

INTEGRANTES DEL EQUIPO Y ROLES:

Nombres y apellidos	Rol	No. de identificación	E-mail	GRUPO
Jesús Miguel Herrera Salazar	Scrum máster	1001887467	herrerajisus@gmail.com	15 - 16
César Ivan Ávila Díaz	Diseñador de software	1067930961	ci.avila.diaz@gmail.com	29 - 30
Marisol Olivia Molina Uribe	Diseñador UI	1067855569	solmolina20@hotmail.com	29 - 30
Andrés Felipe Muñoz Acevedo	Tester	1036663334	anmuce03@gmail.com	29 - 30
CRISTIAN YOHVER SOLART E TORRES	ADMINISTRADOR CONFIGURACIÓN	1061785847	yo.vher@gmail.com	11-12

JACM PROGRAMACION

PROYECTO AGROSOFT

En este documento encontraremos toda la información necesaria para el desarrollo del proyecto AGROSOFT.

AGROSOFT es una microempresa colombiana, ubicada en el municipio de Boyacá – Santander, dedicada a la comercialización de insumos agrícolas y fertilizantes, ofreciendo a los agricultores soluciones orgánicas y agroecológicas en pro del desarrollo ambiental.

AGROSOFT ofrece un amplio catálogo en productos fitosanitarios, tales como herbicidas, fungicidas e insecticidas.

Trabajamos juntos para diseñar, crear y dar valor agregado a su empresa. Proporcionamos innovación tecnológica en los diversos proyectos o ideas que nos plantean desarrollar.

Nuestro equipo se especializa en acompañamiento a quienes requieran de nuestros servicios con un equipo de excelencia en el desarrollo tecnológico.

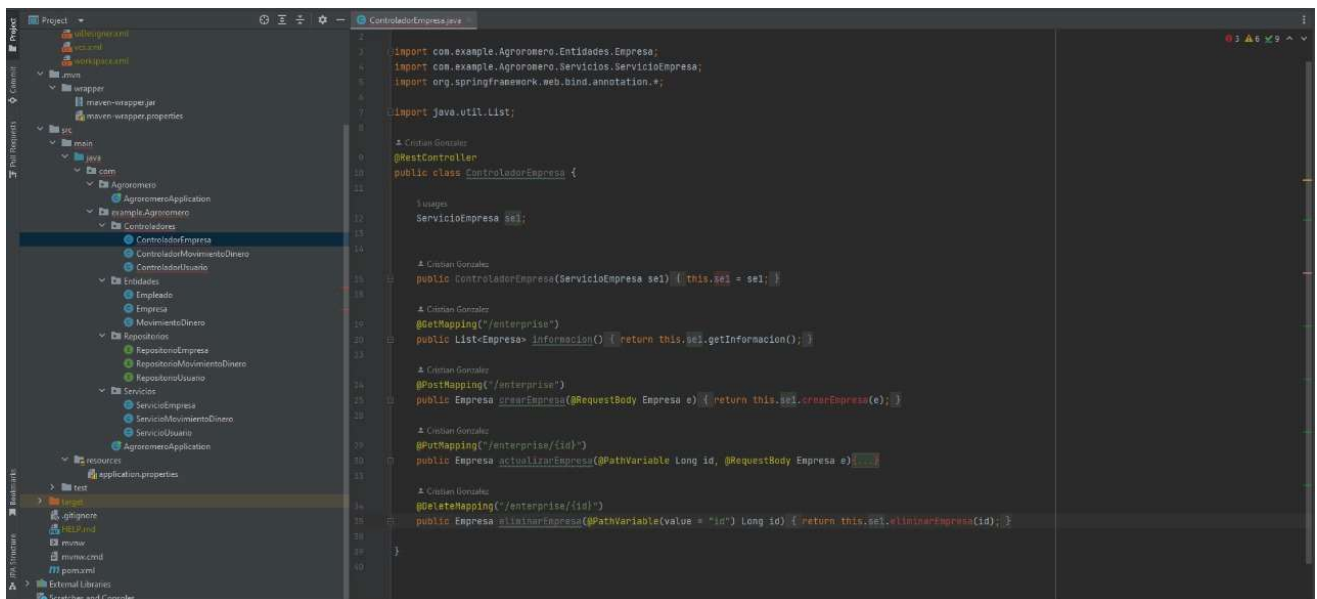
Realizar la creación de las clases para las entidades empresa, empleado y movimiento de dinero, junto con sus respectivos atributos y métodos para el desarrollo de la aplicación.

