
Especificación de Requerimientos de Software

para

Programa Nacional Integral de Sustitución de cultivos ilícitos

Versión 1.0 aprobada

Realizado por Felipe Cardozo, Paola Zárate

Politécnico Grancolombiano

16 de junio de 2018

Tabla de Contenido

Tabla de Contenido	ii
Historial de Cambios	ii
1.Introducción	<u>5</u>
1.1Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	5
1.4 Definiciones, Acrónimos y Abreviaturas	6
1.5 Referencias	6
2 Descripción global del dominio	7
2.1 Perspectivas del producto	7
2.2 Interfaces con el sistema	7
2.3 Interfaces con el usuario	7
2.4 Interfaces con el software	7
2.4 Operaciones	8
3 <i>Requerimientos funcionales del sistema</i>	8
3.1 Requisitos Funcionales	7
3.1.1 Casos de uso del sistema	8
3.2 Requisitos No funcionales	8
4 Catálogo De Requisitos Del Sistema A Desarrollar	21
4.1 Especificación de Tarjetas CRC	21
4.2 Diagrama del dominio	22
4.3 Diagrama de clases	22
4.5 Diagrama de Secuencia	23
4.6 Diagrama de actividades	23
4.6 Patrones Grasp usados en el sistema	25
4.6 Patrones GOF usados en el sistema	26
4.6 Solid	27

Historial de Cambios

Registro del control de cambios en el documento SRS del PNIS (Sistema de Información del programa nacional de sustitución de cultivos ilícitos)

Número Revisión	Fecha	Descripción	Autor	Versión
001	16/06/18	Versión inicial	Felipe Cardozo Paola Zarate	SRS_V1

1. Introducción

En este documento se encuentra la especificación de los requerimientos de la aplicación SI-PNIS, partiendo de la definición de requerimiento tenemos “**especificación de requisitos de software (ERS)** es una descripción completa del comportamiento del sistema que se va a desarrollar”, para lo cual se analizarán y documentarán los casos de uso, el dominio, los requerimientos funcionales entre otros.

1.1. Propósito

Introduzca Contenido Y Borre Cuadro>El presente documento tiene como propósito definir las especificaciones funcionales y del sistema para la implementación de una aplicación que permitirá administrar y consultar la información de los planes de sustitución en Colombia.

Este documento va dirigido al cliente y al equipo de desarrollo, también puede ser usado en el futuro para definir nuevos requerimientos, en el caso que se requiera.

1.2. Alcance

Diseño y desarrollo del sistema SI-PNIS (Sistema de Información del programa nacional de sustitución de cultivos ilícitos)

El SI-PNIS será una aplicación que funcionara en entorno web , que permitirá administrar y consultar toda la información relacionada con los planes de sustitución.

Esta aplicación dará apoyo a los siguientes procesos:

- Administrar usuarios
- Administrar roles
- Administrar terrenos
- Administrar propietarios de terrenos
- Administrar sustituciones

1.3. Personal involucrado

Nombre	Felipe Cardozo
Rol	Programador
Categoría profesional	Ingeniero de sistemas
Responsabilidades	Diseñar y programar la aplicación

Nombre	Paola Zárate
Rol	Programador
Categoría profesional	Ingeniero de sistemas
Responsabilidades	Diseñar y programar la aplicación

Nombre	Luis Gabriel Moreno
Rol	Evaluador
Categoría profesional	
Responsabilidades	Realizar seguimiento al documento

1.4. Definiciones, Acrónimos y Abreviaturas

- IEEE – Institute of Electrical and Electronics Engineers
- ERS - Especificación de requisitos de software
- SI - Sistema de información
- PNIS - Programa nacional integral de sustitución de cultivos de uso ilícito

1.5. Referencias

[1] IEEE Software Requirements Specification Template

2. Descripción global del dominio

En Colombia los altos costos, social, ambiental y político, que representan los cultivos ilícitos fueron la principal motivación para desarrollar una propuesta alterna de sustitución voluntaria. Es importante que exista un programa a lo largo de la nación para que exista un plan a futuro de recuperación ambiental de terrenos que ha sido imposible por la guerra administrar o seguir. Sin embargo, después del conflicto armado existen aun terrenos donde terrenos con cultivos ilícitos se han venido desmontando. Existen algunas propuestas que son basados en modelo agroforestal, estructurado para diferentes especies de terreno. Tal deforestación que sufrieron grandes terrenos en Colombia ha sido históricamente imposible de recuperar debido a la no organizada planta de coca, sin embargo para llegar a la sustitución completa de estos cultivos ilícitos y sus diferentes ecosistemas es importante que la comunidad después del conflicto puedan ayudar a esta reforestación como parte del plan de sustitución de cultivos ilícitos.

Los cultivos ilícitos en Colombia se encuentran ubicados en regiones donde predominan las poblaciones campesinas o indígenas con muy poca presencia del gobierno y es casi nulo que esta presencia que es necesaria en los cultivos o terrenos que harían parte del plan nacional de sustitución tengan algún tipo de seguimiento, control y conocimiento frente a este problema. Desde que la población no cuenta con recursos necesarios para este tipo de control del proyecto se debe iniciar algún plan o proyecto que mantenga un registro que ayude a administrar al estado los diferentes terrenos, parámetros, circunstancias para llevar y continuar el plan de sustitución.

En este sentido es importante retomar estrategias que ayuden al actual gobierno para la erradicación de cultivos ilícitos donde más de 50.000 familias se verían beneficiadas, esta propuesta se considera una alternativa viable para las áreas donde el proceso de sustitución voluntaria garantice un beneficio al gobierno especialmente, indirectamente a la guardia nacional, a los delegados de los municipios afectados y a la comunidad un manejo adecuado y mejor control del plan nacional de sustitución de cultivos ilícitos en Colombia.

El SI-PNIS es un proyecto que pretende cubrir los municipios afectados de esta deforestación que ha sufrido por décadas, se trata de un sistema que permita administrar la información de los cultivos ilícitos que se encuentran en un tipo de sustitución. Los tipos de sustitución son Voluntario que es cuando se llega a un acuerdo con el dueño específicamente de un terreno que tiene o tuvo cultivos ilícitos en el cual, tanto el gobierno como este propietario llegan a un acuerdo firmado y digitalizado y se da una fecha inicial y una fecha final para el término de este proceso de sustitución, en caso de incumplimiento se procederá a una sustitución obligatoria, en este último caso ya bien sea si no existe tal acuerdo desde un inicio o si hay incumplimiento por parte del propietario la guardia nacional hará una erradicación manual. Se deja constancia de igual manera un documento autorizado por alguno de los delegados de zonas que también serán digitalizados independientemente y cargados al igual que la sustitución voluntaria al sistema.

El sistema contará con dos tipos de usuarios uno que será el delegado de algunas zonas, sin embargo es necesario que tengan visibilidad de todos los territorios incluidos en el proceso, y además serán los únicos habilitados para finalizar un proceso de sustitución requiriendo para este el acuerdo digitalizado ya bien sea para obligatorio o voluntario. El otro tipo de usuario es aquel que puede administrar terrenos, administrar planes de sustitución más no podrá realizar la sustitución o finalizar un acuerdo en el sistema.

Finalmente una vez finalizado los acuerdos por el delegado administrativo deberán ser notificados vía email todos los tipo de usuario delegados administrativos mostrando que acuerdo y sustitución finalizó.

De manera general la aplicación tiene dos módulos:

- **Módulo de trabajo:** Contiene el menú con las opciones de cada funcionalidad
- **Módulo de administración:** En este módulo se realiza la creación, consulta y edición de cada entidad del sistema.

