

I.1. PRESENTACION DEL PROYECTO

I.1.1. TITULO

“Control de riego automático para la plaza “Panamericano” a través de un Sistema de Gestión y la Plataforma Arduino”.

I.1.2. AREA DEL PROYECTO

Desarrollo de Sistemas de Información y Robótica Aplicada

I.1.3. DIRECTOR DEL PROYECTO

Copa Apellido Paterno	Mamani Apellido Materno	Ibsen Paul Nombre	7204256 C.I.
Estudiante Profesión	Ingeniería Informática Carrera ó Unidad	Ciencias y Tecnología Facultad:	
Teléf. Oficina	72972778 Celular	Paul.copa@gmail.com Correo electrónico	Firma

TABLA 1: DIRECTOR DEL PROYECTO

I.1.4. INSTITUCION COOPERANTE

Nombre: Ornato Publico			
Dirección: Barrio 6 de Agosto Calle Capitán Carrasco		Telef. Oficina: 6643216	
Nombre y Apellidos	Cargo	C.I.	Firma
Ing. Carlos Burvega M.	Director de Ornato	4141014	

TABLA 2: INSTITUCION COOPERANTE

I.1.5. ACTIVIDADES

Responsable	Actividades
Director	Recopilación de la información requerida para la elaboración del sistema.
Copa Mamani Ibsen Paul	Análisis y Diseño del Sistema. Desarrollo del Software y Hardware Fase de Pruebas del Sistema y Hardware. Diseño y Documentación del manual de usuario. Diseño de la socialización del Sistema.

TABLA 3: ACTIVIDADES

I.2. DESCRIPCION DEL PROYECTO

I.2.1. RESUMEN EJECUTIVO

El presente proyecto significa un aporte a mejorar el riego de esta empresa mediante la implementación de un sistema informático que permita controlar de forma correcta y fácil el riego a diferentes áreas verdes de una plaza y toda la información que esto conlleva. Asimismo, se busca mejorar la gestión de control de manera que la información de los sensores esté disponible en todo momento y la consulta de datos sea más rápida y efectiva. Todas estas acciones permitirán que el manejo y administración de información tan compleja se realice de forma más sencilla y rápida. Se contempla la socialización del proyecto lo que permitirá que los beneficiarios puedan hacer un uso correcto del producto y comprender la importancia de la implementación de las herramientas tecnológicas en el desempeño de sus labores. La implementación de este proyecto permitirá que el manejo y administración de información tan compleja se realice de forma más sencilla y rápida. Se contempla la socialización del proyecto lo que permitirá que los beneficiarios puedan hacer un uso correcto del producto y

comprender la importancia de la implementación de las herramientas tecnológicas en el desempeño de sus labores.

I.2.2. DESCRIPCION Y FUNDAMENTACION DEL PROYECTO

Este proyecto consta de dos componentes:

- Un sistema informático que consta de software y hardware que permitirá mejorar la administración de Riego el cual se encargará del control de aspersores y la gestión de Usuarios.
- Una estrategia de socialización para dar a conocer el proyecto a los funcionarios de la empresa y se resalten los beneficios que implica el implementar este proyecto.

La realización de estos componentes es necesaria ya que permitirá que “Ornato Público” pueda ofrecer un servicio de mayor calidad a la población en general. En la actualidad la empresa no cuenta con un sistema informático para la realización de sus operaciones; esto se ve reflejado en la dificultad de realizar un control eficiente al servicio de riego que ofrece, la empresa no cuenta con un registro de usuarios y mucho menos con un control de los aspersores.

También es necesario ejecutar una estrategia de socialización que permita que los beneficiarios sean capaces de utilizar el sistema informático y puedan comprender las ventajas que representa el mismo al momento de realizar su trabajo.