PROYECTO AGROROMERO

SPRINT 3: CREAR EL BACKEND PARA LA APLICACIÓN

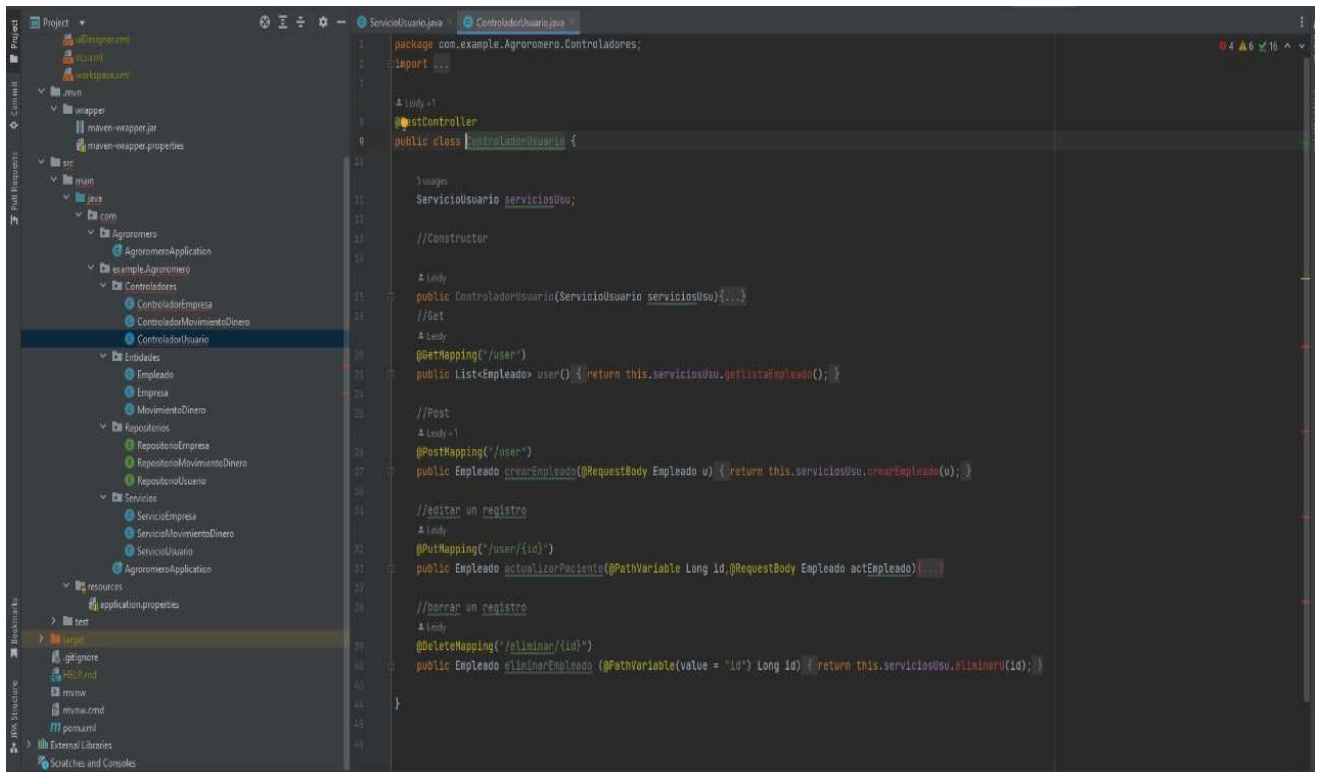
El backend es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos como, por ejemplo, la comunicación con el servidor.