2.1. Perspectivas del producto

El presente documento tiene como propósito definir las especificaciones funcionales y del sistema para la implementación de una aplicación que permitirá administrar y consultar la información de los planes de sustitución en Colombia. Este tipo de aplicación no existe en el mercado y las herramientas que usa actualmente el gobierno no tienen contemplado la gestión de los planes de sustitución. Esta aplicación es la primera que representa a través del software el plan de sustitución de cultivos ilícitos en Colombia.

La aplicación está diseñada para que los delegados del PNIS la puedan usar desde cualquier municipio del país y desde cualquier tipo de dispositivo.

2.2. Interfaces con el sistema

De acuerdo a lo mencionado anteriormente, al ser una aplicación nueva no tiene interfaces o sistemas con los cuales interactuar, debido que no existe ningún sistema con el que se pueda integrar.

2.3. Interfaces con el usuario

La interfaz con el usuario consistirá de un conjunto de páginas con formularios, botones, listas, campos de texto. Estos serán construidos específicamente para la aplicación.

2.4. Interfaces con el software

Para el correcto funcionamiento de SI-PNIS , la aplicación debe de interactuar con estos softwares:

Nombre
Spring 4.4
mysql 5.1
jpa 2.2
html
Bootstrap

2.5. Operaciones

En el “Modos de Operación Usuario”s se pueden observar los modos de usuario que puede utilizar el programa y operaciones que tienen estos con respecto a las funciones de la aplicación:

Modos de Operación Usuario	
Usuario	Función
Director	Crear delegados
Delegado	<ul style="list-style-type: none"> ● Crear sustituciones y hacer seguimiento de estas. ● Crear terrenos ● Adjuntas acuerdos

3. Requerimientos funcionales del sistema

3.1 Requisitos funcionales

El sistema quiere permite llevar un control de la información de las sustituciones, debido a que actualmente se hace manualmente.

El sistema permite al usuario delegado y director, crear sustituciones y para esto estos usuarios debieron haber creado previamente un terreno y un propietario.

Los usuario podrán consultar propietarios, consultar terrenos, consultar terrenos, como parte de la tarea de actualización dinámica de la información se puede editar y eliminar los registros.

En la creación de la sustitución el delegado tiene la responsabilidad de adjuntar el acuerdo con las firmas de los propietarios.

3.1.1 Casos de uso del sistema

Aquí se puede ver la funcionalidad básica de SI-PNIS, en esta sección se mencionan los User Stories como las funcionalidades principales que ofrece el sistema

Con el siguiente formato se documentaron los casos de uso:

Use case			
Fecha		Versión	
Nombre		Id	
Objetivo			

Actores	
Entradas	
Salidas	
Pre-condicion es	
Post-Condicio nes	Condición de Éxito
	Condición de Fallo

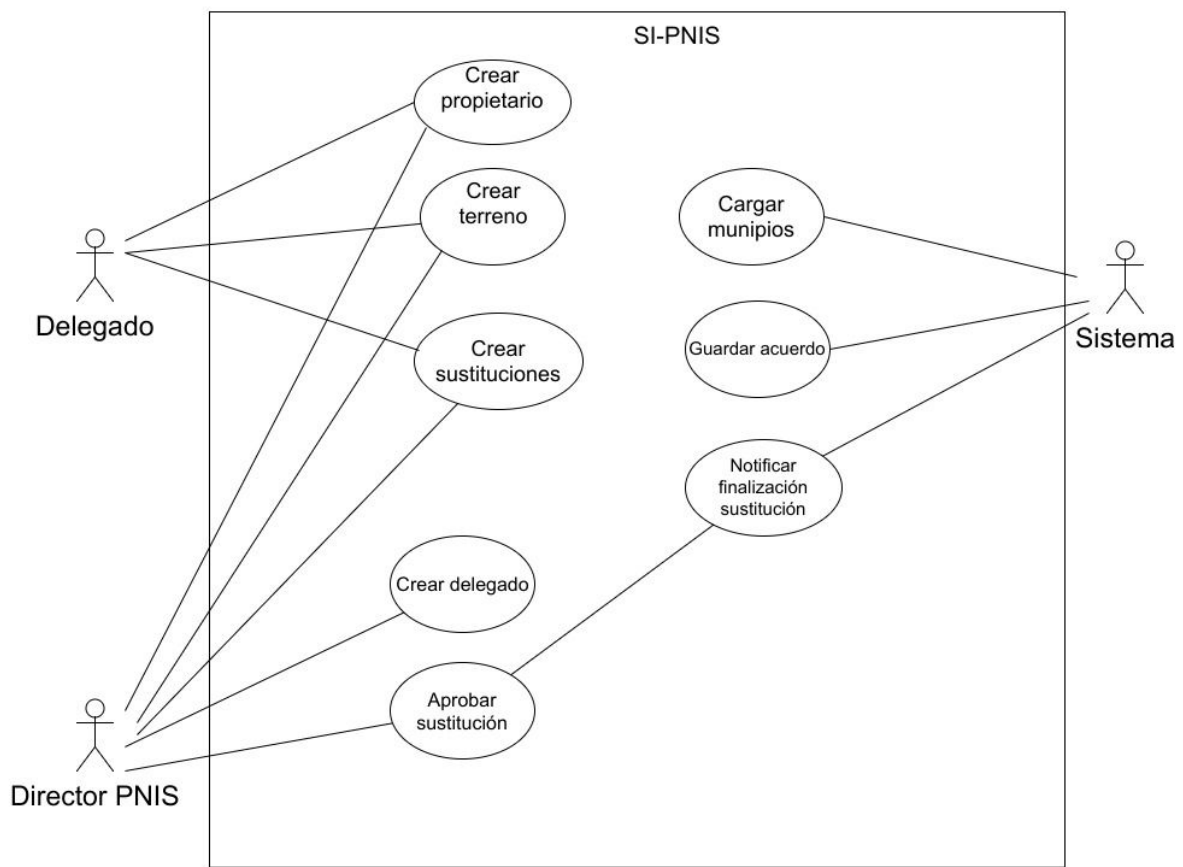
Flujo			
No.	USUARIO	No.	Sistema

Flujo Alternativo			
No.	Actor	No.	Sistema

Relaciones		
Relaciones	Id	Nombre
Extensiones		
Requerimientos asociados		

A continuación ilustramos los casos de uso definidos en el sistema, para comenzar a detallar los casos de uso:

Use Case Diagram



USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	CRUD Usuario	Id	CU001
Objetivo	Director puede registrar en el sistema a los usuarios		
Actores	Director		
Entradas	Información del usuario: nombre, apellidos, número documento y tipo documento.		
Salidas	Alerta en pantalla		
Pre-condicion es	El director debe acceder a la aplicación		
Post-Condicio nes	<i>Condición de Éxito</i>		
	Registro de usuario		
	<i>Condición de Fallo</i>		
	No se registra usuario		

--	--

Flujo			
No.	USUARIO	No.	Sistema
01	Director Ingresa al sistema		
02	Director ingresa a Usuarios	03	Muestra listado de usuarios
04	Hace clic en "Crear usuario"	05	Muestra formulario con los datos a llenar
06	Llenar campos y hace clic en "enviar"		
		07	Valida información
		08	Guarda Información usuario
		09	Muestra mensaje "el usuario se guardó exitosamente"

Flujo Alternativo			
No.	Actor	No.	Sistema
01	Director Ingresa al sistema		
02	Director ingresa a Usuarios	03	Muestra listado de usuarios
04	Hace clic en "Crear usuario"	05	Muestra formulario con los datos a llenar
06	Llenar campos y hace clic en "enviar"		
		07	Valida información
		08	Rechaza Validación

