

# Especificación de Requerimientos

## Conexión agro con empresas

### Plaza Campesina

### La Plaza

**Tabla 1. Nombres, correo electrónico y rol del equipo de trabajo SCRUM.**

Apellidos, Nombres	Correo electrónico	Rol
Cagua Carrillo Sayda Yamile	saydayamilecaguac@gmail.com	Gestor de Proyecto
José Daniel Ramírez Saldaña	Jose206@utp.edu.co	Gestor Base de Datos
Julián Andrés Segura González	julian888s@gmail.com	Desarrollador Fronted
Dora Paola Pacheco	paolapacheco.moreno@gmail.com	Desarrollador Backend
Isis Nirvana Segura Valero	isiissegura@gmail.com	Tester

**Tabla 2. Especificaciones del proyecto SCRUM.**

<b>CATEGORÍA</b>	Agro
<b>NOMBRE</b>	Conexión agro con empresas
<b>DESCRIPCIÓN</b>	Cuando los productos agrícolas salen de la tierra son susceptibles de sumar valor por medios procesos que les den algún otro tipo de características. Por ejemplo, los frutos secos cuando son acaramelados. Pero no siempre los mismos productores tienen la capacidad de hacerlo por lo que se hace necesario generar alianzas con personas y empresas que estén en condición de hacerlo eliminando así el papel del intermediario lo que podría aumentar la ganancia para los productores.
<b>OBJETIVO ESTRATÉGICO</b>	Conectar a productores agrícolas y campesinos con empresas o emprendedores para generar alianzas donde estos últimos agreguen valor sin necesidad de intermediarios.
<b>PÚBLICO OBJETIVO</b>	Productores y/o campesinos - Empresas y/o emprendedores
<b>IMPACTO ESPERADO</b>	Mayores ingresos de los campesinos al suprimir a los intermediarios y mejor precio al consumidor

- A continuación, se presentan las cartas de información que complementan las historias de usuarios del proyecto: Conexión agro con empresas:

**Tabla 3. Especificaciones de historias de usuario.**

Historias de Usuario	
<b>Número: 01</b>	<b>Nombre: REGISTRAR USUARIO</b>
<b>Puntos estimados:</b>	
<b>Descripción:</b> El usuario (Emprendedor y Productor) requiere registrarse en la aplicación, con la finalidad de poder ingresar mis datos al sistema.	
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>Cuando se ingresa un usuario y/o contraseña no válida, el sistema no debe permitir el ingreso y se presentará un error con mensaje: “usuario y/o contraseña incorrecta, por favor ingresar caracteres válidos (Solo se aceptan números y letras)”.</li> <li>Los datos para este registro son:  Nombres y apellidos y/o razón social, Identificación, Ciudad/Municipio, Dirección, Email, Tipo de usuario (Emprendedor o Productor)</li> </ul>	

Historias de Usuario	
<b>Número: 02</b>	<b>Nombre: INICIAR SESIÓN</b>
<b>Puntos estimados:</b>	

**Descripción:** El usuario (Emprendedor y Productor), requiere ingresar al sistema, para poder hacer uso del sistema.

**Criterios de Aceptación:**

- Cuando el ingreso del usuario y contraseña son incorrectos, entonces el sistema NO permitirá el ingreso y el sistema presentará una alerta con el siguiente mensaje: “Usuario y/o contraseña no válida (Sólo se aceptan números y letras)”.

**Historias de Usuario**

**Número:** 03

**Nombre:** CAMBIAR CONTRASEÑA

**Puntos estimados:**

**Descripción:** Yo como Usuario, deseo cambiar la contraseña, para poder ingresar al sistema.

**Criterios de Aceptación:**

- Si el usuario digita un carácter incorrecto, debe aparecer un mensaje de error “carácter no valido, por favor ingrese una contraseña válida”
- Solo se aceptan números y letras.

Historias de Usuario	
<b>Número: 04</b>	<b>Nombre: BUSCAR PRODUCTO</b>
<b>Puntos estimados:</b>	
<b>Descripción:</b> El Emprendedor ingresa al sistema con la necesidad de encontrar un producto de interés, para esto se requiere un listado de productos.	
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>Si el usuario ingresa un producto que no se encuentra en la lista, debe aparecer un error con el siguiente mensaje "El producto ingresado no se encuentra disponible por los productores".</li> </ul>	

Historias de Usuario	
<b>Número: 05</b>	<b>Nombre: SELECCIONAR PRODUCTO</b>
<b>Puntos estimados:</b>	
<b>Descripción:</b> El Emprendedor identifica el producto de interés, para poder encontrar los datos de contacto necesita seleccionar el producto.	
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>El usuario selecciona el producto, pero no hay respuesta del servidor para efectuar la consulta a la base de datos. El sistema debe presentar un error</li> </ul>	



donde se presente el siguiente mensaje “Problemas con el servidor, por favor realizar la consulta otra vez”

Historias de usuario	
Número: 6	Nombre: CREAR PRODUCTO(Crear)
Puntos Estimados:	
Descripción: El Productor, necesita crear un producto, con la finalidad de poder comercializar su producto en el sistema.	
<p>Criterios de aceptación:</p> <p>El producto debe quedar registrado con un nombre real y único, ya que podrá ser más fácil su ubicación por otros usuarios.</p> <p>Para la creación de un producto se debe tener en cuenta el, Nombre del producto, Descripción del producto (Tipo de producto, Calidad/Estado, Tamaño, Cantidad, Precio, Disponibilidad), Adjuntar foto del producto a comercializar, de lo contrario no quedará guardado en el sistema.</p>	

Historias de usuario	
Número: 7	Nombre: LISTAR PRODUCTOS

Puntos Estimados:

El Productor necesita ver la lista de productos creados en el sistema.