I.2.2.1. ANALISIS DE CAUSAS DEL PROBLEMA

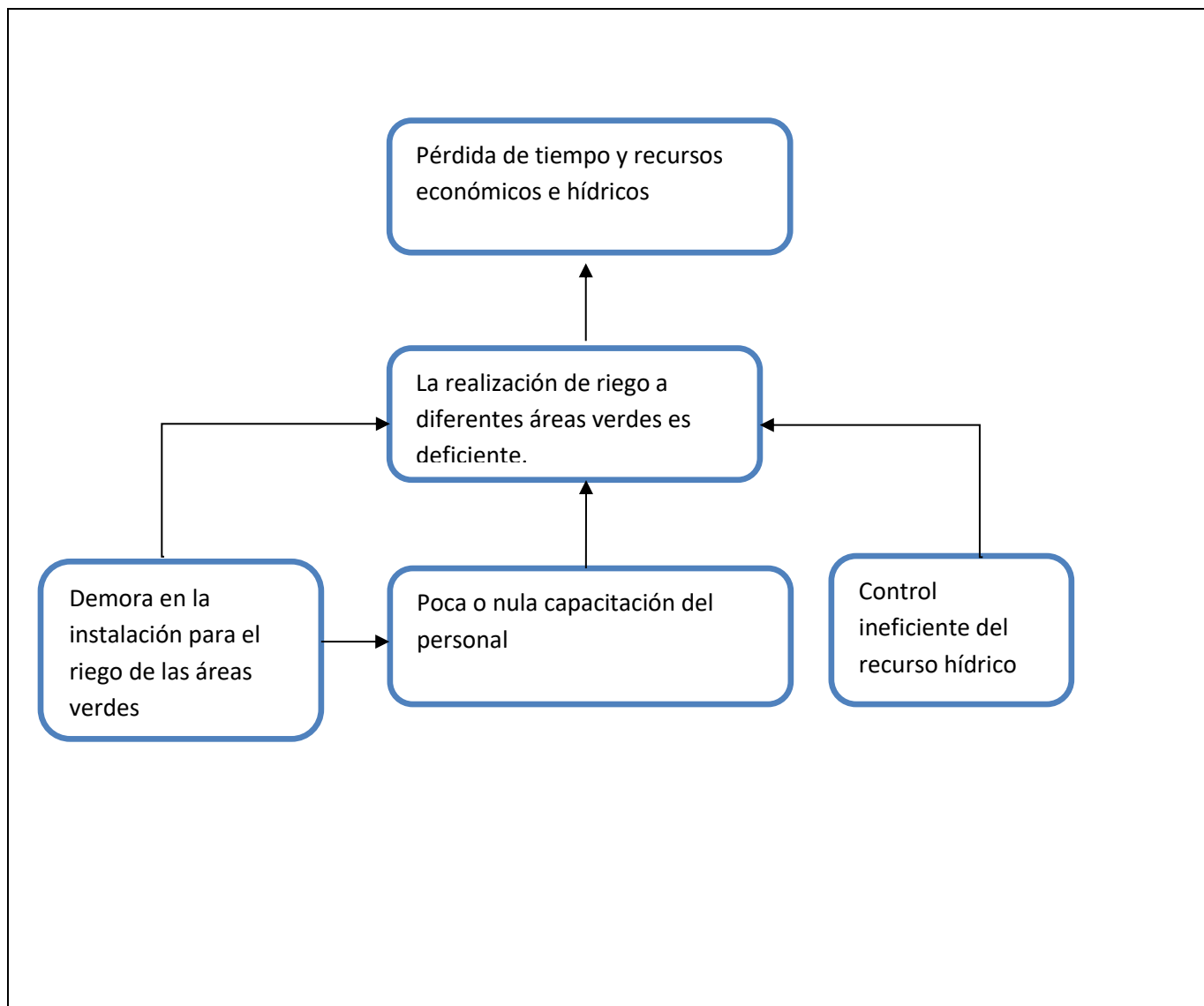


FIGURA 1: ARBOL DE PROBLEMAS

I.2.2.2. ANALISIS DE OBJETIVOS

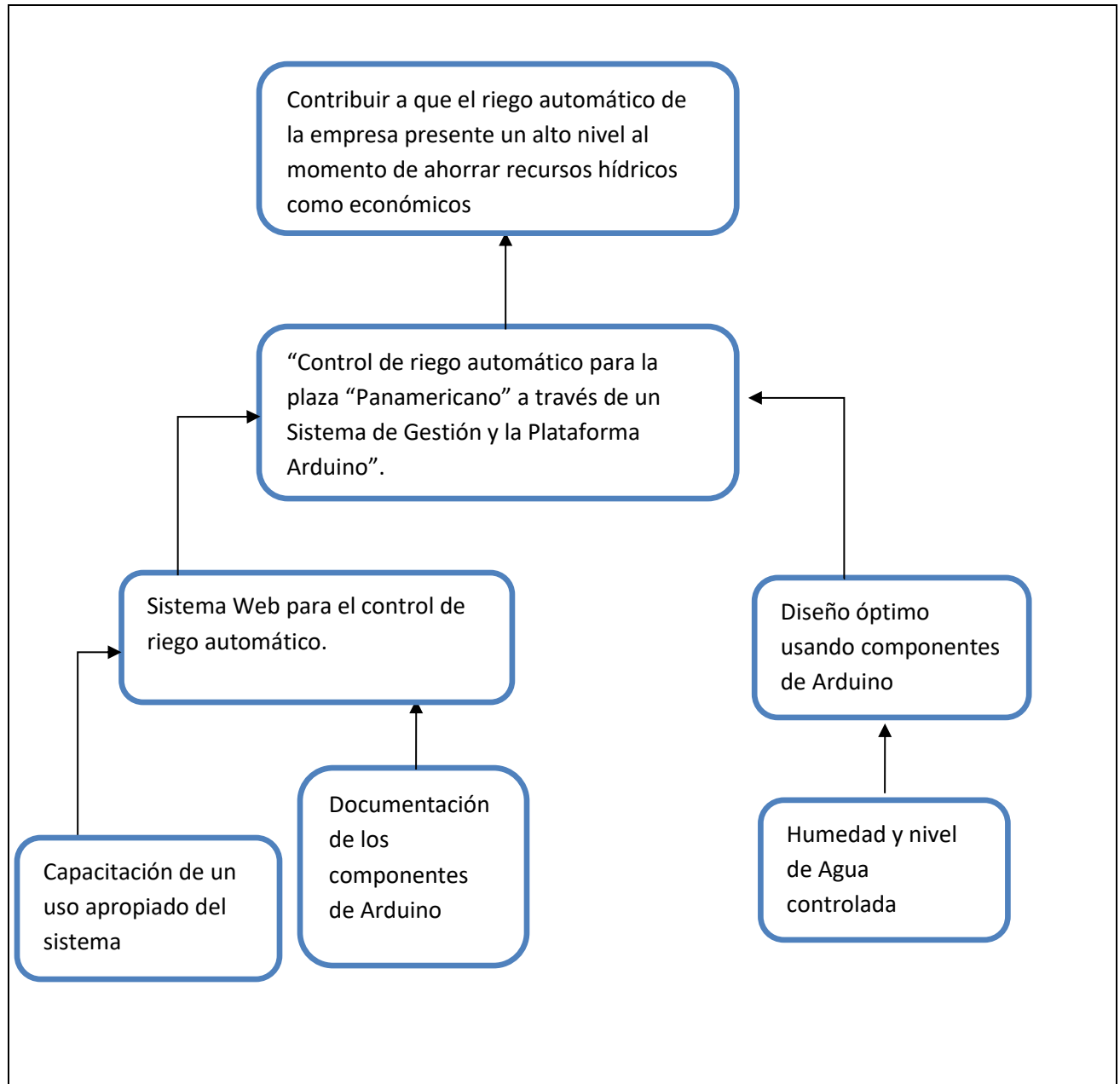


FIGURA 2: ARBOL DE OBJETIVOS

I.2.2.3. SITUACION PLANTEADA CON Y SIN PROYECTO

Situación sin proyecto	Situación con proyecto
➤ Riego manual de todas las áreas verdes durante el día, generan un gasto económico innecesario. ➤ Incomodidad al momento de realizar el riego cuando la persona no se encuentra en el lugar. ➤ Gasto innecesario de recursos hídricos. ➤ No existe un control óptimo del consumo de agua. ➤ La sociedad tiene costumbres muy arraigadas y se resiste al uso de tecnologías electrónicas. ➤ La sociedad no está capacitada en el uso de tecnologías electrónicas. ➤ No existe motivación en el uso de tecnologías electrónicas que permitan el control del riego automático en las plazas. ➤ Ineficiencia, incomodidad y gasto económico en el control del riego de áreas verdes.	➤ Control del encendido de riego el día, de forma automática o manual. ➤ Diseño del sistema de control de riego intuitivo para el usuario. ➤ Control del apagado de riego en los sectores desocupados. ➤ Mayor comodidad del usuario en el momento de encender o apagar el riego en áreas verdes. ➤ Personas capacitadas en usar tecnologías electrónicas que permitan el control del riego en las distintas áreas verdes. ➤ Interés en usar tecnologías electrónicas que permitan el control del riego automático en las áreas verdes. ➤ Control óptimo del riego en las áreas verdes, reduciendo el gasto económico. ➤ Manera ágil de interactuar en el momento de realizar riego.