1. Crear controladores REST para la empresa El sistema devuelve respuestas 200 en la ruta /enterprises con los siguientes verbos:
GET POST El sistema devuelve respuestas 200 en la ruta



/enterprises/{id} con los siguientes verbos: GET PATCH DELETE

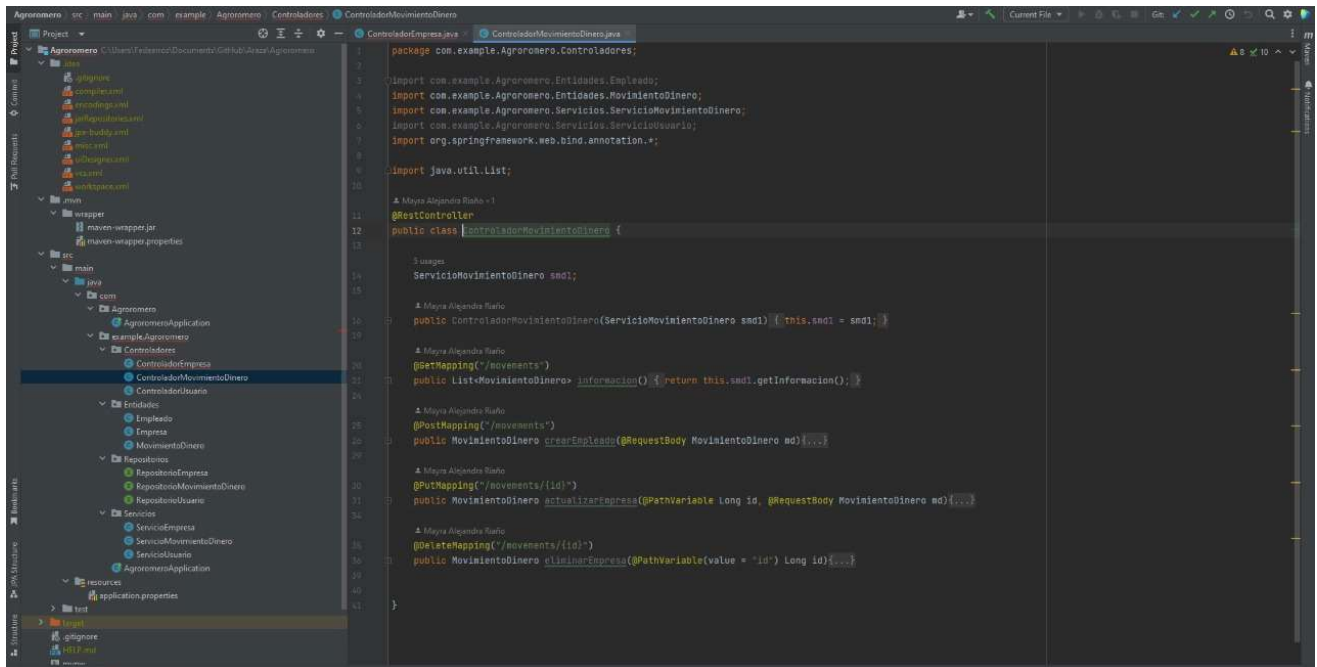
2. Crear controladores REST para los usuarios: El sistema devuelve respuestas 200 en la ruta /users con los siguientes verbos: GET POST El sistema devuelve respuestas 200 en la ruta /user/[id] con los siguientes verbos: GET PATCH DELETE



```
1 package com.example.Agropecuario.Controladores;
2 import ...
3
4 @RestController
5 public class ControladorUsuario {
6
7     // Inyección de dependencias
8     ServicioUsuario servicioUsu;
9
10    // Constructor
11
12    @Init
13    public ControladorUsuario(ServicioUsuario servicioUsu){...}
14
15    // GET
16    @GetMapping("/users")
17    public List<Empleado> user() { return this.servicioUsu.getListadoEmpleados(); }
18
19    // POST
20    @PostMapping("/user")
21    public Empleado crearEmpleado(@RequestBody Empleado u) { return this.servicioUsu.crearEmpleado(u); }
22
23    // Editar un registro
24    @PutMapping("/user/{id}")
25    public Empleado actualizarEmpleado(@PathVariable Long id, @RequestBody Empleado actEmpleado){...}
26
27    // Borrar un registro
28    @DeleteMapping("/eliminar/{id}")
29    public Empleado eliminarEmpleado(@PathVariable(value = "id") Long id) { return this.servicioUsu.eliminarU(id); }
30
31 }
```