Criterios de aceptación:

El listado de productos estará organizado por el nombre.

El Productor puede validar en la parte derecha de la pantalla la lista de productos añadidos así mismo validar la información agregada de cada producto, permite modificación y eliminación adicional arroja una alerta (¿Está seguro de eliminar este producto? opción SI NO) en caso de eliminar algún producto.

Historias de usuario

Número: 8

Nombre: MODIFICAR PRODUCTO (Editar)

Puntos Estimados:

El Productor, necesita modificar su producto creado, con la finalidad de poder realizar algún cambio o actualización en el sistema.

Criterios de aceptación:

El productor debe actualizar el producto con los requerimientos solicitados por el sistema y de forma clara y concisa, de lo contrario el registro no será válido y/o almacenado en el sistema.

Para la actualización de un producto se debe tener en cuenta el Tipo de producto, Calidad/Estado, Tamaño, Precio, Disponibilidad, Adjuntar foto del producto a comercializar, de lo contrario no quedará guardado en el sistema.

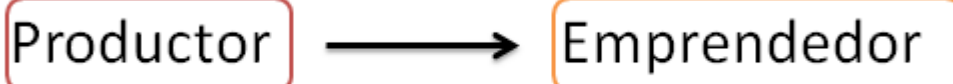


Historias de usuario	
Número: 9	Nombre: BORRAR PRODUCTO (Eliminar)
Puntos Estimados:	
El Productor, necesita eliminar un producto, con la finalidad de poderlo borrar del sistema.	
<p>Criterios de aceptación: El productor debe tener en claro cuál es el producto que desea eliminar, debe eliminar el producto que no desea seguir ofreciendo y no cuenta con disponibilidad.</p> <p>Antes de eliminar saldrá una advertencia: “¿Está seguro de eliminar este producto?” opción SI NO.</p>	

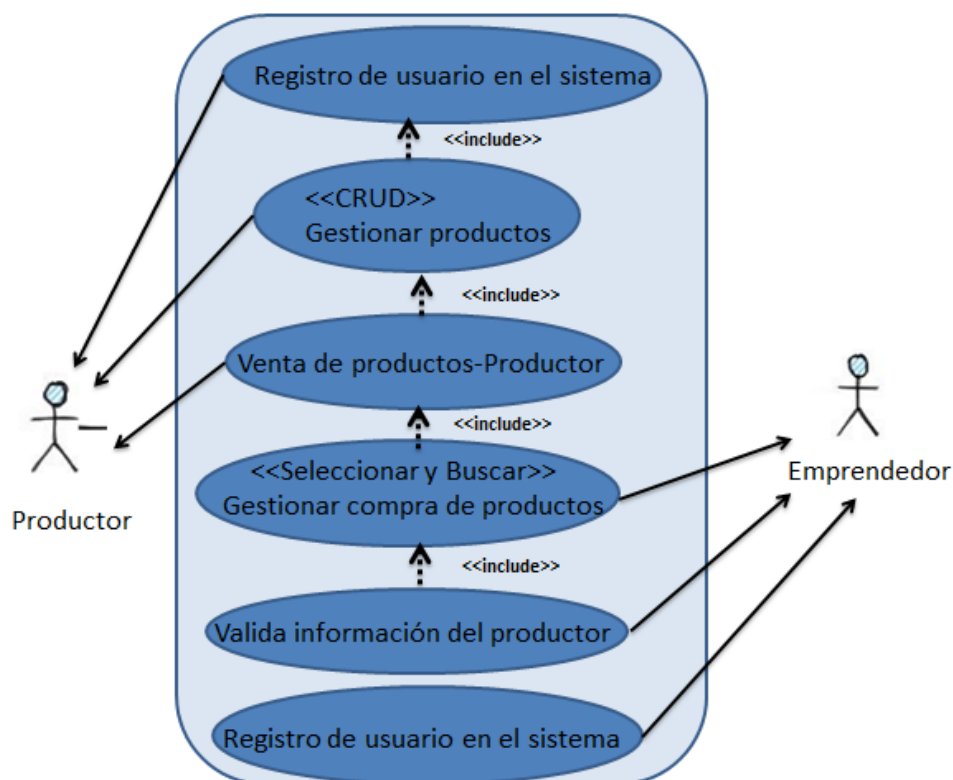
Este software se dividirá en los siguientes módulos:

- Módulo de Registro e ingreso
- Módulo de productos
- Módulo de usuario/emprendedor
- Módulo de productor

