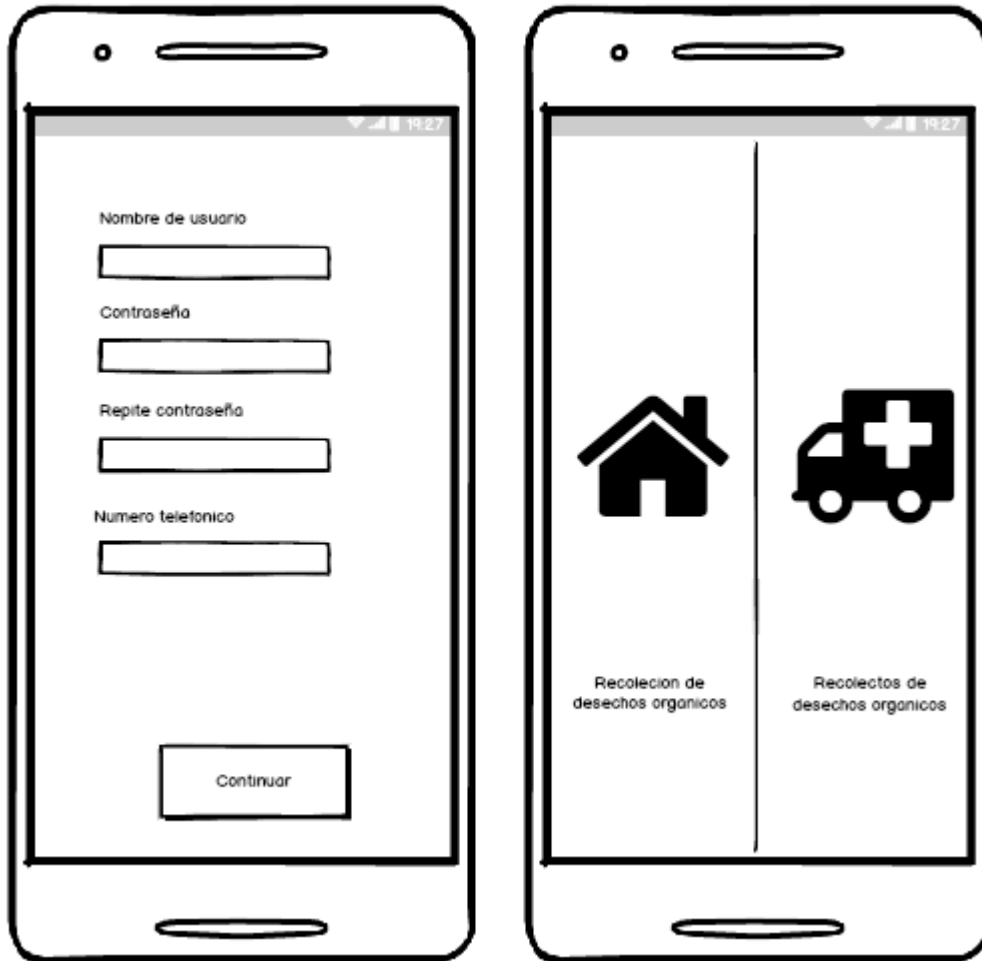
Pantalla de presentación



Pantalla de Inicio de sección



Pantalla de registro de usuarios



## Pantallas del usuario donador de residuos orgánicos

Recolecciones programadas

Direcc	fecha	Hora	Selecc	repetir c	modif
direcc	00-00-	12:00am-	<input type="checkbox"/>	nunca	
direcc	00-00-	12:00am-	<input type="checkbox"/>	semanal	
direcc	00-00-	12:00am-	<input type="checkbox"/>	diario	

Agendar Recoleccion

Direccion de recoleccion

Fecha

☐ Recolectar diariamente

☐ Recolectar semanalmente

APRIL 2021

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

## Pantallas del usuario recolector de residuos orgánicos

Recolecciones programadas

Dirección	fecha	Hora	Selección	repetir o	modif
direccion	00-00-	12:00am-	<input type="checkbox"/>	nunca	
direccion	00-00-	12:00am-	<input type="checkbox"/>	semanal	
direccion	00-00-	12:00am-	<input type="checkbox"/>	diario	

Agendar Recoleccion

Seleccionar usuarios para recolectar residuos

Dirección	fecha	Hora	Seleccionar
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>
direccion	00-00-20	12:00am-3:00	<input type="checkbox"/>