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a 'Controladores' package with 'ControladorUsuario'. The code editor shows the implementation of 'ControladorUsuario' as a REST controller. It uses Spring's '@RestController' annotation and implements methods for GET, POST, PUT, and DELETE operations on the '/users' and '/user/{id}' endpoints. The controller depends on 'ServicioUsuario' and uses 'Empleado' as the data model.

3. Crear controladores REST para los movimientos de una empresa específica: El sistema devuelve responses 200 en la ruta /enterprises/[id]/movements con los siguientes verbos: GET POST PATCH



The screenshot shows an IDE with a project named 'Agronomero'. The left sidebar displays the project structure, with the package 'com.example.Agronomero.Controladores' selected. The main editor shows the code for 'ControladorMovimientoDinero.java'. The code defines a REST controller with the following methods:

```
package com.example.Agronomero.Controladores;

import com.example.Agronomero.Entidades.Empleado;
import com.example.Agronomero.Entidades.MovimientoDinero;
import com.example.Agronomero.Servicios.ServicioMovimientoDinero;
import com.example.Agronomero.Servicios.ServicioUsuario;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@Mayra Alejandra Rialto <1
@RestController
public class ControladorMovimientoDinero {

    @Mayra Alejandra Rialto
    ServicioMovimientoDinero smd1;

    @Mayra Alejandra Rialto
    public ControladorMovimientoDinero(ServicioMovimientoDinero smd1) { this.smd1 = smd1; }

    @Mayra Alejandra Rialto
    @GetMapping("/movements")
    public List<MovimientoDinero> informacion() { return this.smd1.getInformacion(); }

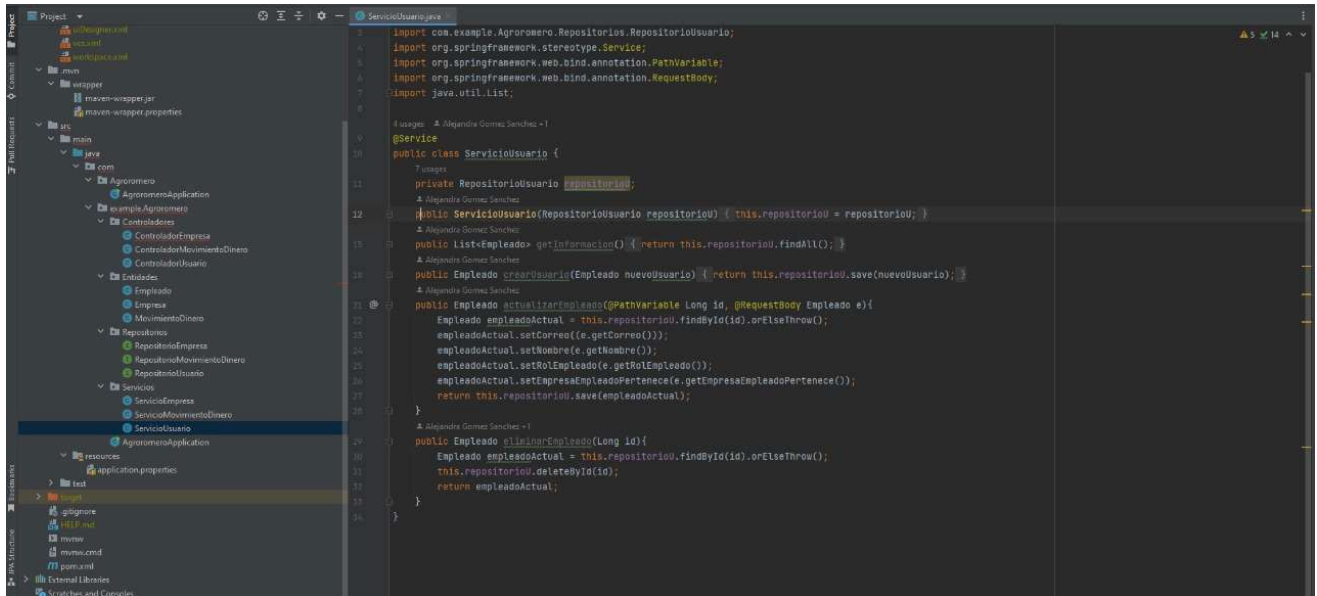
    @Mayra Alejandra Rialto
    @PostMapping("/movements")
    public MovimientoDinero crearEmpleado(@RequestBody MovimientoDinero md) { ... }

    @Mayra Alejandra Rialto
    @PutMapping("/movements/{id}")
    public MovimientoDinero actualizarEmpleado(@PathVariable Long id, @RequestBody MovimientoDinero md) { ... }

    @Mayra Alejandra Rialto
    @DeleteMapping("/movements/{id}")
    public MovimientoDinero eliminarEmpleado(@PathVariable(value = "id") Long id) { ... }
}
```