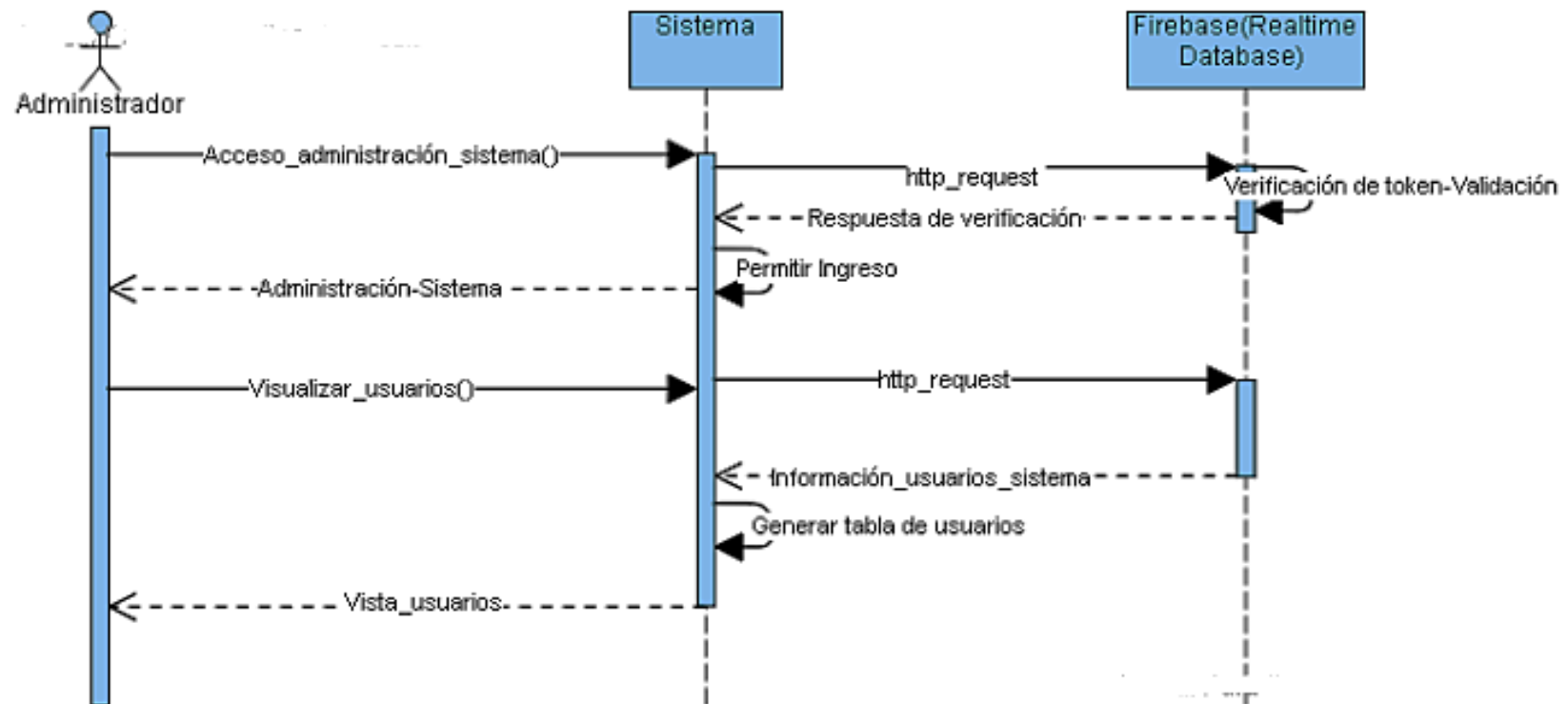
Proceso de comercialización:



Definición caso de uso:



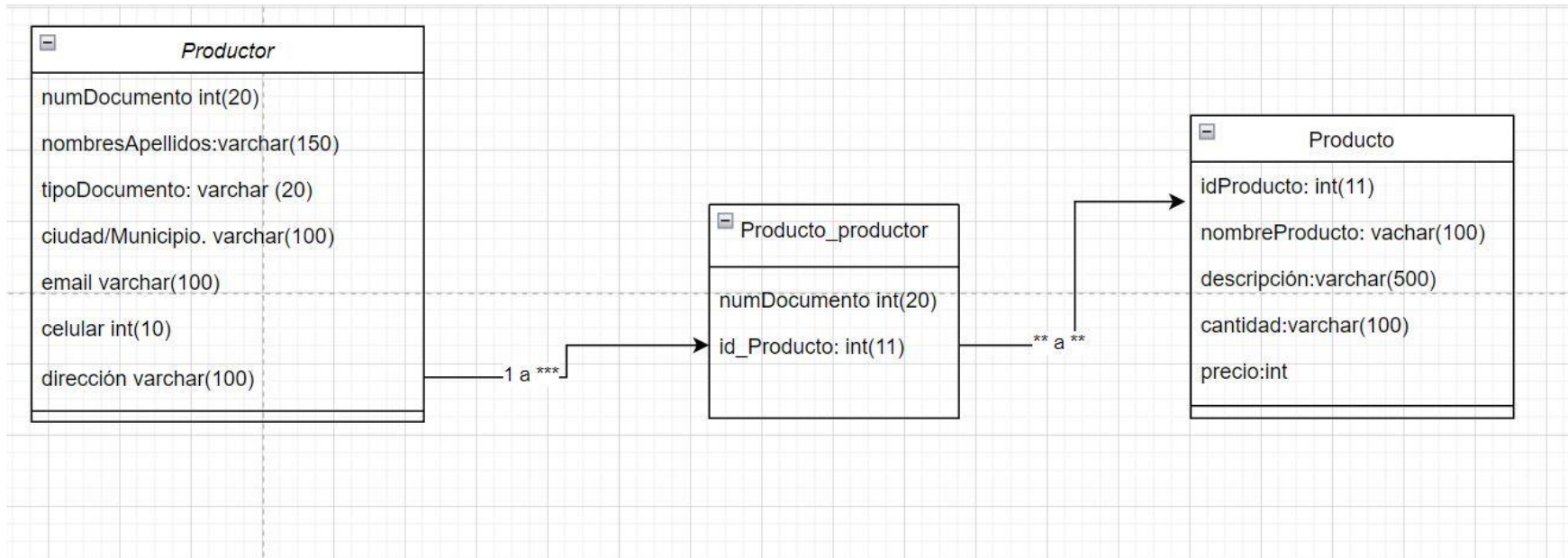
### Diagrama de secuencia-ejemplo-acceso y petición administrador:



### Caso de uso-Sistema- #1:

Caso de uso	Venta de productos de productor a Emprendedor
Actores	Productor, Emprendedor
Tipo	Include
Propósito	Realizar la venta dentro del proceso de comercialización
Resumen	Se realiza de productor a Emprendedor
Precondiciones	Producto en época de cosecha
Flujo Principal	Se validan las condiciones del producto para determinar el valor
Excepciones	Los emprendedores en algunos casos realizan la compra directamente con el productor en puntos de venta físicos, (pequeños locales o puestos en vías.)