TABLA 4: SITUACION PLANTEADA

I.2.3. OBJETIVOS

I.2.3.1. OBJETIVO GENERAL

Controlar el riego a distintas áreas verdes de la Plaza “Panamericano” a través de un Sistema Web de control y monitorización

I.2.3.2 OBJETIVOS ESPECIFICOS

- ✓ Simular la operación del sistema de software y hardware para el control de riego automático con el uso de una maqueta.
- ✓ Disminuir el uso de agua potable.
- ✓ Realizar la capacitación a los usuarios que van a usar el sistema.
- ✓ Los rendimientos deben ser más altos al menor costo posible.
- ✓ Plantear una estrategia de socialización del sistema informático desarrollado.

I.2.4. ALCANCES

- ✓ El sistema web permite el registro de nuevos usuarios.
- ✓ El sistema web permitirá tener el control del riego de la plaza por áreas.
- ✓ El sistema web controlará que las personas encargadas de riego solo cumplan con esa función dentro del sistema.
- ✓ El sistema web será manipulable desde cualquier parte con internet.

I.2.5. LIMITACIONES

- ✓ El sistema no registrara plazas solo es para la plaza “Panamericano”.
- ✓ El sistema web no realizara el aviso de cualquier componente en falla.
- ✓ El sistema no realizara informes de riego sobre que usuario lo manipulo.
- ✓ El sistema no contemplara reportes.

I.2.6. MARCO LOGICO DEL PROYECTO

Marco Lógico del Proyecto			
Resumen Narrativo del Proyecto	Indicadores	Medios de Verificación	Supuestos
Fin Contribuir a Mejorar la gestión de riego automático en las distintas áreas verdes de Ornato Público.	A partir del 2018, la satisfacción de la ciudadanía con referencia al riego de áreas verdes, se ha incrementado con referencia al año base.	Se brindara informe de satisfacción del sistema implementado y una proyección de cómo será en un futuro de un Ingeniero experto en el tema	Las áreas verdes utilizan el sistema que le proporciona el proyecto.
Objetivo General (Propósito) "Control de riego automático para la plaza "Panamericano" a través de un Sistema de Gestión y la Plataforma Arduino".	Al finalizar el proyecto, un 65% de los procesos del sistema de riego se encuentra automatizada con respecto a la gestión anterior según el consumo de agua establecido en el pozo.	Carta del Director que expresa su satisfacción del proyecto "Control de riego automático para la Plaza Panamericano de la organización Ornato Público mediante un Sistema Web de control y monitorización".	Avance de proyecto en orden según cronograma Se espera la aceptación de nuevas tecnologías por parte del personal y la aceptación de los cursos.
Objetivos Específicos (Componentes) C1.- Desarrollar y simular la operación del sistema web y sistema de riego automático para el control de riego a áreas verdes.	Del 4 de Abril al 31 de Octubre se desarrolló un 80% del sistema utilizando la metodología de desarrollo ágil SCRUM y de acuerdo a la norma	1.1 Carta de conformidad de la funcionalidad del prototipo firmada por el	Participación activa de la sociedad interesada en adquirir el sistema de

C2.- Realizar la capacitación a los usuarios que van a usar el sistema.	IEEE830 de determinación de requerimientos.	encargado de su administración.	riego automático a diferentes áreas verdes. • Predisposición de la sociedad interesada en la capacitación.
Actividades C1.- Desarrollar y simular la operación del sistema web y sistema de riego automático para el control de riego a áreas verdes. • Tecnología inteligente que garantiza la calidad del producto C2.- Capacitación del sistema 1. Elaboración de los manuales del Instalación y los Manejo del sistema.	Resumen presupuesto SERVICIOS TELEFONICOS 320,00 Bs SERVICIOS DE IMP. Y FOTOGRAF. 532,50 Bs MATERIAL DE ESCRITORIO 176,38 Bs EQUIPAMIENTO 2736,50 Bs SERVICIOS PROFESIONALES 800,00 Bs TRANSPORTE VARIOS 250,00 Bs OTROS 185,00 Bs TOTAL, GASTOS..... 5.000,38 Bs	1.1 Resumen de los costos de las piezas y costo operacional	• Disponibilidad de equipos de computación. • Disponibilidad de personal

I.2.7. METODOLOGIA DE TRABAJO

Las siguientes metodologías serán utilizadas para el desarrollo del sistema, siguiendo cada una de sus fases de acuerdo a los requerimientos del sistema a desarrollar:

Metodología SCRUM para el desarrollo del software:

Una ventaja de esta metodología es que en cada iteración se tiene un producto terminado, siendo por este motivo una de las metodologías más ágiles para alcanzar un grado de certificación en el desarrollo del software.

Metodología UML (Lenguaje unificado de modelado) para el análisis y diseño del software:

El lenguaje unificado de modelado (UML) es el lenguaje de [modelado](#) de sistemas de [software](#) más conocido y utilizado en la actualidad; está respaldado por el [OMG](#) (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

I.2.7.1.DESCRIPCION Y RELACION DE LAS ESTRATEGIAS CON LOS OBJETIVOS

Estrategias	Objetivos Específicos
Se seguirán las fases de SCRUM; eso implica elaborar los distintos Sprint, un cronograma de actividades, realizar la especificación de requerimientos y el modelado del negocio. Posteriormente se procede a diseñar la base de datos, realizar los diagramas en UML y crear los prototipos de interfaces de usuarios. Posteriormente se realiza la programación y documentación del código fuente, así como la elaboración de las pruebas del sistema y la aplicación de las mismas.	Desarrollar un sistema web de control y monitorización para el control de riego automático de la plaza.

TABLA 6: RELACION DE ESTRATEGIAS CON OBJETIVOS

I.2.7.2. CRONOGRAMA DE ACTIVIDADES

Nº	Actividad	Nº días	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
1	Sistema Informático	202										
	Determinación de requerimientos	20				30	19					
	Desarrollo del análisis y diseño	42					20		2			
	Desarrollo del hardware	100						27	2			
	Desarrollo de la programación	100							3			2
	Diseño y documentación del manual del usuario.	15								2-17		
	Realización de pruebas del software	30									2-22	
	Capacitación	33										3-30

TABLA 7: CRONOGRAMA DE ACTIVIDADES

I.2.8. RESULTADOS ESPERADOS

Con el desarrollo del proyecto se pretende contribuir a tener un mejor manejo y control de lo que es el riego de la plaza “Panamericano” y lo que es un ahorro tanto hídrico como económico de la empresa “Ornato Público”, también de aportar a la población en general. Aplicando el componente del sistema informático se busca reducir los tiempos al momento de realizar el riego, realizar un mejor control de los aspersores y contar con la información completa y un historial de datos sobre los sensores.