DELETE

4. Crear servicios para el usuario El sistema permite consultar todos los usuarios El sistema permite consultar un solo usuario El sistema permite crear un usuario El sistema permite editar un usuario El sistema permite eliminar un usuario



The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a package hierarchy for 'com.agrocom' with sub-packages like 'controladores', 'entidades', 'repositorios', and 'servicios'. The code editor displays the implementation of the 'ServicioUsuario' class, which includes methods for finding, saving, updating, and deleting users, as well as a method for finding a user by ID.

```
1 import com.example.Agrocom.Repositorios.RepositorioUsuario;
2 import org.springframework.stereotype.Service;
3 import org.springframework.web.bind.annotation.PathVariable;
4 import org.springframework.web.bind.annotation.RequestBody;
5 import java.util.List;
6
7
8
9 @Service
10 public class ServicioUsuario {
11     private RepositorioUsuario repositorio;
12
13     @Autowired
14     public ServicioUsuario(RepositorioUsuario repositorio) { this.repositorio = repositorio; }
15
16     public List<Empleado> getInformacion() { return this.repositorio.findAll(); }
17
18     public Empleado crearUsuario(Empleado nuevoUsuario) { return this.repositorio.save(nuevoUsuario); }
19
20     public Empleado actualizarEmpleado(@PathVariable Long id, @RequestBody Empleado e) {
21         Empleado empleadoActual = this.repositorio.findById(id).orElseThrow();
22         empleadoActual.setCorreo(e.getCorreo());
23         empleadoActual.setNombre(e.getNombre());
24         empleadoActual.setRolEmpleado(e.getRolEmpleado());
25         empleadoActual.setEmpresaEmpleadoPertenencia(e.getEmpresaEmpleadoPertenencia());
26         return this.repositorio.save(empleadoActual);
27     }
28
29     public Empleado eliminarEmpleado(Long id) {
30         Empleado empleadoActual = this.repositorio.findById(id).orElseThrow();
31         this.repositorio.deleteById(id);
32         return empleadoActual;
33     }
34 }
```

¿Cuáles actividades del sprint se cumplieron?

Se cumplieron todas las actividades y se adiciono servicios para movimientos de dinero.

¿Cuáles no se cumplieron?

No aplica

¿Qué preguntas o dificultades surgieron durante el desarrollo del sprint?

La principal dificultad se presentó debido a una confusión al momento de la elaboración de las bases de datos, el tema fue abordado de diferentes formas por cada tutor de grupo entonces una de las integrantes tuvo que acoplarse.

Otra dificultad que se presentó para la solución del sprint 3 fue que los últimos temas sobre las bases de datos ya fueron vistas casi al final del sprint lo que nos dio poco tiempo para la realización del proyecto ya que la agenda de entrega estaba para antes de la última clase de programación.