## Diagrama de entidad relación:



## Mockups

# PROYECTO AGRO

PRO

Contraseña y/ o usuario incorrectos,  
intente otra vez

Aceptar

G

INICIAR SESIÓN

USUARIO

CONTRASEÑA

INICIAR

OLVIDO SU CONTRASEÑA?

ERES NUEVO/A? REGÍSTRATE ACA

INICIAR SESIÓN

USUARIO

CONTRASEÑA

INICIAR

OLVIDO SU CONTRASEÑA?

ERES NUEVO/A? REGÍSTRATE ACA

# PROYECTO AGRO

INICIAR SESIÓN

NOMBRES Y APELLIDOS/ RAZÓN SOCIAL

IDENTIFICACION/NIT

CIUDAD MUNICIPIO

DIRECCION

EMAIL

☒ EMPRENDEDOR

☐ PRODUCTOR

INICIAR



# PROYECTO AGRO

EMPRENDEDOR

PRODUCTO

DESCRIPCION DEL PRODUCTO/  
PRODUCTORES, PROVEEDORES

PRECIO

BUSCAR

SELECCIONAR

# PROYECTO AGRO

PRODUCTOR

PRODUCTO

DESCRIPCION DEL PRODUCTO

CANTIDAD

PRECIO

CREAR

MODIFICAR

BORRAR

LISTAR

ADJUNTAR FOTO

PLATANO

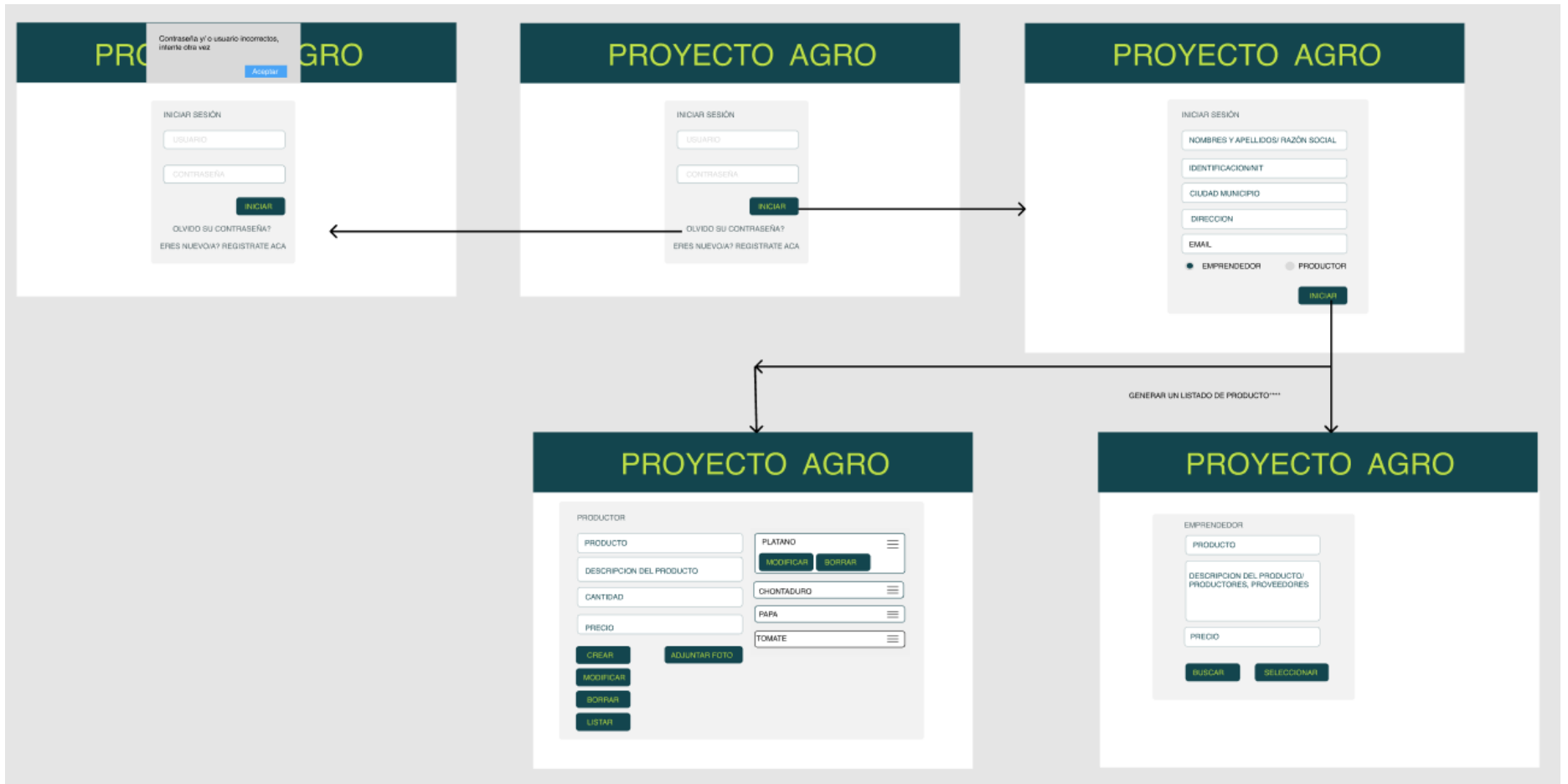
MODIFICAR

BORRAR

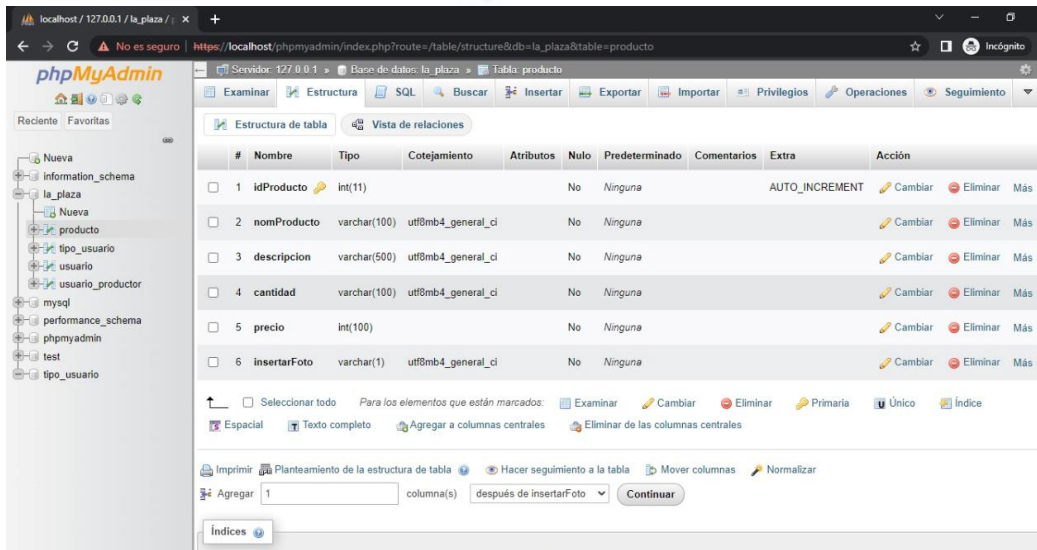
CHONTADURO

PAPA

TOMATE



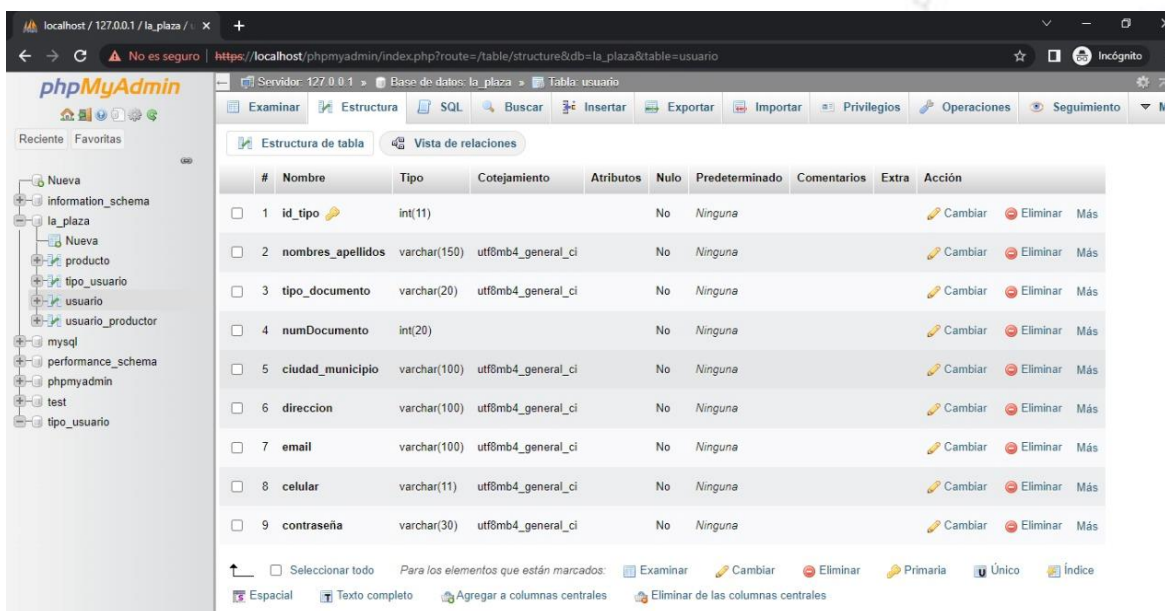
## Tablas de Bases de Datos



phpMyAdmin interface showing the structure of the 'producto' table in the 'la\_plaza' database. The table has 6 columns:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	idProducto	int(11)			No	Ninguna		AUTO_INCREMENT	Más