Apartar

## Desarrollo

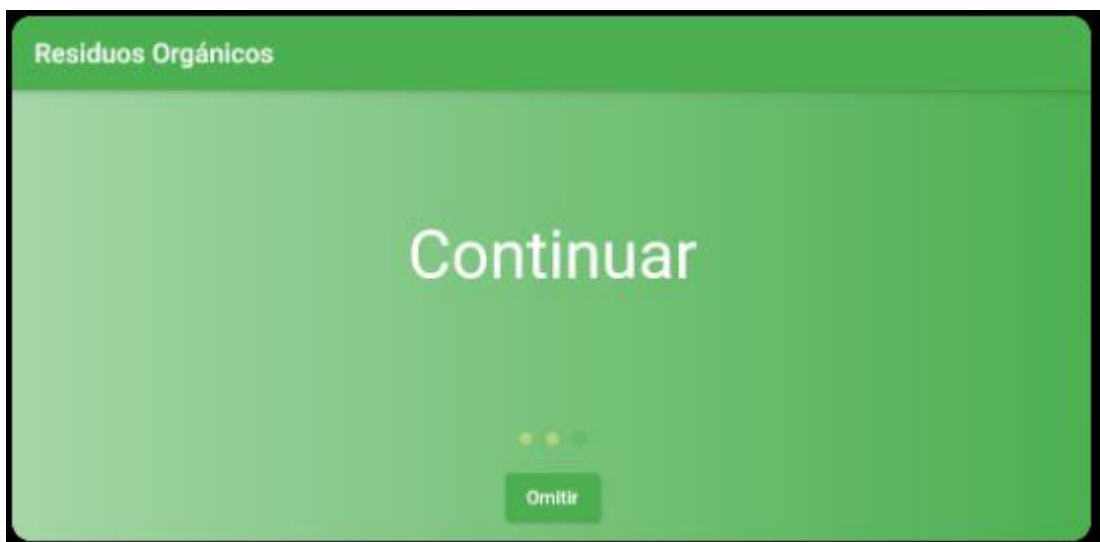
Pantalla presentación

(Vertical)





(Horizontal)



Pantalla inicio de sección  
(Horizontal)



Residuos Orgánicos



Usuario

Nombre Usuario

Contraseña

Password

(Vertical)