Relaciones		
Relaciones	Id	Nombre
Extensiones		
Requerimientos asociados		

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Crear propietario	Id	CU002
Objetivo	El delegado puede crear propietarios		
Actores	delegado		
Entradas	email y contraseña		
Salidas	Alerta "Propietario creado exitosamente!"		
Pre-condiciones	El delegado debe haber ingresado		
Post-Condiciones	<i>Condición de Éxito</i>		
	Listado de propietarios con el nuevo propietario		
	<i>Condición de Fallo</i>		
	No crea el propietario		

Flujo			
No.	USUARIO	No.	Sistema
01	Ingreso de email		
02	Ingreso contraseña		
03	Presiona botón "Login"		
		04	Validación datos
		05	Menú principal
06	Selecciona opción Propietarios Terrenos		
		07	Muestra botón "Crear Propietario terreno"
08	Presiona botón "Crear Propietario terreno"		
		09	Muestra formulario con campos a llenar
10	Ingresar tipo de identificación, número de identificación, nombre		
11	Presiona botón Enviar		
		12	Muestra mensaje "Propietario creado exitosamente!"
		13	Actualiza lista de propietarios

Flujo Alternativo			
No.	USUARIO	No.	Sistema

01	Ingreso de email		
02	Ingreso contraseña		
03	Presiona botón "Login"		
		04	Validación datos
		05	Menú principal
06	Selecciona opción Propietarios Terrenos		
		07	Muestra botón "Crear Propietario terreno"
08	Presiona botón "Crear Propietario terreno"		
		09	Muestra formulario con campos a llenar
10	Ingresa tipo de identificacion, numero de identificacion, nombre		
11	Presiona boton Enviar		
		12	Muestra mensaje de error

Relaciones		
Relaciones	Id	Nombre
Extensiones		
Requerimientos asociados	CU001	CRUD Usuario

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Crear terreno	Id	CU003
Objetivo	El delegado crea un terreno		
Actores	Delegado		
Entradas	email y contraseña		
Salidas	Mensaje: Terreno creado exitosamente!		

Pre-condicion es	El delegado debe haber ingresado
Post-Condicio nes	<i>Condición de Éxito</i>
	Mensaje: Terreno creado exitosamente!
	<i>Condición de Fallo</i>
	Mensaje: "no se pudo guardar el terreno"

Flujo			
No.	USUARIO	No.	Sistema
01	Ingreso de email		
02	Ingreso contraseña		
03	Presiona botón "Login"		
		04	Validación datos
		05	Menú principal
06	Selecciona opción Propietarios Terrenos		
		07	Muestra botón "Crear Terreno"
08	Presiona botón "Crear Terreno"		
		09	Muestra formulario con campos a llenar
10	Ingresa información		
11	Presiona boton Enviar		
		12	Muestra mensaje exitoso

Flujo Alternativo			
No.	Actor	No.	Sistema
01	Ingreso de Usuario		
02	Ingreso contraseña		
03	Presiona botón "Login"		
		04	Validación datos
		05	Menú principal
06	Selecciona opción Terrenos		
		07	Muestra listado vacío de Terrenos

Relaciones		
Relaciones	Id	Nombre
Extensiones		
Requerimientos asociados	CU001	CRUD Usuario
	CU002	Crear propietario

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Crear sustitución	Id	CU004
Objetivo	El delegado crea una sustitución		
Actores	Delegado		
Entradas	email y contraseña		
Salidas	Mensaje: Sustitución creada exitosamente!		
Pre-condicion es	El delegado debe haber ingresado		
Post-Condicio nes	<i>Condición de Éxito</i>		
	Mensaje: Sustitución creada exitosamente!		
	<i>Condición de Fallo</i>		
	Mensaje: No se pudo guardar la sustitución		

Flujo			
No.	USUARIO	No.	Sistema
01	Ingreso de email		
02	Ingreso contraseña		
03	Presiona botón "Login"		
		04	Validación datos
		05	Menú principal
06	Selecciona opción Sustituciones		
		07	Muestra botón "Crear Sustitucion"
08	Presiona botón "Crear Sustitucion"		
		09	Muestra formulario con campos a llenar
10	Ingresa información		
11	Presiona boton Enviar		
		12	Muestra mensaje exitoso

Flujo Alternativo			
Flujo			
No.	USUARIO	No.	Sistema

01	Ingreso de email		
02	Ingreso contraseña		
03	Presiona botón "Login"		
		04	Validación datos
		05	Menú principal
06	Selecciona opción Sustituciones		
		07	Muestra botón "Crear Sustitucion"
08	Presiona botón "Crear Sustitución"		
		09	Muestra formulario con campos a llenar
10	Ingresa información		
11	Presiona boton Enviar		
		12	Muestra mensaje error

Relaciones		
Relaciones	Id	Nombre
Extensiones		
Requerimientos asociados	CU001	CRUD Usuario
	CU002	Crear propietario
	CU006	Cargar municipios

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Aprobar sustitucion	Id	CU004
Objetivo	El director aprueba una sustitución		
Actores	Director		
Entradas	email y contraseña		
Salidas	Mensaje: Sustitucion aprobada!		
Pre-condicion es	El director debe haber ingresado a la opcion "Sustituciones"		
Post-Condicio nes	<i>Condición de Éxito</i>		
	Mensaje: Sustitucion aprobada!		

	Condición de Fallo
	Mensaje: No se pudo aprobar la sustitución

Flujo			
No.	USUARIO	No.	Sistema
01	Selecciona opción Sustituciones		
		02	Muestra listado de sustituciones con un boton "aprobar"
03	Presiona botón "Aprobar"		
		04	Muestra mensaje "¿Esta seguro que desea aprobar esta sustitucion?"
		05	
06	Selecciona opción "Si"		
		07	Cambia el estado de la sustitución a aprobada
		08	Lista las sustituciones actualizadas

Flujo Alternativo			
Flujo			
No.	USUARIO	No.	Sistema
01	Selecciona opción Sustituciones		
		02	Muestra listado de sustituciones con un boton "aprobar"
03	Presiona botón "Aprobar"		
		04	Muestra mensaje "¿Esta seguro que desea aprobar esta sustitucion?"
		05	
06	Selecciona opción "No"		
		07	Muestra listado de sustituciones con un boton "aprobar"

Relaciones		
Relaciones	Id	Nombre
Extensiones		
Requerimientos asociados	CU001	CRUD Usuario
	CU002	Crear propietario
	CU006	Cargar municipios

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Cargar municipios	Id	CU006
Objetivo	Cargar los municipios para consulta		
Actores	Sistema		
Entradas	Tabla de municipios		
Salidas	Listado de municipios		
Pre-condiciones	El delegado debe haber ingresado a la opcion "Crea Terreno"		
Post-Condiciones	<i>Condición de Éxito</i>		
	Información municipio		
	<i>Condición de Fallo</i>		
	No retorna informacion del munipio		

Flujo			
No.	USUARIO	No.	Sistema
01	Selecciona opción Terrenos		
		02	Muestra listado de terrenos con un boton "Crear terreno"
03		03	Muestra formulario con los datos a llenar
		04	Muestra campo Municipio con la informacion de los municipios
		05	
06	Ingresa el texto de un municipio		
		07	Retorna el id y descripcion del municipio

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Cargar municipios	Id	CU006
Objetivo	Cargar los municipios para consulta		
Actores	Sistema		

Entradas	Tabla de municipios
Salidas	Listado de municipios
Pre-condiciones	El delegado debe haber ingresado a la opcion "Crea Terreno"
Post-Condiciones	<i>Condición de Éxito</i>
	Información municipio
	<i>Condición de Fallo</i>
	No retorna informacion del munipio