2	nomProducto	varchar(100)	utf8mb4_general_ci		No	Ninguna			Más
3	descripcion	varchar(500)	utf8mb4_general_ci		No	Ninguna			Más
4	cantidad	varchar(100)	utf8mb4_general_ci		No	Ninguna			Más
5	precio	int(100)			No	Ninguna			Más
6	insertarFoto	varchar(1)	utf8mb4_general_ci		No	Ninguna			Más

Ilustración 1. Tabla de Producto.



phpMyAdmin interface showing the structure of the 'usuario' table in the 'la\_plaza' database. The table has 9 columns:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id_tipo	int(11)			No	Ninguna			Más
2	nombres_apellidos	varchar(150)	utf8mb4_general_ci		No	Ninguna			Más
3	tipo_documento	varchar(20)	utf8mb4_general_ci		No	Ninguna			Más
4	numDocumento	int(20)			No	Ninguna			Más
5	ciudad_municipio	varchar(100)	utf8mb4_general_ci		No	Ninguna			Más
6	direccion	varchar(100)	utf8mb4_general_ci		No	Ninguna			Más
7	email	varchar(100)	utf8mb4_general_ci		No	Ninguna			Más
8	celular	varchar(11)	utf8mb4_general_ci		No	Ninguna			Más
9	contraseña	varchar(30)	utf8mb4_general_ci		No	Ninguna			Más