Con la estrategia de socialización se pretende que los beneficiarios sean capaces de manejar el sistema informático y conocer del beneficio que aporta al desempeño de sus trabajos.

I.2.9. TRANSFERENCIA DE RESULTADOS

I.2.9.1. MEDIOS Y ESTRATEGIAS PARA LA TRANSFERENCIA DE RESULTADOS

La transferencia de resultados se realizará mediante un convenio establecido entre la Universidad y los directos beneficiarios; en este convenio también se definirán los medios que se utilizarán.

I.2.9.2. GRUPO DE BENEFICIARIOS DE LOS RESULTADOS

Los beneficiarios con el desarrollo del proyecto son la empresa y la población en general.

I.3. PRESUPUESTO

ITEM	RUBROS	Aporte Universidad	Otro Aporte	TOTAL (Bs.)
	II.1.1 Sub total rubro			
20000	SERVICIOS NO PERSONALES			
	21000. Servicios Básicos			350
	22000. Servicios de transporte			100
	23000. Alquileres			350
	24000. Mantenimiento y reparación			
	25000. Servicios Profesionales y Comerciales			16000
	II.1.2 Sub total rubro			16800
30000	MATERIALES Y SUMINISTROS			
	31000. Alimentos y Productos Forestales			60
	32000. Productos de Papel, Cartón e Impresos			180
	33000. Textiles y Vestuario.			

	34000. Productos Químicos, Combustibles y Lubricantes			
	39000. Productos Varios.			
	II.1.3 Sub total rubro			240
40000	ACTIVOS REALES			
	43000. Maquinaria y Equipo.			
	46000. Descripción de estudios y proyectos para inversión			
	49000. Otros Activos			
	II.1.4 Sub total rubro			0
	II.1.5 TOTAL			17040
	TOTAL + 40% Incentivo			23861.6

TABLA 8: RUBROS

1) GRUPO 10000. SERVICIOS PERSONALES

a) SUB GRUPO 12000. Empleados no Permanentes

Partida	Personal	Remuneración	Tiempo/meses	Total
12100	Personal Eventual			
Total				

TABLA 9: SERVICIOS PERSONALES

2) GRUPO 20000. SERVICIOS NO PERSONALES

b) SUB GRUPO 21000. Descripción de los gastos de servicios básicos

Partida	Tipo de servicio básico *	Costo	Tiempo mes	Costo Total
21100	Comunicación			
21200	Energía Eléctrica	160	8	200
21300	Agua			
21400	Servicios Telefónicos	100	8	800
Total				1000

TABLA 10: DESCRIPCION DE LOS GASTOS DE SERVICIOS BASICOS

c) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal

Partida	Personal	Lugar	Nº de viajes	Costo unitario*	Costo total
22100	Pasajes	Plaza Panamericano y Ornato publico	50	2	100
22200	Viáticos				
22300	Fletes y Almacenamientos				
22600	Transporte de Personal				
Total					100
Total sub grupo 22000					100

TABLA 11: DESCRIPCION DE LOS GASTOS DE VIAJES Y TRANSPORTE DE PERSONAL

d) SUB GRUPO 23000. Descripción de los gastos por concepto de alquileres de equipos y maquinarias

Partida	Alquiler de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
23100	Alquiler de Edificios			
23200	Alquiler de Equipos y Maquinaria	350	1	350
23300	Alquiler de Tierras y Terrenos			
Total				350

TABLA 12: DESCRIPCION DE LOS GASTOS POR CONCEPTO DE ALQUILERES DE EQUIPOS Y MAQUINARIAS

e) SUB GRUPO 24000. Descripción mantenimiento y reparación

Partida	Mantenimiento y reparación de equipo y maquinaria	Costo unitario	Tiempo mes	Costo total
24100	Mantenimiento y Reparación de Lugar y Equipos			
24300	Otros Gastos por Mantenimiento y Reparación			
Total				

TABLA 13: DESCRIPCION MANTENIMIENTO Y REPARACION

f) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos

Partida	Tipo de material *	Cantidad	Costo/Unitario	Total
32100	Papel de Escritorio	1000	0.18	180

32200	Productos de Artes Gráficas, Papel y Cartón			
32300	Libros y Revistas			
32400	Textos de Enseñanza			
32500	Periódicos			
Total				180

TABLA 14: DESCRIPCION DE LOS GASTOS PRODUCTOS DE PAPEL, CARTON E IMPRESOS

g) SUB GRUPO 34000. Combustibles, Productos Químicos, Farmacéuticos y Otros

Partida	Combustibles, Productos Químicos, Farmacéuticos y Otros	Cantidad	Costo/Unitario	Total
34110	Combustibles y Lubricantes para Consumo			
34200	Productos químicos y Farmacéuticos			
34400	Productos de Cuero y Caucho			
34500	Productos de Minerales no Metálicos y Plásticos			
34600	Productos Metálicos			
34700	Minerales			
34800	Herramientas Menores			
Total				

TABLA 15: COMBUSTIBLES, PRODUCTOS QUIMICOS, FARMACEUTICOS Y OTROS

II.1. COMPONENTE 1: “SISTEMA WEB”

II.1.1. PLAN DE DESARROLLO DEL SOFTWARE

II.1.1.1. INTRODUCCION

Este plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto de prácticas de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Universidad Autónoma Juan Misael Saracho. Este documento provee una visión global del enfoque de desarrollo propuesto.

En esta parte del documento se especifica de la forma más detallada posible las características del sistema web de control y monitorización de Ornato Público para la plaza “Panamericano”.

Para el desarrollo del sistema se utilizó la metodología SCRUM y el lenguaje de modelado UML; en este documento también se definen los alcances y limitaciones del sistema propuesto, haciendo énfasis en que este es un sistema a medida para la plaza.

II.1.1.2. PROPOSITO

El propósito del Plan de Desarrollo de Software es proporcionar la información y descripción de las necesidades en cuanto a funciones, restricciones y limitaciones necesarias para controlar el proyecto. En él se describe el enfoque de desarrollo del software y hardware.

El jefe del proyecto lo utiliza para organizar la agenda, necesidades de recursos y realizar su seguimiento.

Posteriormente lo usara para entender lo que debe hacer, cuándo debe hacerlo y que otras actividades dependen de ello.

II.1.1.3. ALCANCE DEL DOCUMENTO

En este documento se establecen el propósito, los alcances, limitaciones, un marco tecnológico, un marco Teórico, la ingeniería de Requerimientos.

En resumen, una descripción global del sistema, su arquitectura, su topología, las características, el modelo del negocio, los casos de uso del sistema y la base de datos.