Flujo			
No.	USUARIO	No.	Sistema
01	Selecciona opción Terrenos		
		02	Muestra listado de terrenos con un boton "Crear terreno"
03		03	Muestra formulario con los datos a llenar
		04	Muestra campo Municipio con la informacion de los municipios
		05	
06	Ingresa el texto de un municipio		
		07	Retorna el id y descripcion del municipio

USE CASE			
Fecha	17/06/2018	Versión	V002
Nombre	Cargar municipios	Id	CU007
Objetivo	Guardar acuerdo		
Actores	Sistema		
Entradas	Documento con el Acuerdo		
Salidas	Guardar el acuerdo		
Pre-condiciones	El delegado debe haber ingresado a la opcion "Crea Sustitucion"		
Post-Condiciones	<i>Condición de Éxito</i>		
	Información municipio		
	<i>Condición de Fallo</i>		

	No retorna informacion del munipio
--	------------------------------------

Flujo			
No.	USUARIO	No.	Sistema
01	Selecciona opción Terrenos		
		02	Muestra listado de terrenos con un boton "Crear terreno"
03		03	Muestra formulario con los datos a llenar
		04	Muestra campo Municipio con la informacion de los municipios
		05	
06	Ingresa el texto de un municipio		
		07	Retorna el id y descripcion del municipio

3.2 Requisitos No funcionales

Esta sección está dedicada a verificar los factores que no son visibles para el usuario, pero que ayudan a que la aplicación sea eficiente, mantenible y eficiente.

1. Almacenar la información en una base datos.
2. Base de datos no relacional
3. Versionar el codigo:El codigo fuente de la aplicación se encuentra almacenado en github.
4. Arquitectura orientada a servicios rest.

4 Catálogo De Requisitos Del Sistema A Desarrollar

4.1 Especificación de Tarjetas CRC

Como parte de la tarea de desarrollo del sistema, se eligió las posibles clases, se definió sus responsabilidades y sus colaboradores.

A continuación listamos las tarjetas del SI-PNIS:

Class Name: Usuario	
SuperClass / SubClass:	
Responsabilidades	Colaboradores

Conoce id Conoce nombre Conoce numeroIdentificacion Conoce email Conoce password Conoce idTipoUsuario	
--	--

Class Name: TipoSustitucion	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce nombre Conoce codigo Conoce id	

Class Name: Departamento	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce id Conoce idPais Conoce latitud Conoce longitud Conoce nombre	

Class Name: Municipio	
SuperClass / SubClass:	
Responsabilidades	Colaboradores

Conoce id Conoce nombre Conoce latitud Conoce longitud Conoce departamento	Departamento
--	--------------

Class Name: Propietario	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce id Conoce nombre Conoce numeroidentificacion Conoce tipoDocumento	

Class Name: Terreno	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce id Conoce latitud Conoce longitud Conoce tamano Conoce nombre	Conoce Propietario Conoce Municipio

Class Name: Sustitucion	
SuperClass / SubClass:	
Responsabilidades	Colaboradores

Conoce id Conoce fechaInicio Conoce fechaFin Conoce estado	Usuario Terreno TipoSustitucion Acuerdo
---	--

Class Name: Acuerdo	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce id Conoce fechaAcuerdo Conoce aprobado	

Class Name: AcuerdoRepository	
SuperClass: CrudRepository	
Responsabilidades	Colaboradores
Hacer los Crud para Acuerdo	Acuerdo

Class Name: PropietarioRepository	
SuperClass: CrudRepository	
Responsabilidades	Colaboradores
Hacer los Crud para Propietario findByNumeroidentificacion para obtener Propietario	Propietario

--	--

Class Name: SustitucionRepository	
SuperClass: CrudRepository	
Responsabilidades	Colaboradores
Hacer los Crud para Sustitucion	Sustitucion

Class Name: TerrenoRepository	
SuperClass: CrudRepository	
Responsabilidades	Colaboradores
Hacer los Crud para Terreno findAllByPropietario para retornar lista de terrenos	Terreno

Class Name: TipoSustitucionRepository	
SuperClass: CrudRepository	
Responsabilidades	Colaboradores
Hacer los Crud para TipoSustitucion findByNombre retornar tipo sustitucion	TipoSustitucion

Class Name: UsuarioRepository	
SuperClass: CrudRepository	

Responsabilidades	Colaboradores
Hacer los Crud para Usuario findByEmailAndPassword para logear al sistema findByEmail retornar el usuario findAllEmailByTipoUsuario retorna la lista de emails dado el tipo de usuario 1 o 2	Usuario

Class Name: LoginService	
SuperClass: CrudRepository	
Responsabilidades	Colaboradores
Login para autenticar usuarios	UsuarioRepository

Class Name: DefaultSustitucionService	
SuperClass: SustitucionService	
Responsabilidades	Colaboradores
getAllSustituciones createSustitucion updateSustitucion deleteSustitucion getSustitucionById	SustitucionRepository TerrenoRepository UsuarioRepository TipoSustitucionRepository SustitucionBuilder SustitucionRequestDTO Sustitucion

Class Name: AcuerdoVoluntarioService	
SuperClass: AcuerdoService	
Responsabilidades	Colaboradores

Realizar el acuerdo Finalizar el plan de sustitucion Obtener los email de tipo usuario tipo 1 y registrarlos para ser notificados	SustitucionRepository TipoSustitucionRepository UsuarioRepository AcuerdoRepository RealizarAcuerdoDTO RealizarAcuerdoBuilder Sustitucion MensajeDTO NotificarDelegados DelegadoObserver Observer
---	---

Class Name: RealizarAcuerdoBuilder	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
buildSustitucion buildFile buildArchivo buildTipoSustitucion buildDelegado buildEstado buildAcuerdo	SustitucionRepository TipoSustitucionRepository UsuarioRepository AcuerdoRepository Sustitucion FileBridge

Class Name: FileBridge	
SubClass: MacFileBridge, WindowsFileBridge	
Responsabilidades	Colaboradores
buildArchivo guardar en el repositorio buildCodigo buildIdFile	

Class Name: SustitucionBuilder	
SuperClass:	
Responsabilidades	Colaboradores

buildTerreno builDelegado buildTipoSustitucion buildEstado buildFechas Build construye una sustitución	TerrenoRepository UsuarioRepository TipoSustitucionRepository Sustitucion
---	--

Class Name: Observer	
SubClass: DelegadoObserver	
Responsabilidades	Colaboradores
Update actualiza los delegado	

Class Name: DelegadoObserver	
SuperClass: Observer	
Responsabilidades	Colaboradores
Conoce el subject Actualiza a los delegados	

Class Name: Subject	
SubClass: NotificarDelegados	
Responsabilidades	Colaboradores
registerObserver notifyObserver unRegisterObserver getEmail lista de emails getSustitucion	

Class Name: NotificarDelegados	
SuperClass: NotificarDelegados	
Responsabilidades	Colaboradores
Conoce delegados Conoce emails registerObserver notifyObserver unRegisterObserver getEmail lista de emails getSustitucion	

Class Name: FileDTO	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce mensaje	

Class Name: LoginDTO	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce email Conoce password	

Class Name: MensajeDTO	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce mensaje	

Class Name: RealizarAcuerdoDTO	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce idSustitucion Conoce idTipoSustitucion Conoce idDelegado Conoce estado Conoce file Conoce aprobado	MultipartFile (Spring)

Class Name: SustituirRequestDTO	
SuperClass / SubClass:	
Responsabilidades	Colaboradores
Conoce idSustitucion Conoce idTipoSustitucion Conoce idDelegado Conoce estado Conoce aprobado	

4.2 Diagrama del dominio

En esta sección se presenta el modelo del dominio, un **modelo de dominio**, es un modelo conceptual de todos los temas relacionados con un problema específico, en el que se describen las entidades, sus atributos, sus roles y relaciones, además de las restricciones que rigen el dominio del problema.