Ilustración 2. Tabla Usuario

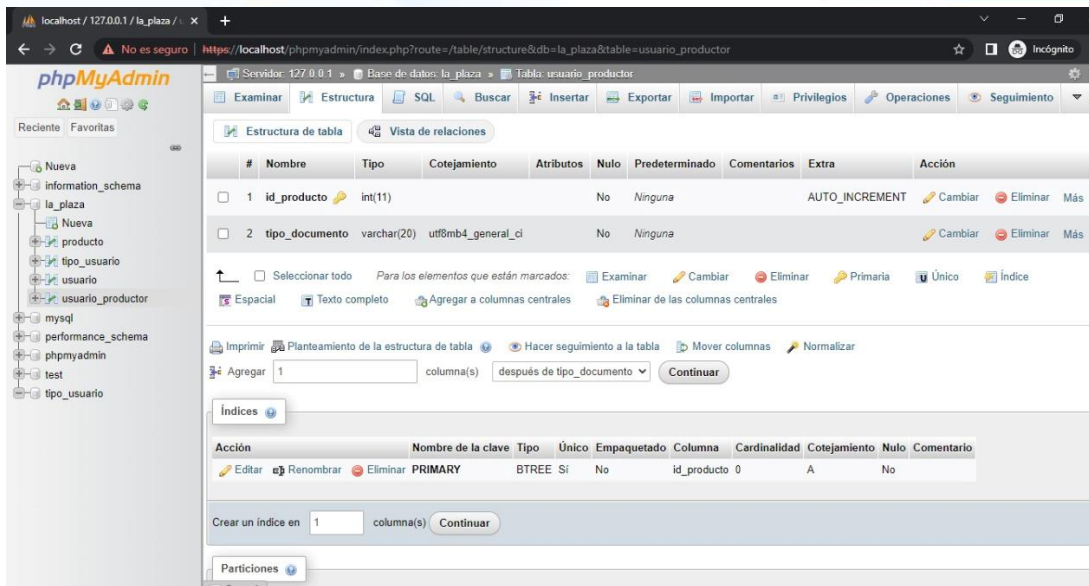
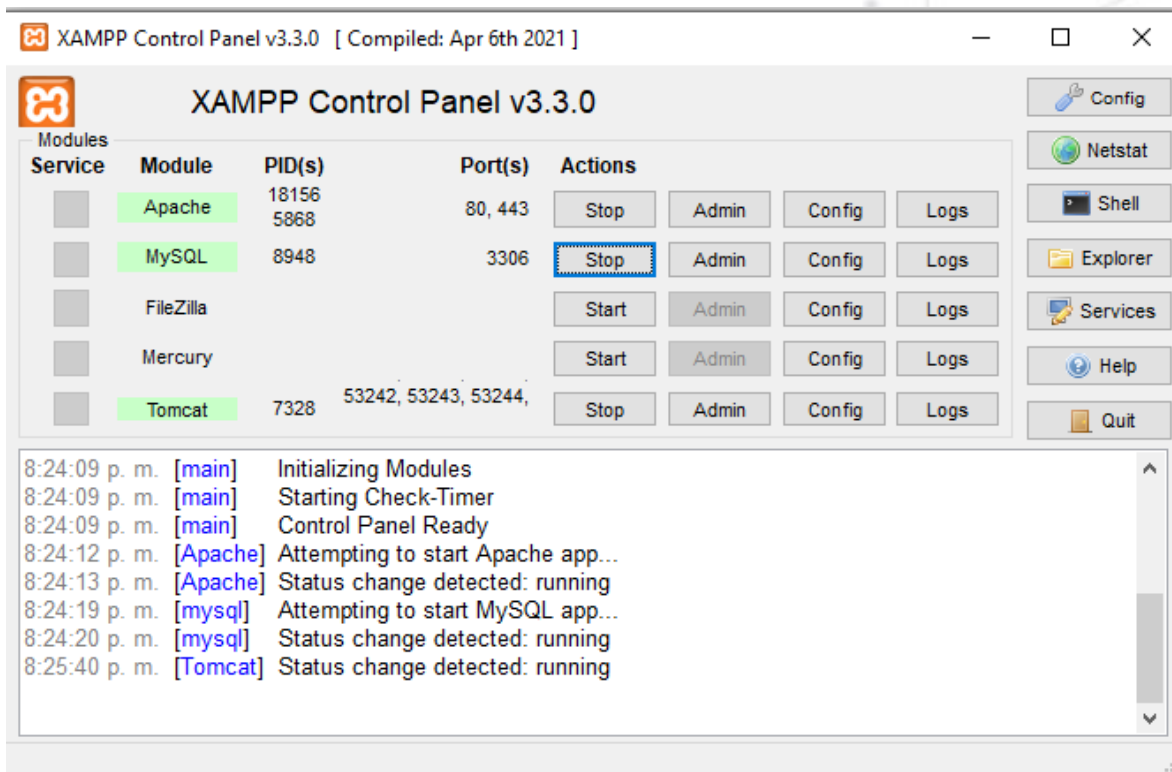


Ilustración 3. Tabla Usuario Productor

## Conexión con base de datos.

Es necesario establecer una conexión con la base de datos donde se almacena la información. Para el desarrollo del proyecto La Plaza, se utiliza un servidor local para emular una base de datos. Esto se realiza a través de la aplicación XAMPP.



Creación de Base de Datos.