II.1.1.4. OBJETIVOS

II.1.1.4.1. OBJETIVO GENERAL

Desarrollar un Sistema Web del Control y monitorización utilizando tecnologías adecuadas para el desarrollo del Software y hardware para el control de los procesos del proyecto Riego Automático para la plaza “Panamericano”.

II.1.1.4.2. OBJETIVOS ESPECIFICOS

- ✓ Aplicar la plataforma Arduino al sistema web.
- ✓ Analizar toda la información que se recopile del proyecto Riego Automático.
- ✓ Determinación de requerimientos con la norma IEEE830.
- ✓ Diseñar una base de datos que nos permita almacenar toda la información del proyecto Riego Automático.
- ✓ Analizar y diseñar diagramas bajo el modelado UML (Lenguaje Unificado de Modelado).

II.1.1.5. ENTREGABLES DEL PROYECTO

Plan de Desarrollo del Software

Es el presente documento.

Modelo de Casos de Uso del Negocio

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (agentes de registro, solicitantes finales, otros sistemas etc.); permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

Modelo de Objetos del Negocio

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio.

Modelo de Casos de Uso

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

Especificación de los Casos de Uso

Para los casos de usos que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluye: pre-condiciones, post-condiciones y flujo de eventos.

Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

Modelo de Datos

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases.

Diseño de Pantallas de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros serán

desechados. Asimismo, este artefacto será desechado en la fase de Construcción en la medida que los resultados de las iteraciones vayan desarrollando el producto final.

Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

Material de Apoyo al Usuario Final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Manual de Usuario.

Producto y Prototipo

Los ficheros del producto empaquetados y almacenados en un CD con los mecanismos apropiados para facilitar su instalación. El producto, a partir del primer sprint de la fase es desarrollado incremental e iterativamente, obteniéndose una nueva versión al final de cada sprint también será entregado el primer prototipo que es la maqueta.

II.1.1.6. ALCANCE Y LIMITACIONES DEL COMPONENTE

De forma general el sistema será capaz de trabajar sobre las siguientes funciones:

- ✓ **Gestión de usuarios:** permitirá que se registre información de los usuarios del sistema y se lleve un control de las acciones que se ejecutan dentro del mismo.
- ✓ **Gestión Riego:** permitirá el control de riego a distintas áreas verdes de la plaza, también contará con una información sobre datos de los distintos sensores.

Sin embargo, este sistema no tendrá un modelo de reportes por usuario que utilice el sistema con el hardware.

II.1.2. MARCO METODOLOGICO

II.1.2.1. MODELO DE PROGRAMACION

Se utilizara un programación orientada a objetos para el sistema web y para el arduino no orientada a objetos, el modelo orientado a objetos usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en

varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990.

II.1.2.2. LENGUAJES DE PROGRAMACION

Se utilizará los siguientes lenguajes de programación:

- ✓ JavaScript-JQuery: **jQuery** es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. Esta librería será utilizada en la vista del proyecto.
- ✓ Java: Lenguaje orientado a objetos que será usado en los controladores del proyecto.



- ✓ C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.



II.1.3. MARCO TEORICO

II.1.3.1. MODELO VISTA CONTROLADOR

II.1.3.1.1. INTRODUCCION

Modelo Vista Controlador (MVC) es un patrón o modelo de abstracción de desarrollo de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. El patrón de llamada y retorno MVC (según CMU), se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

II.1.3.1.2. DESCRIPCION DEL PATRON

- ✓ **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.
- ✓ **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- ✓ **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del **MVC** corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre **Vista** y su correspondiente **Controlador** de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y

mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de **MVC**, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, se podría utilizar el patrón Observador para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. En general el controlador no pasa objetos de dominio (el modelo) a la vista, aunque puede dar la orden a la vista para que se actualice.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

II.1.4. OPCIONES DE LAS TECNOLOGIAS A USAR EN EL RIEGO AUTOMATIZADO

Programador de Tiempos



Diseñado para residenciales, el **RPS 46** tiene 4 programas independientes para permitir un **riego eficiente** en programas diferentes.

Una característica clave de este **programador de riego** es el ajuste estacional que permite ajustar los tiempos de **riego** a medida que cambian las estaciones del año.

Características:

- *Cuatro programas* independientes con 4 arranques cada uno (máximo 16 arranques por día).
- *Tiempo de riego* por estación de 1 a 12 horas y 59 minutos .
- *Días de riego individuales*, intervalo de hasta 15 días o pares/impares.
- *Caja para interior* con transformador externo enchufable.
- Disponible en 4 y 6 sectores

Costo:

- 3000 bs

Controlador lógico Pogramable (PLC)



Características:

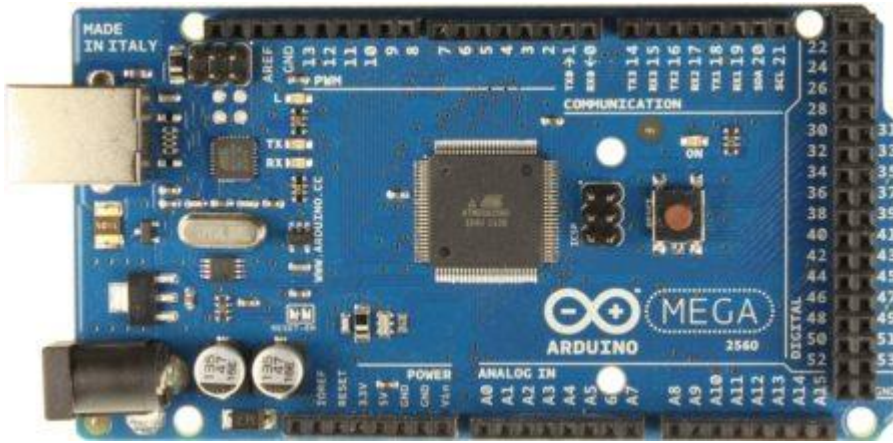
- Alta capacidad de procesamiento. Cálculo de 64 bits.
- Interfaz Ethernet / PROFINET integrado.
- Entradas analógicas integradas.
- Bloques de función para control de ejes conforme a PLCopen.
- Programación mediante la herramienta de software STEP 7 Basic v13 para la configuración y programación no sólo del S7-1200, sino de manera integrada los paneles de la gama Simatic Basic Panels.

El sistema S7-1200 desarrollado viene equipado con cinco modelos diferentes de CPU (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C y CPU 1217C) que se podrán expandir a las necesidades y requerimientos de las máquinas.