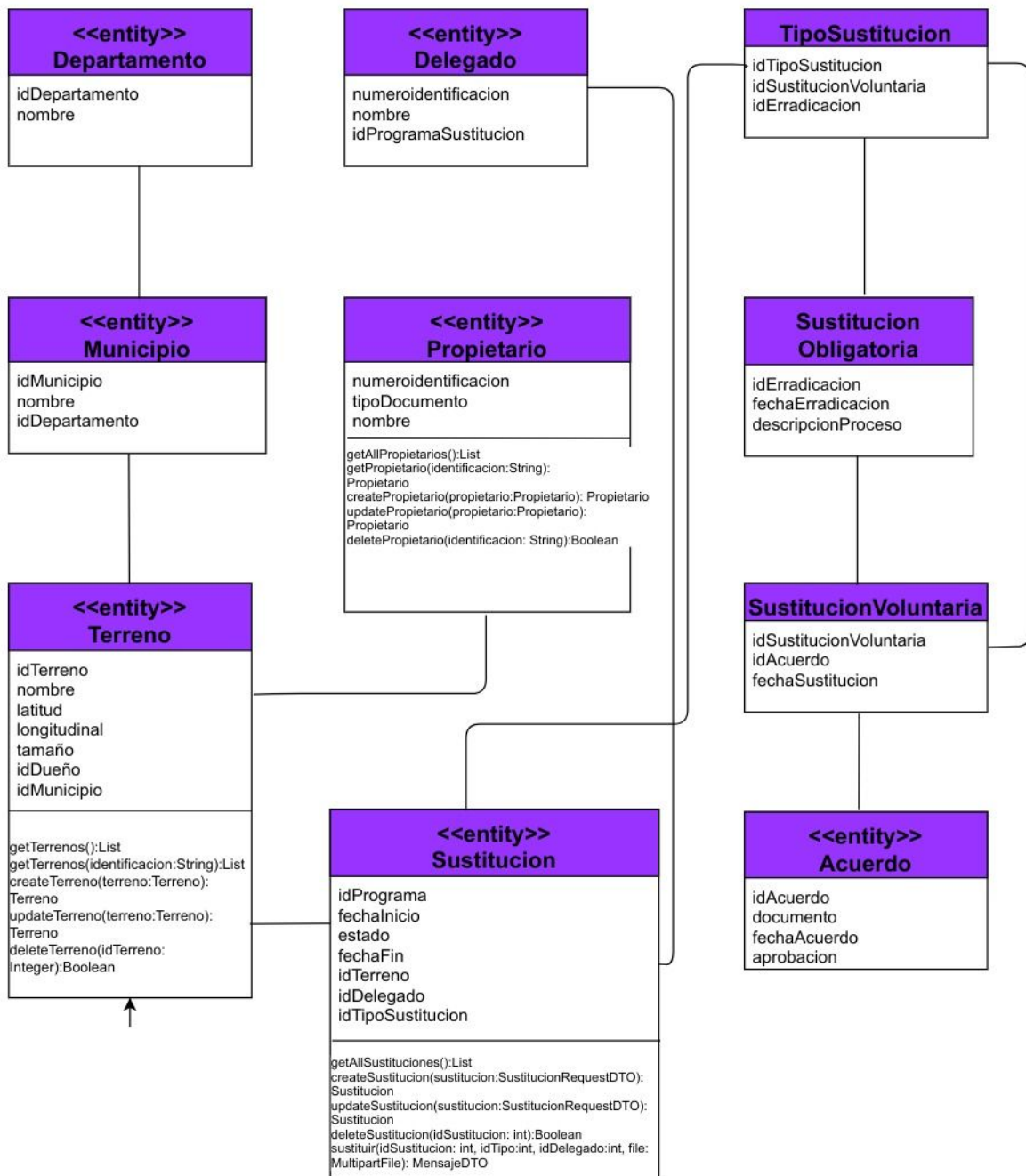
Vamos al crear el dominio del problema, con el objetivo de representar el vocabulario y los conceptos clases del sistema.

Además de igual forma, también describiremos cada uno de los elementos en el modelo del dominio a través de su documentación.

El modelo de dominio de SI-PNIS lo constituyen un total de 10 entidades.

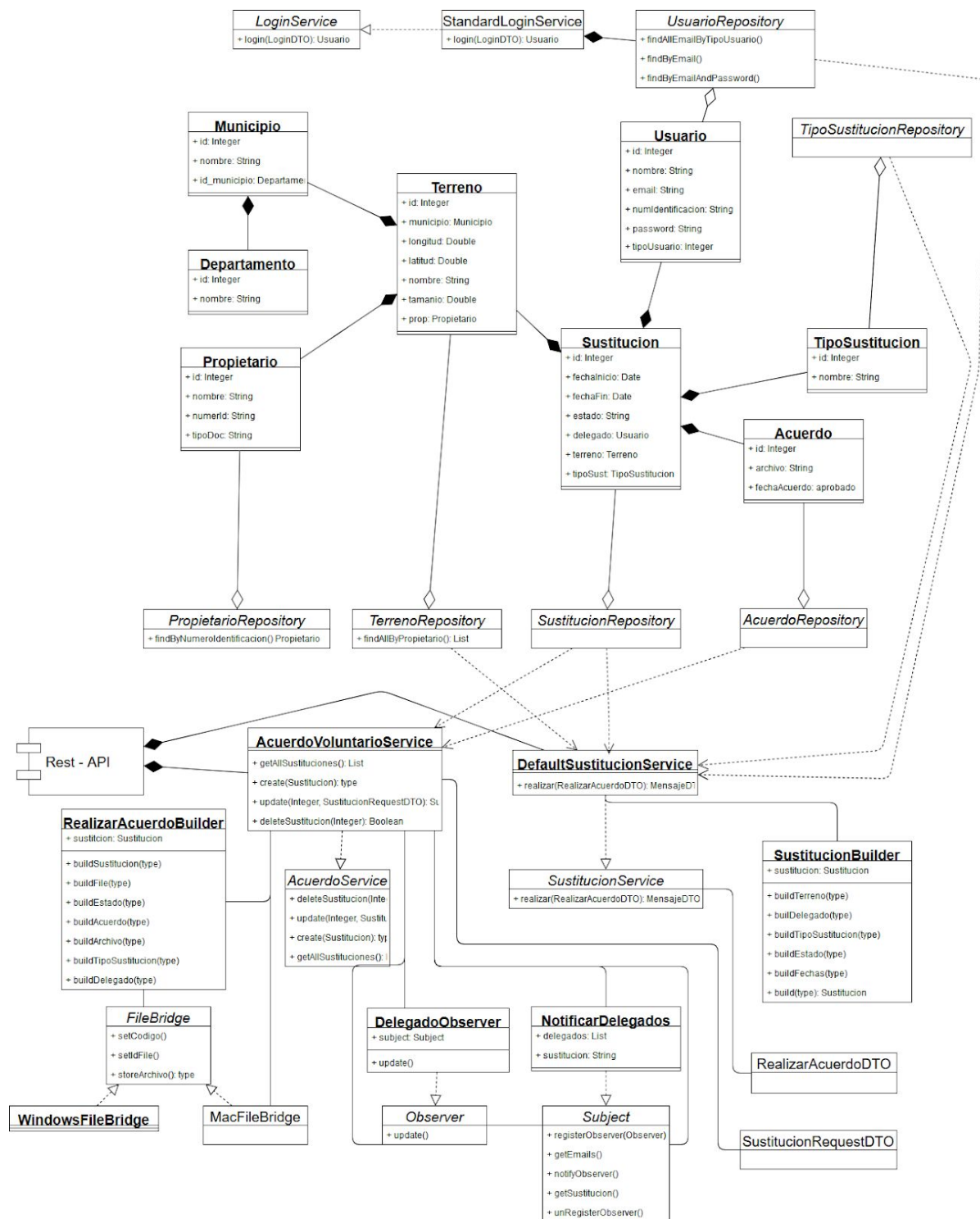
En la siguiente ilustración se muestra el modelo del dominio que conforma el núcleo de entidades que están relacionados en la aplicación:

Domain model



4.2 Diagrama de clases

La aplicacion se dividio en diferentes paquetes para mantener una



4.4 Diagrama de Secuencia

Se realizó el diagrama de actividades de los principales flujos del sistema que son, el login de un usuario, crear una sustitución, crear un acuerdo dentro de la sustitución, realizar el acuerdo en el sistema y editar un terreno.

Diagrama de secuencia del proceso de login

Sequence Diagram

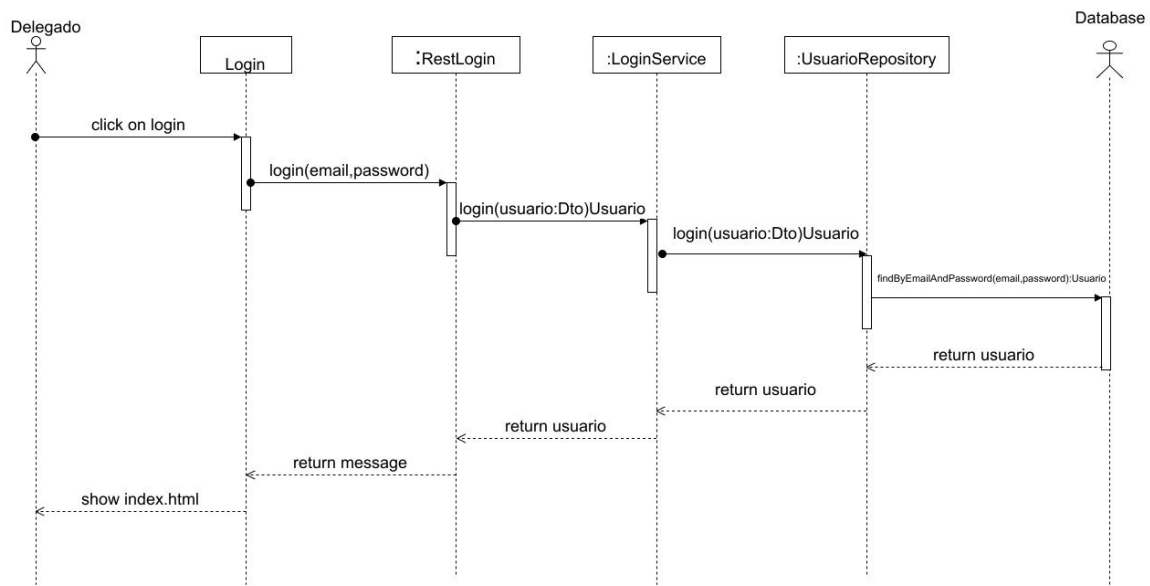


Diagrama de secuencia del proceso de creación de una sustitución

Sequence Diagram

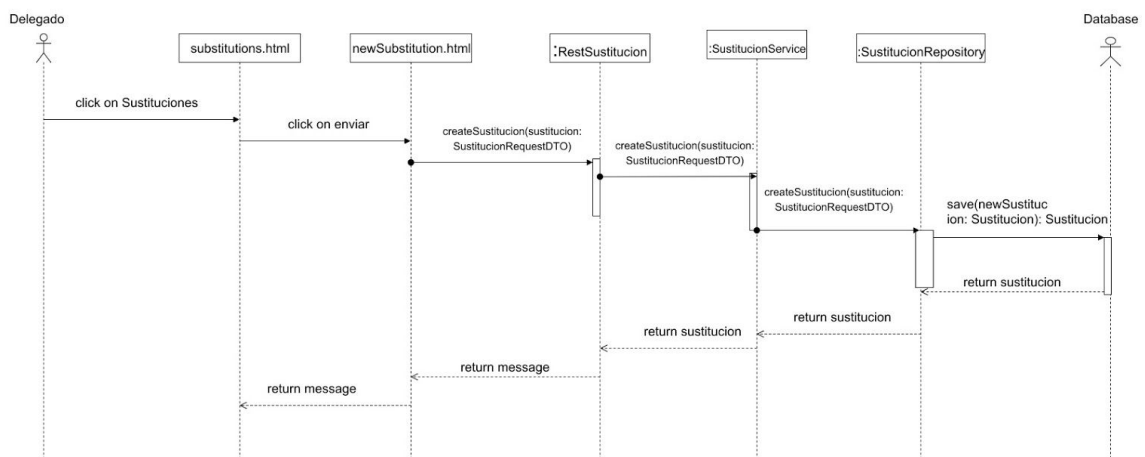


Diagrama de secuencia del proceso de crear sustitución

Sequence Diagram