Para crea la base de datos, se utiliza lenguaje SQL, a través del siguiente lenguaje:

```
CREATE DATABASE
```

Crea la tabla para producto en lenguaje SQL:

```
CREATE TABLE producto (  
    id_producto INT PRIMARY KEY,  
    cantidad VARCHAR(255) NOT NULL,  
    descripcion VARCHAR(255) NOT NULL,  
    nombre_producto VARCHAR(255) NOT NULL,  
    precio INT NOT NULL  
)
```

Crear Tabla Productor

```
CREATE TABLE productor (  
    cedula INT PRIMARY KEY,  
    id_productor INT,  
    nombre_productor VARCHAR(100) NOT NULL,  
    municipio VARCHAR(100) NOT NULL,  
    correo VARCHAR(100) NOT NULL,  
    celular VARCHAR(100) NOT NULL,  
)
```

Conexión con la base de datos

La configuración se realiza a través de fichero application.properties, el cual se crea al ejecutar el proyecto Spring Bot.



```

application.properties x ProductorService.java ProductoService.java Controlador.java
src > main > resources > application.properties
1 spring.datasource.url=jdbc:mysql://localhost:3306/bdpersona
2 spring.datasource.username=root
3 spring.datasource.password=
4 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
5 spring.jpa.hibernate.ddl-auto=update
  
```

Aquí se indica el nombre de la base de datos, el cual es bdpersona, el usuario y contraseña de la base de datos.

Consulta del método Obtener información

Para poder obtener la información de un registro de la base de datos, se necesita implementar un método que se desarrolla en el back end.

```

@Service
public class PersonaService implements IPersonaService {

    @Autowired
    private IPersona data;

    @Override
    public List<Persona> listar() {
        return (List<Persona>) data.findAll();
    }

    @Override
    public Optional<Persona> listarId(int id) {
        return Optional.empty();
    }

    @Override
    public int save(Persona p) {
        int res=0;
        Persona persona =data.save(p);

        if(!persona.equals(null)){
            res=1;
        }
        return res;
    }

    @Override
    public void delete(int id) {

    }
}
  
```



En este caso tenemos los métodos que implementa en la clase Persona, esto se hace a través del paquete de servicio.

Para poder mostrar el resultado de la implementación del método, se necesita crear archivo HTML.

En este archivo tenemos que agregar el thymeleaf, el cual realiza una query para implementar la consulta a la base de datos.

```
<!DOCTYPE html>
<html lang="en" xmlns:th="http://www.thymeleaf.com">
  <head>
```

Es importante llamar a la dependencia thymeleaf a la hora de crear la tabla en HTML, para poder incorporar elementos dinámicos en el fichero HTML sin tener que crear un archivo javascript.

```
<thead>
  <tr>
    <th>ID</th>
    <th>NOMBRES</th>
    <th>TELEFONO</th>
    <th>ACCIONES</th>
  </tr>
</thead>
<tbody>
```

Esta es la parte estática de la tabla de la pantalla /listar, pero para poder mostrar los datos desde la base de datos, se requiere utilizar la dependencia de thymeleaf.

```
<tbody>
  <!--lectura desde el Controlador -->
  <tr th:each="persona:${personas}">
    <td th:text="${persona.id}"></td>
    <td th:text="${persona.nombre}"></td>
    <td th:text="${persona.telefono}"></td>
    <td>
      <a class="btn btn-warning">Editar</a>
      <a class="btn btn-danger">Eliminar</a>
    </td>
  </tr>
</tbody>
```

Esta petición a recibe el controller, el cual va implementar el método a través de la interface del servicio, el cual a su vez implementa el servicio.

```

J ProductorService.java  J ProductoService.java  J PersonaService.java  J Contr
> java > com > crudEjemplo > demoCrud > controller > J Controlador.java > Language Support for Java(TM) by Red Hat

//Metodos para persona

@GetMapping("/listar")
public String listar(Model model){
    List<Persona> personas = service.listar();
    model.addAttribute(attributeName: "personas", personas);
    return "index";
}

@GetMapping("/new")
public String agregar(Model model){
    model.addAttribute(attributeName: "persona", new Persona());
    return "form";
}

@PostMapping("/save")
public String save(@Validated Persona p, Model model) {
    service.save(p);
    return "redirect:/listar";
}

```

Para llamar al método listar del objeto persona, se utiliza la petición a través de /listar, a través del programa POSTMAN probamos la petición de la API.

API / http://localhost:8080/listar

GET http://localhost:8080/listar **Send**

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (6) Test Results Status: 200 OK Time: 16 ms Size: 984 B Save Response

Pretty Raw Preview Visualize

**Nuevo**

ID	NOMBRES	TELEFONO	ACCIONES

A realizar la petición GET que es obtener, aparece STATUS:OK, cuando se revisa la dirección URL http://localhost:8080/listar, se presenta la tabla estática sin ningún registro.

localhost:8080/listar

Guía League of Kin... pdf - Google Drive Grupos generales -... (60) R2 Ciclo 3 - De... (60) R2 Ciclo 3 - De... (60) R2 Ciclo 3 - De...

Nuevo

ID NOMBRES TELEFONO ACCIONES

Para crear un registro se utiliza el método post, probar el método POST se utiliza el programa postman. Para acceder a este método es a través de /new.

API / http://localhost:8080/new

Save

Send

POST http://localhost:8080/new

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Al ingresar a esa dirección url, podemos ingresar los atributos correspondientes a la clase.

localhost:8080/new

Guía League of Kin... pdf - Google Drive Grupos generales -... (60) R2 Ciclo 3 - De... (60) F

## Modulo Persona

Content

ID

NOMBRES

TELEFONO

Guardar

Cuando se ingresa los campos correspondientes, se le da a guardar.

## Modulo Persona

Content

ID 1

NOMBRES Jose Daniel Ramirez

TELEFONO 3105914405

Guardar

Cuando guarda el registro, se devuelve a método listar para que el usuario pueda ver que se realizó el registro con éxito.

localhost:8080/listar

Guía League of Kin... pdf - Google Drive Grupos generales -... (60) R2 Ciclo 3 - De... (60) R2 Ciclo 3 - De... (60) R2 Ciclo 3 - De...

Nuevo

ID	NOMBRES	TELEFONO	ACCIONES
11	Jose Daniel Ramirez	3105914405	<a href="#">Editar</a> <a href="#">Eliminar</a>

Cuando se revisa la base de datos, se puede observar que el registro se guardó exitosamente.