Costo:

- 5000 \$us

Arduino Microcontrolador: ATmega2560



Características:

- **Microcontrolador:** ATmega2560
- **Voltaje Operativo:** 5V
- **Voltaje de Entrada:** 7-12V
- **Voltaje de Entrada(límites):** 6-20V
- **Pines digitales de Entrada/Salida:** 54 (de los cuales 15 proveen salida PWM)
- **Pines análogos de entrada:** 16
- **Corriente DC por cada Pin Entrada/Salida:** 40 mA
- **Corriente DC entregada en el Pin 3.3V:** 50 mA
- **Memoria Flash:** 256 KB (8KB usados por el bootloader)
- **SRAM:** 8KB
- **EEPROM:** 4KB
- **Clock Speed:** 16 MHz

Costo:

- 200 bs

Analisis de las distintas Tecnologias

Se eligió Arduino ya que es de software libre como hardware y por qué es más económico y cumple mayores funcionalidades que las demás tecnologías además puede adaptarse a mayores componentes para proyectos funcionales.

II.1.4.1.1. HERRAMIENTAS UTILIZADAS EN EL PROYECTO MOTOR DE BASE DE DATOS

PostgreSQL

Es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Algunas de sus principales características son: Llaves ajenas o llaves foráneas (foreign keys), disparadores (triggers), vistas, integridad transaccional, acceso concurrente multiversión (no se bloquean las tablas, ni las filas, cuando un proceso escribe), capacidad de albergar programas en el servidor en varios lenguajes, herencia de tablas, tipos de datos y operaciones geométricas. En cuanto a rendimiento se encuentra a la altura de ORACLE.



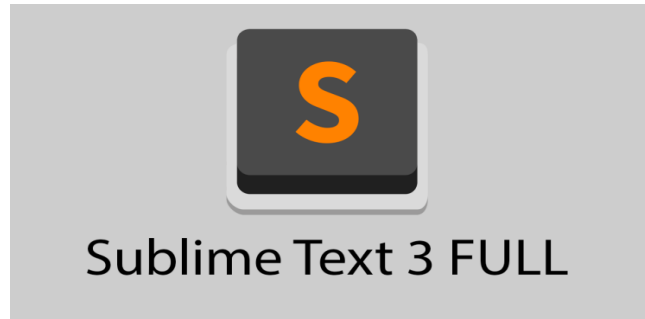
SOFTWARE PARA LA PROGRAMACION Y DISEÑO DEL SISTEMA

Eclipse

Es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto

Sublime Text 3

Sublime Text es un editor de código multiplataforma, rápido que brinda muchas las facilidades de: tabulaciones, filtraciones, autocompletado de código, permite programar el Back-end y el Front-end del sistema en desarrollo. Además, te brinda interfaces que te permiten tener una mejor ubicación y reconocimiento de código.



HERRAMIENTAS SOFTWARE PARA EL PROCESO DE MODELADO

A continuación, se enuncia el software utilizado para el proceso de modelado, creación del diagrama de clases, casos de uso, base de datos, etc.

Navicat Data Modeler:

Navicat Data Modeler es una herramienta que ayuda a construir modelos de datos conceptuales, de manera rápida y simplificando la tarea de crear modelos entidad-relación.



Enterprise Architect

Es una herramienta Computer Aided Software Engineering (CASE), para diseñar y construir sistemas software, para el proceso de modelado. Está basado en la especificación de UML 2.1.



II.1.4.2. METODOLOGIA DE DESARROLLO

II.1.4.2.1. METODOLOGÍA SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Las actividades que se llevan a cabo en Scrum son las siguientes

II.1.4.2.2. PLANIFICACIÓN DE LA ITERACIÓN

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

- **Selección de requisitos.** El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
- **Planificación de la iteración.** El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.

II.1.4.2.3. EJECUCIÓN DE LA ITERACIÓN

Cada día el equipo realiza una reunión de sincronización. Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ✓ ¿Qué he hecho desde la última reunión de sincronización?
- ✓ ¿Qué voy a hacer a partir de este momento?
- ✓ ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador (Scrum Master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

- ✓ Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- ✓ Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

Durante la iteración, el cliente junto con el equipo refina la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o re planifican los objetivos del proyecto para maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- **Demostración.** El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, re planificando el proyecto.
- **Retrospectiva.** El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

II.1.4.2.4. CÓMO FUNCIONA SCRUM

Actividades

- Planificación de la iteración (Sprint Planning)
- Ejecución de la iteración (Sprint)
- Reunión diaria de sincronización del equipo (Scrum Daily Meeting)
- Demostración de los requisitos completados (Sprint Review)
- Retrospectiva (Sprint Retrospective)
- Refinamiento de la lista de requisitos y cambios en el proyecto

Responsabilidades

- Cliente (ProductOwner)
- Facilitador (Scrum Master)
- Equipo (Team)

Herramientas:

- Lista de requisitos priorizada (ProductBacklog)
- Lista de tareas de la iteración (Sprint Backlog)
- Gráficos de trabajo pendiente (Burndown Chart)

Marco aplicativo

Para llevar a cabo cada iteración en SCRUM se define la lista de actividades priorizada de todas las actividades a realizar durante el desarrollo del proyecto, una vez obtenida esta lista, se empieza con la planeación de cada sprint para completar el proyecto final y tener un producto terminado.

Lista priorizada del proyecto

En el siguiente cuadro se muestra la lista priorizada para el proyecto, donde las actividades más importantes van al comienzo.

Nro.	Nombre de la actividad a realizar
1	Análisis de requisitos
2	Identificación de tecnologías
3	Diseño de modelo de la base de datos
4	Implementación de la base de datos
5	Diseño de la interfaz del Sistema web
6	Comunicar Arduino con los diferentes sensores
7	Integración del servidor

Sprint para el proyecto

Obtenida esta lista priorizada se da comienzo a realizar cada sprint, para el primer Sprint realizado se tiene la siguiente lista de actividades.

Nro.	Actividad	Tareas
1	Análisis de requisitos	- Definir la lista de todas las funcionalidades que el sistema realizará - Priorizar requisitos más importantes
2	Identificación de tecnologías	- Identificar tecnologías para el Backend (Servidor) - Identificar tecnologías para el Frontend (Cliente) - Seleccionar tecnología para la base de datos - Seleccionar entornos de desarrollo - Seleccionar lenguajes de programación

3	Diseño de la interfaz del Sistema web	- Definir el diseño de las pantallas del sistema - Plasmar este diseño utilizando tecnologías web como ser HTML, Css, JavaScript y JQuery.
4	Diseño de modelo de la base de datos	- Definir las tablas a utilizar en el Sistema

Una vez concluido este primer Sprint se empieza con la ejecución del segundo Sprint, en el cual se tiene las siguientes actividades a realizar

Nro.	Actividad	Tareas
5	Diseño de la interfaz de Arduino	- Definimos los componentes q utilizaremos en el prototipo y su posterior conexión - Definir el modelo de software de Arduino
6	Comunicar Arduino con los diferentes sensores	- Crear el Sketch (código para la placa Arduino en lenguaje C++) - Probar la implementación con el Arduino y los sensores - Probar la conexión con varios sensores conectados - Implementar la conexión de Arduino y sus sensores en la maqueta
7	Integración del servidor	- Comunicar la placa Arduino con el servidor mediante Java utilizando la librería PanaHitek

II.1.4.2.5. UML- LENGUAJE DE MODELADO UNIFICADO

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group) gracias a Jesús Moreno.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

II.1.4.2.5.1. OBJETIVOS DE UML

UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.

UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

II.1.4.2.5.2. DIAGRAMAS DE UML

En UML existen dos tipos de diagramas; los diagramas que nos dan una vista estática y dinámica.

II.1.4.2.5.2.1. DIAGRAMAS ESTATICOS

Estos diagramas nos describen el comportamiento del sistema en el tiempo

Diagramas de Casos de Uso: Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

Diagrama de Clases: El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clases incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

II.1.4.2.5.2.2. DIAGRAMAS DINAMICOS

Diagramas de Secuencia: Diagrama que muestra las interacciones entre los objetos organizadas en una secuencia temporal. En participar muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados.

Representa una interacción, un conjunto de comunicaciones entre objetos organizadas visualmente por orden temporal. A diferencia de los diagramas de colaboración, los diagramas de secuencia incluyen secuencia temporal, pero no incluyen las relaciones

entre objetos. Pueden existir de forma de descriptor (describiendo todos los posibles escenarios) y en forma de instancia (describiendo un escenario real).

Diagramas de Actividades: Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad. Un diagrama de actividad es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

II.1.5. ANALISIS DE REQUERIMIENTOS

II.1.5.1. INTRODUCCION

La etapa de Análisis de Requerimientos, es la primera etapa en el desarrollo de un sistema Informático. En esta etapa, debemos de responder a una pregunta fundamental: ¿Qué es lo que quiere el cliente? Y para ello, debemos diagnosticar la situación actual, recopilar los requerimientos del cliente, tanto en relación al sistema, como generales respecto del área informática, es decir la situación ideal, para así poder definir alternativas de solución, según las cuales podremos avanzar desde lo que hoy se posee, hacia el punto que se pretende llegar.

Como parte de nuestro trabajo, deberemos señalar cuál de las alternativas es la más conveniente y justificarlo en la propuesta.

La "Ingeniería de Requerimientos" se utiliza para definir todas las actividades involucradas en el descubrimiento, documentación y mantenimiento de los requerimientos para un producto determinado. El uso del término "ingeniería" implica que se deben utilizar técnicas sistemáticas y repetibles para asegurar que los requerimientos del sistema estén completos y sean consistentes y relevantes.

Los requerimientos son la pieza fundamental en un proyecto de desarrollo de software, en ellos se basan muchos participantes del proyecto para:

Planear el proyecto y los recursos que se usarán en él. Se utilizarán los requerimientos como una base para la estimación del esfuerzo necesario en un proyecto.

Especificar el tipo de verificaciones que se habrán de realizar al sistema. Por ejemplo: cuando se está tratando de alinear a cierta norma oficial o estándar.

Planear la estrategia de prueba a la que habrá de ser sometido el sistema. Los requerimientos son la base sobre la cual se decide si un caso de prueba fue ejecutado exitosamente por el sistema o no.

Son el fundamento del ciclo de vida del proyecto. Los requerimientos documentados son la base para crear la documentación del sistema. De ahí su importancia y la importancia de que deban de ser definidos y manejados de la forma más adecuada posible.

II.1.6. MODELADO DE LOS CASOS DE USO DEL NEGOCIO

II.1.6.1. INTRODUCCION

El Modelo de Casos de Uso del Negocio. En un artefacto de la disciplina requisitos en la metodología Scrum la cual se está implementando.

II.1.6.2. PROPOSITO

- ✓ Comprender la estructura y la dinámica del Riego en la plaza de Ornato público.
- ✓ Comprender los problemas actuales e identificar posibles mejoras.
- ✓ Comprender los procesos del negocio de la Empresa Ornato Público.

II.1.6.3. ALCANCE

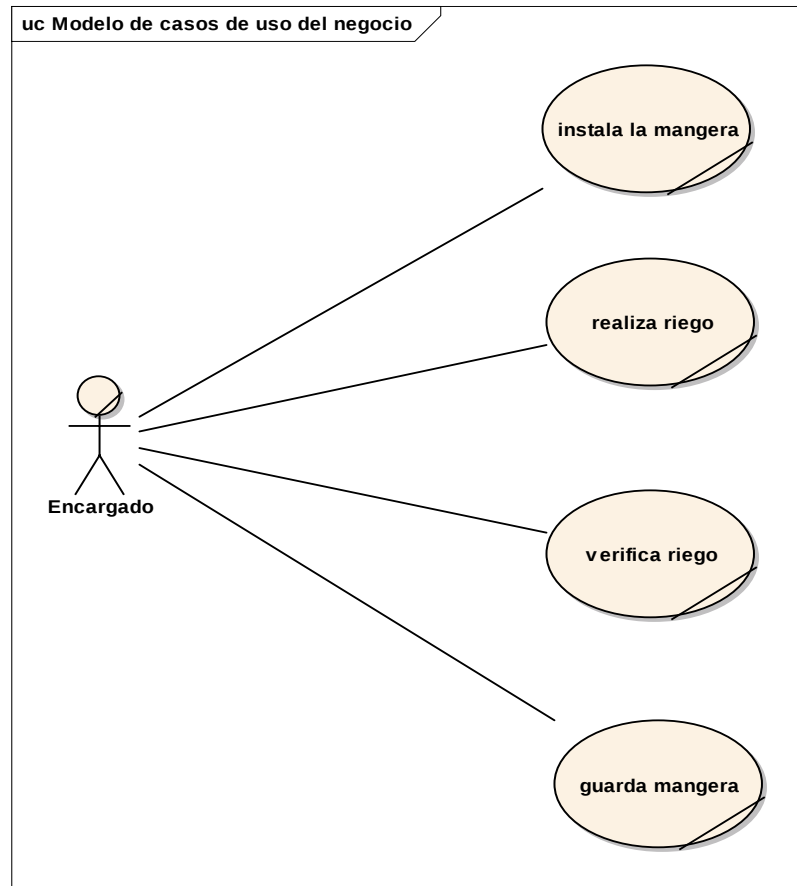
- ✓ Describir los procesos de negocio y los usuarios
- ✓ Identificar y definir los procesos del negocio según los objetivos de la organización.
- ✓ Definir un caso de uso del negocio para cada proceso del negocio (diagrama de casos de uso del negocio puede mostrar el contexto y los límites del establecimiento)

II.1.6.4. DESCRIPCION DE LOS ACTORES DEL NEGOCIO

Se identificaron los siguientes actores los cuales interactúan con los procesos principales que se llevan a cabo en Ornato Público.