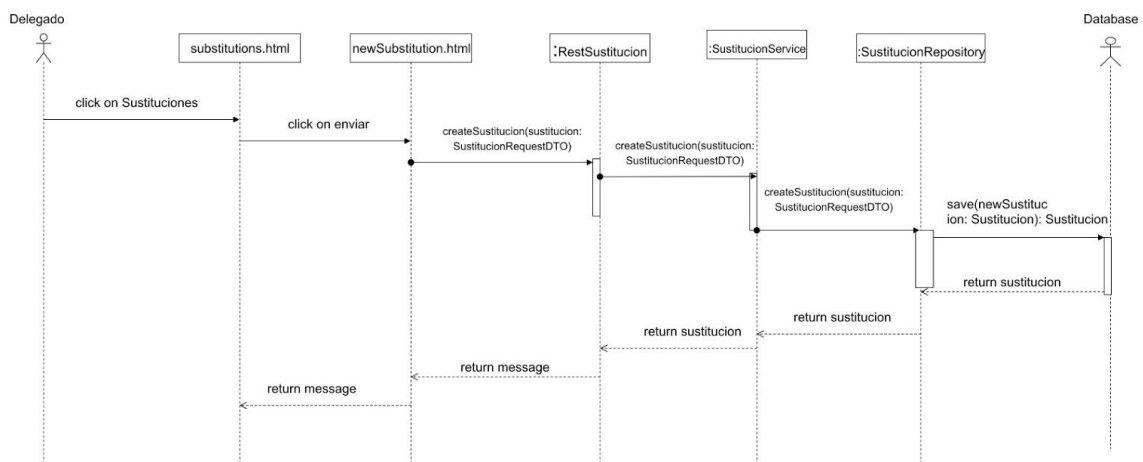


Diagrama de secuencia del proceso de crear acuerdo

Sequence Diagram

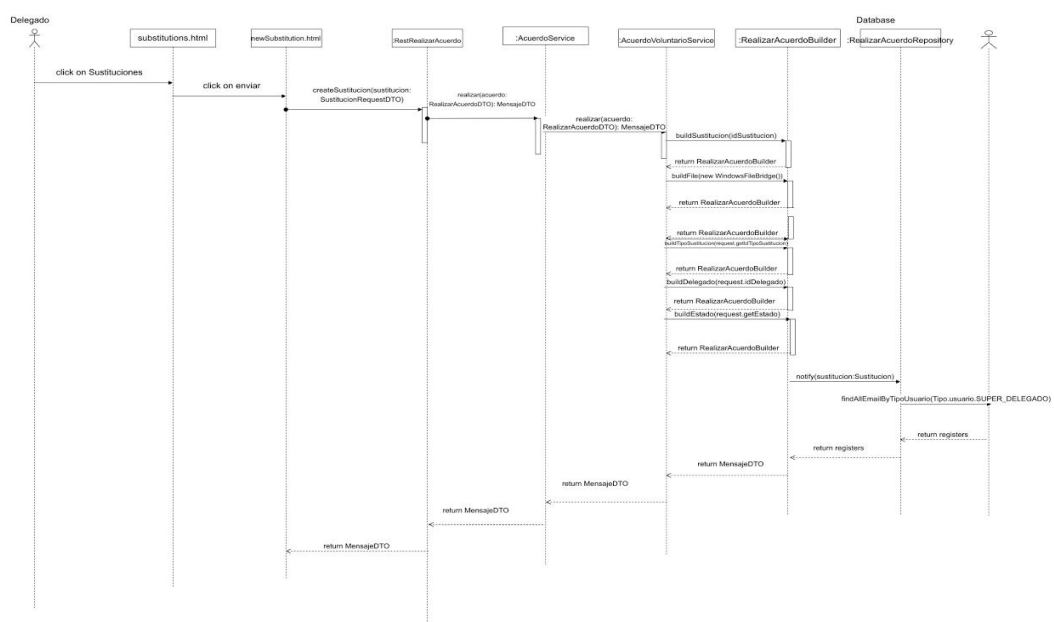
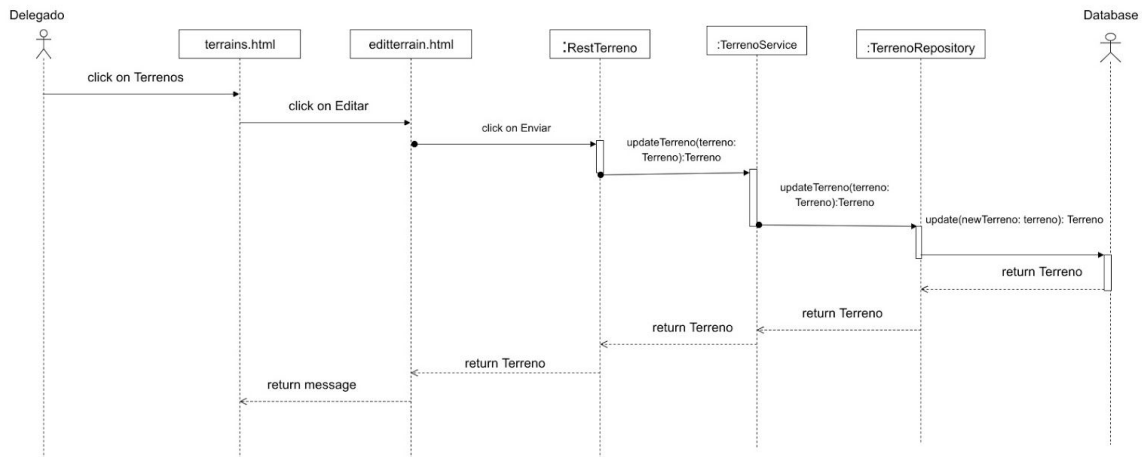
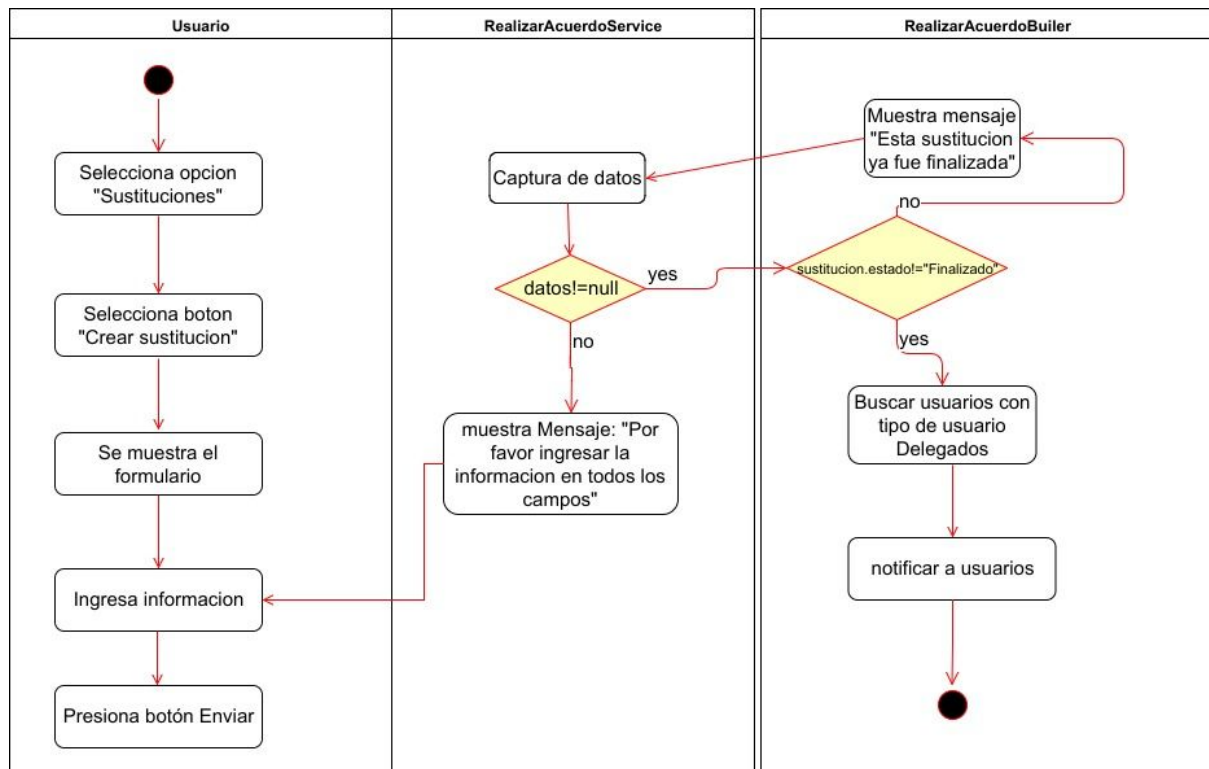


Diagrama de secuencia del proceso de editar terreno
Sequence Diagram



4.5 Diagrama de actividades

Se realizó el diagrama de actividades para el flujo de creación de sustituciones, debido a que es el núcleo del sistema, en este proceso tenemos validaciones relacionada con la captura de datos en el formulario al crear una sustitución, también se muestra como al finalizar la sustitución se busca los usuarios y se notifica a estos.