- 'bdpersona'. 'persona'
- 'bdpersona'. 'producto'
- 'bdpersona'. 'productor'
- 'bdproducto'. 'producto'
- 'dbejemplo'. 'persona'
- 'laplaza\_db'. 'producto'
- 'laplaza\_db'. 'productor\_user'
- 'laplaza\_db'. 'usuario'
- 'laplaza\_db'. 'tipo\_usuario'
- + dbejemplo
- + information\_schema
- + laplaza\_db
- + mysql

SELECT \* FROM 'persona'

☐ Perfilando [ [Editar en línea](#) ] [ [Editar](#) ] [ [Explicar SQL](#) ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

Opciones extra

	id	nombre	telefono
<input type="checkbox"/> <a href="#">Editar</a> <a href="#">Copiar</a> <a href="#">Borrar</a>	11	Jose Daniel Ramirez	3105914405

↑ ☐ Seleccionar todo Para los elementos que están marcados: [Editar](#) [Copiar](#)

## INFORME DE RETROSPECTIVA:

En este sprint como tripulantes en proceso de aprendizaje, hemos logrado distribuir de manera equitativa según la metodología Scrum los requerimientos para cumplir con la evolución del proyecto, debido a esto hemos trabajado de una manera dinámica y organizada. Se logro corregir modelo de base de datos, se implementa el front y se realiza MVC. Adicionalmente se crea funcionalidad de la estructura de la plataforma trabajada gracias al seguimiento y supervisión continúa del tutor scrum. Las habilidades que hemos adquirido a lo largo del aprendizaje continuo bajo la supervisión conjunta con los recursos necesarios para este proyecto.

Una de las herramientas que nos ha servido de gran ayuda a sido investigación constante que nos ha permitido adjuntar y complementar lo solicitado en este sprint. Hemos tenido en cuenta el desarrollo de los anteriores sprint los cuales nos han permitido adquirir bases y desarrollarnos a lo largo de estas semanas.

En este sprint aprendimos a realizar el front del proyecto adicionalmente se logró enlazar con código y estructurar la funcionalidad de la página según las historias de usuario, ha ido evolucionando constantemente el proyecto y la visual, siendo así como con constante practica se ha construido una sólida base para el trabajo en equipo y ayuda entre los integrantes de este mismo.

Se ha tenido en cuenta la visual del cliente junto con los diferentes aportes de parte del equipo de trabajo con la metodología de Scrum y así también desarrollar habilidades de comunicación como equipo junto con los tutores.

las herramientas donde se está editando el desarrollo del proyecto son las siguientes aplicaciones:

**Trello:** <https://trello.com/b/HFOhNBkj/proyecto-software>

**GitHub:** <https://github.com/La-plaza>



## GESTIÓN DE CONFIGURACIÓN

**Gestor Proyecto** Sayda Yamile Cagua Carrillo

**Gestor Base Datos** José Daniel Ramírez Saldaña

**Desarrollador Frontend** Julián Andrés Segura González

**Desarrollador Backend** Dora Paola Pacheco Moreno

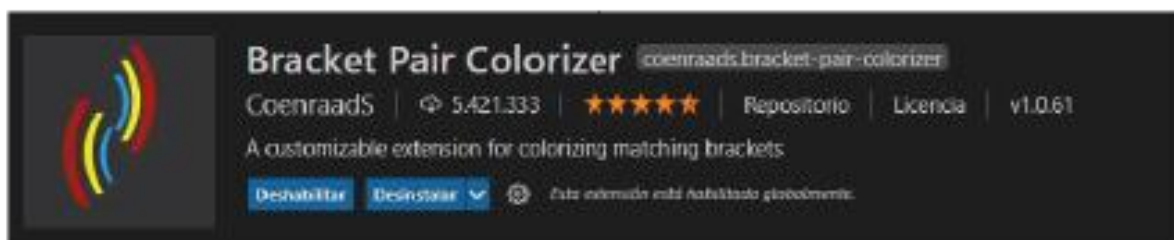
**Tester** Isis Nirvana Segura Valero

## Instalación del editor de código

Selección de editor de código: **Visual Studio Code**

Instalación de extensiones:

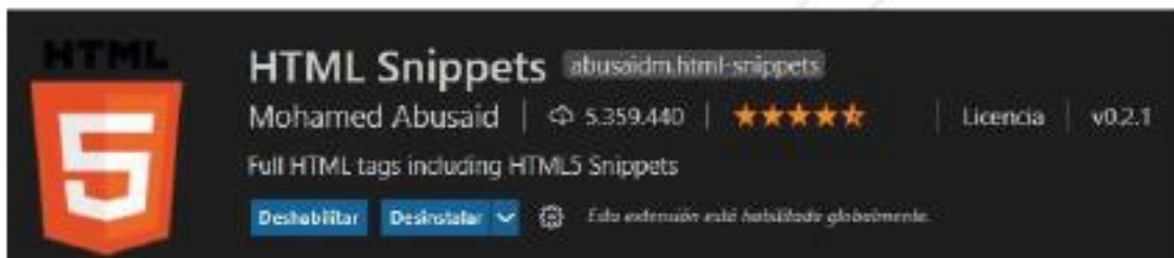
Bracket pair colorizer



ESLint: corregir errores de sintaxis en java script



HTML Snippets: completa código HTML



Intellisense for CSS class names in HTML: autocompletar o auto rellenar las clases CSS



**IntelliSense for CSS class names in HTML** zignod.html-css-class-completion  
Zignod | 3,459,339 | ★★★★★ | Repositorio | licencia | v1.20.0  
CSS class name completion for the HTML class attribute based on the definitions found in your workspace.  
Desactivar Desinstalar  Esta extensión está habilitada globalmente.