Residuos Orgánicos



Usuario

Nombre Usuario

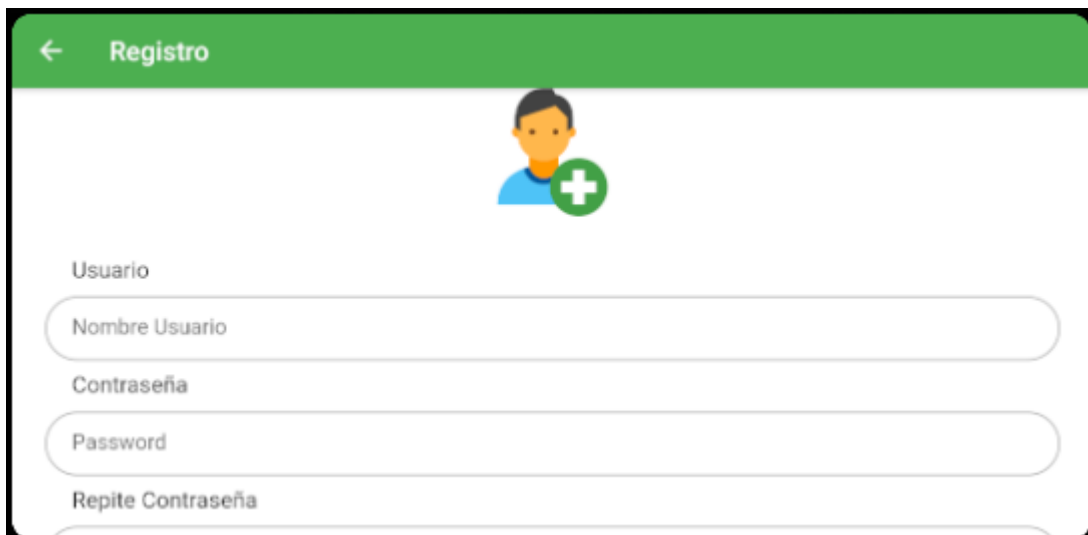
Contraseña

Password

Iniciar sesión

Registrarse

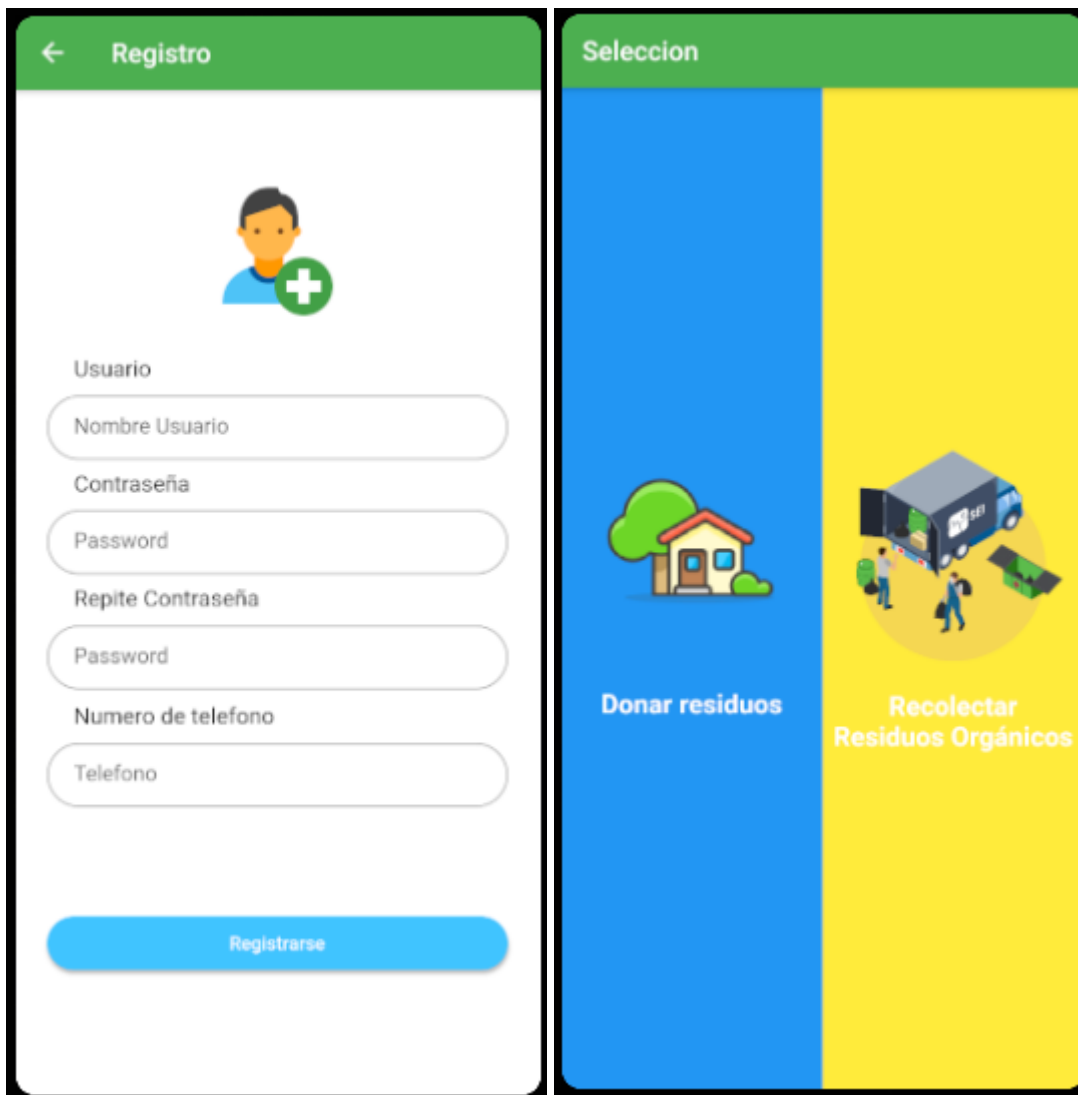
Pantalla registro  
(Horizontal)



Horizontal registration screen with a green header bar containing a back arrow and the title "Registro". Below the header is a white area with a user icon (a person with a green plus sign). The form contains four input fields: "Usuario" (with sub-label "Nombre Usuario"), "Contraseña" (with sub-label "Password"), and "Repite Contraseña".