4.6 Patrones Grasp usados en el sistema

- Controller
- Alta cohesión
- bajo acoplamiento
- polimorfismo

4.7 Patrones GOF usados en el sistema

Strategy

```

1 Loader.java
2 package com.pnis.loader;
3 import java.io.FileNotFoundException;
4
5 public interface Loader {
6
7     public void loadFile() throws FileNotFoundException;
8
9 }
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

```

1 LoaderMunicipios.java
2 package com.pnis.loader;
3 import java.io.File;
4
5 public class LoaderMunicipios implements Loader {
6
7     private MunicipioRepository municipioRepository;
8
9     public LoaderMunicipios(MunicipioRepository municipioRepository) {
10         this.municipioRepository = municipioRepository;
11     }
12
13     @Override
14     public void loadFile() throws FileNotFoundException {
15         File file = new File("C:\\Users\\rene.cardozo\\Documents\\");
16         Scanner sc = new Scanner(file);
17
18         Municipio municipio = new Municipio();
19
20         while( sc.hasNextLine() ) {
21             String line = sc.nextLine();
22             String input[] = line.split(",");
23             municipio.setId(Integer.parseInt(input[0]));
24             municipio.setNombre(input[1]);
25             if( input.length>4 ) municipio.setLatitud(Double.parseDouble(input[2]));
26             if( input.length>4 ) municipio.setLongitud(Double.parseDouble(input[3]));
27             Departamento d = new Departamento();
28             d.setId(Integer.parseInt(input[2]));
29             municipio.setDepartamento(d);
30             municipioRepository.save(municipio);
31         }
32     }
33 }

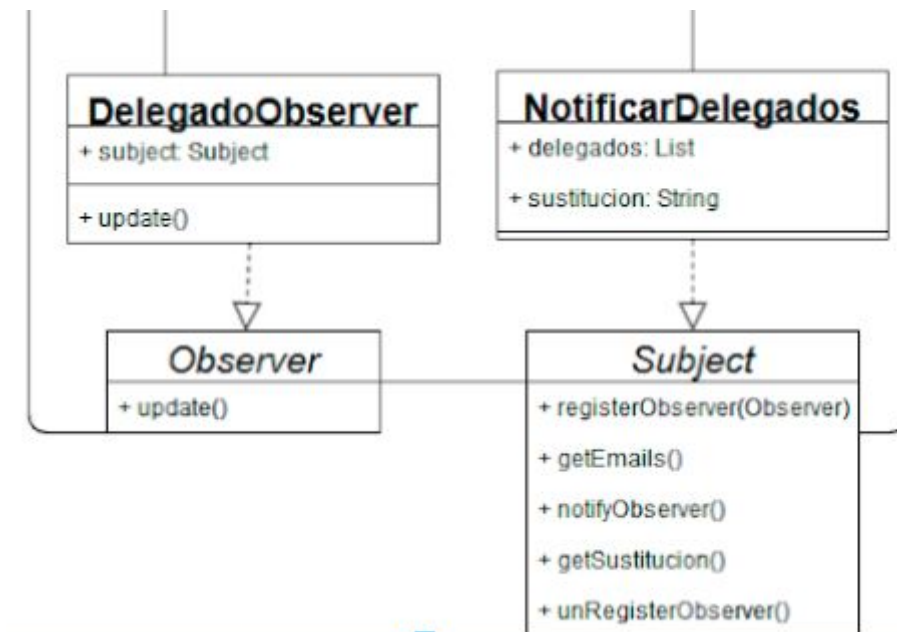
```

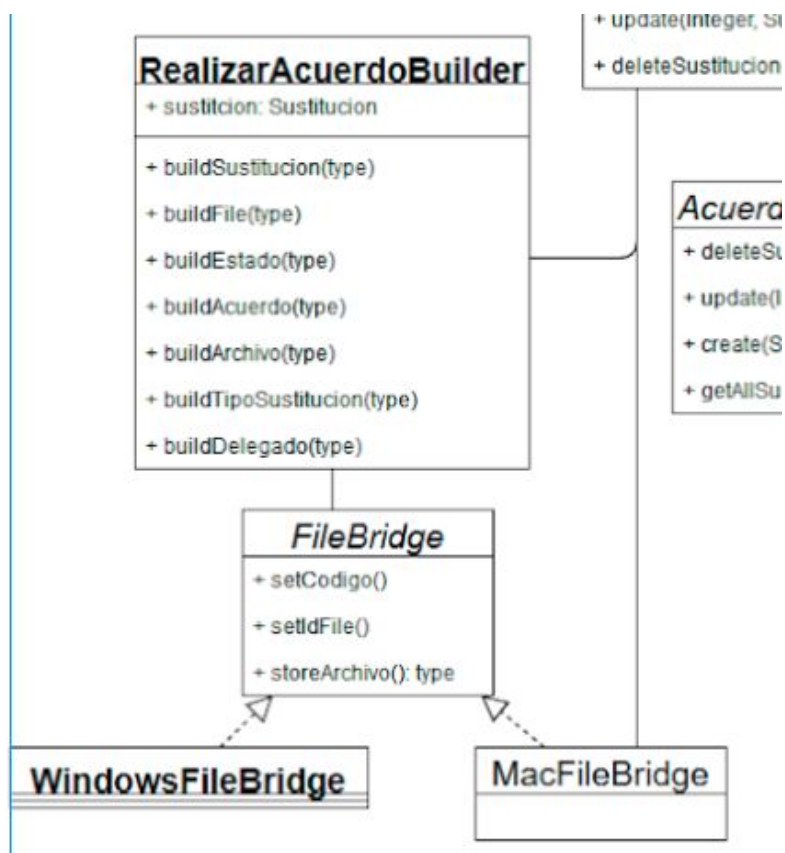
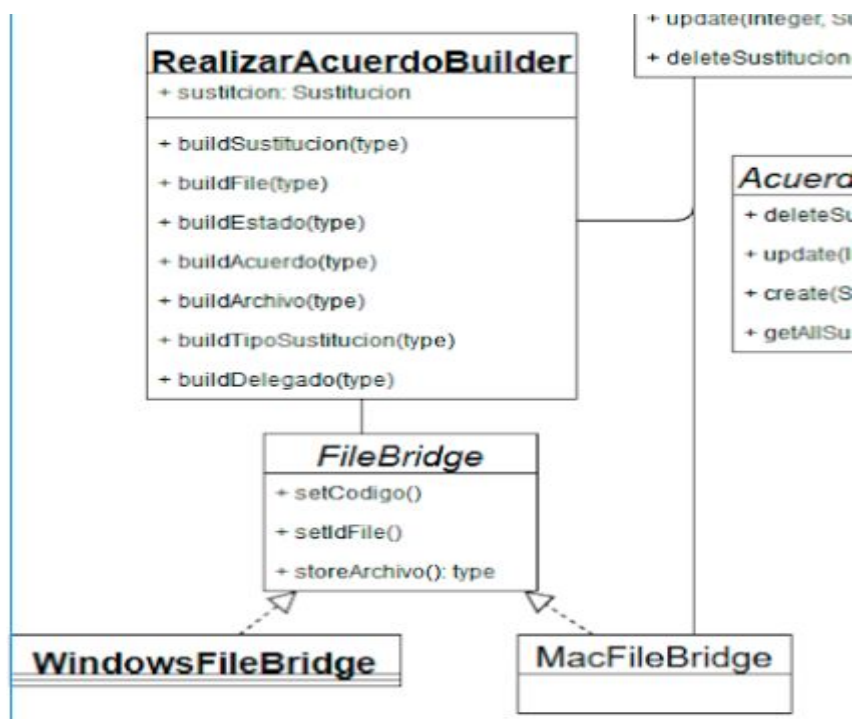
```

1 LoaderDeptos.java
2 package com.pnis.loader;
3 import java.io.File;
4
5 @Component
6 public class LoaderDeptos implements Loader {
7
8     private DepartamentosRepository departamentosRepository;
9
10     public LoaderDeptos(DepartamentosRepository departamentosRepository) {
11         this.departamentosRepository = departamentosRepository;
12     }
13
14     @Override
15

```

observer



Builder**bridge**

4.8 Solid

Principio de responsabilidad única

Principio de inversión de dependencia

Principio de abierto/cerrado