(Vertical)



Pantallas del usuario donador de residuos orgánicos

(Horizontal)

Residuos Orgánicos					
Recolecciones programadas					
Fecha de	limite	cantidad	descripción	Repetición	
2021-04-22 18:00:00	2021-04-23 04:00:00	3 kilos		unaVez	Eliminar

(Vertical)

Residuos Orgánicos

Recolecciones programadas

Estado	Dirección
En asignacion	Av hidalgo numero 321 numer

1-7 of 1<>

Agendar Recolección

Pantallas del usuario recolector de residuos orgánicos  
(Horizontal)

Residuos Orgánicos				
Lista de usuarios que se recolectaran				
Dirección	Fecha de	limite	cantidad	d
dir 2 rol 1	2021-04-22 15:00:00	2021-04-23 09:00:00	1 kilo	2
Av hidalgo numero 321 numero interior 18	2021-04-22 23:00:00	2021-04-23 09:00:00	3 kilos	

(Vertical)

Residuos Orgánicos	
Lista de usuarios que se recolectaran	
Dirección	Fecha
dir 2 rol 1	2021-04-22 15:00:00
Av hidalgo numero 321 numero interior 18	2021-04-22 23:00:00
1-7 of 2 < >	
Agendar Recolección	

Pantallas del usuario donador de residuos orgánicos agendar

(Horizontal)

A screenshot of a mobile application interface for scheduling organic waste collection. The interface is titled "Residuos Orgánicos" in a green header bar. Below the header, the main title "Agendar recolección de Residuos Orgánicos" is centered. The form consists of three input fields: "Direccion" with the value "Av hidalgo numero 321 numero interior 18", "Fecha de" with the value "2021-4-22 13:0", and "Fecha hasta" with the value "2021-4-22 23:0".

← Residuos Orgánicos

Agendar recolección de Residuos Orgánicos

Direccion  
Av hidalgo numero 321 numero interior 18

Fecha de  
2021-4-22 13:0

Fecha hasta  
2021-4-22 23:0

(Vertical)

A screenshot of a mobile application interface for scheduling organic waste collection, shown in a vertical orientation. The interface is titled "Residuos Orgánicos" in a green header bar. Below the header, the main title "Agendar recolección de Residuos Orgánicos" is centered. The form consists of five input fields: "Direccion" with the value "Av hidalgo numero 321 numero interior 18", "Fecha de" with the value "2021-4-22 13:0", "Fecha hasta" with the value "2021-4-22 23:0", "Cantidad aproximada" with the value "3 kilos", and "Descripcion" which is empty. At the bottom, there are three radio button options: "Unica vez" (selected), "Recolectar diariamente", and "Recolectar semanalmente".

← Residuos Orgánicos

Agendar recolección de Residuos Orgánicos

Direccion  
Av hidalgo numero 321 numero interior 18

Fecha de  
2021-4-22 13:0

Fecha hasta  
2021-4-22 23:0

Cantidad aproximada  
3 kilos

Descripcion

☒ Unica vez  
☐ Recolectar diariamente  
☐ Recolectar semanalmente

## Pantallas del usuario recolector de residuos orgánicos agendar

(Horizontal)



The horizontal view of the 'Residuos Orgánicos' app screen displays a table of collection requests. The table has five columns: 'Dirección', 'Fecha de', 'limite', 'cantidad', and 'de'. The first two rows contain data, while the remaining three rows are empty. Each row has a checkbox on the left for selection.

<input type="checkbox"/>	Dirección	Fecha de	limite	cantidad	de
<input type="checkbox"/>	Av hidalgo numero 321 numero interior 18	2021-04-22 18:00:00	04:00:00 hrs	3 kilos	
<input type="checkbox"/>	dir rol1	2021-04-25 09:00:00	19:00:00 hrs	q	
<input type="checkbox"/>					
<input type="checkbox"/>					

(Vertical)



The vertical view of the 'Residuos Orgánicos' app screen displays a list of collection requests. The list has a checkbox on the left for selection. The first three rows contain data, while the remaining four rows are empty. At the bottom of the list, there is a green button labeled 'Agendar Recolección'.

<input type="checkbox"/>	Dirección
<input type="checkbox"/>	Av hidalgo numero 321 numero interior 18
<input type="checkbox"/>	dir rol1
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

1-7 of 2 < >

Agendar Recolección