- ✓ **Administrador:** Administra toda el área de bienes raíces, pasa reportes diarios al gerente sobre el estado de los inmuebles y se encarga de supervisar a los corredores de bienes raíces.
- ✓ **Encargado:** Es el encargado de controlar el riego con el sistema propuesto para la organización.

II.1.6.5. DIAGRAMA DE CASOS DE USO DEL NEGOCIO



II.1.7. MODELO DE CASOS DE USO

II.1.7.1. INTRODUCCION

Los casos de uso nos sirven para representar la manera de cómo el personal interactúa con el sistema que se desarrollará.

II.1.7.2. PROPOSITO

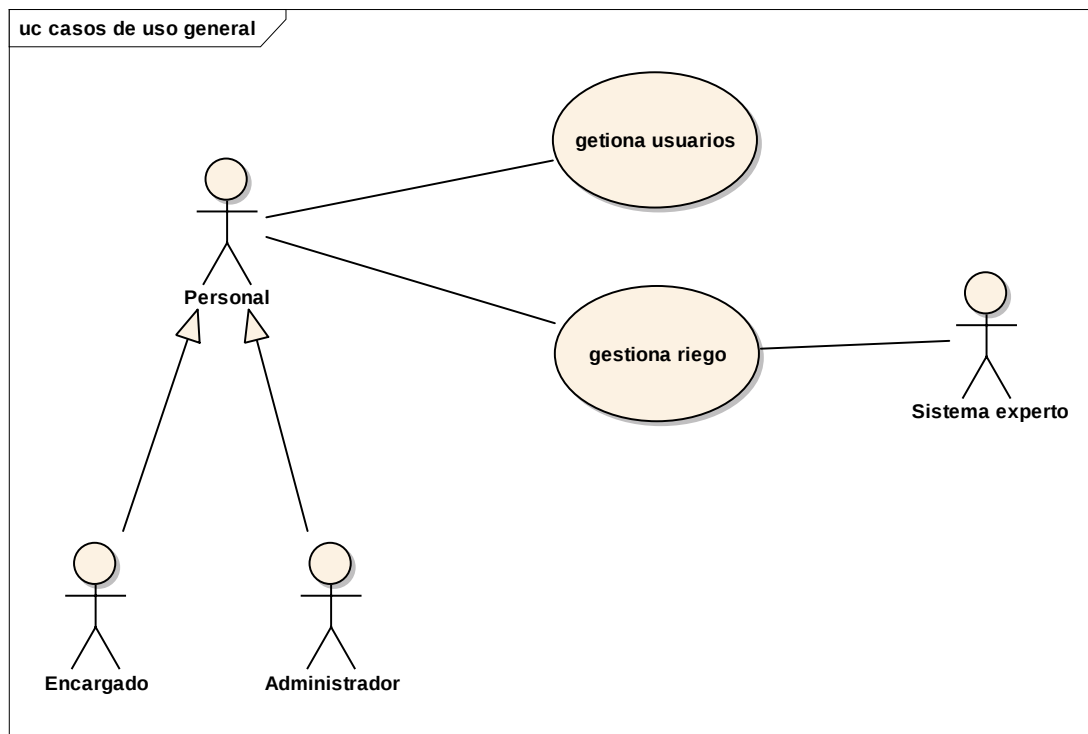
- ✓ Modelar el contexto del sistema

- ✓ Modelar los requerimientos del sistema
- ✓ Identificar los procesos del sistema
- ✓ Estimular a que los usuarios potenciales hablen del sistema desde su propio punto de vista
- ✓ Involucrar a los usuarios en las etapas iniciales del análisis y diseño del sistema
- ✓ Ayudarnos a obtener los requerimientos desde el punto de vista del usuario

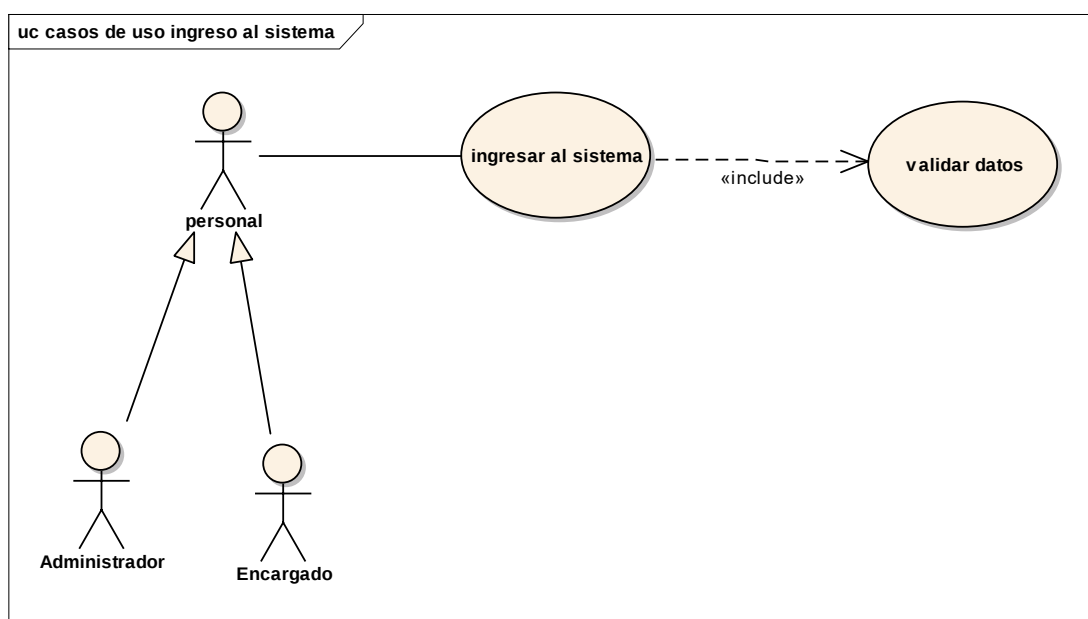
II.1.7.3. ALCANCE

- ✓ Describir lo que el sistema informático realizará dentro de la organización
- ✓ Describir los alcances del sistema
- ✓ Describir los procesos del sistema y del cliente

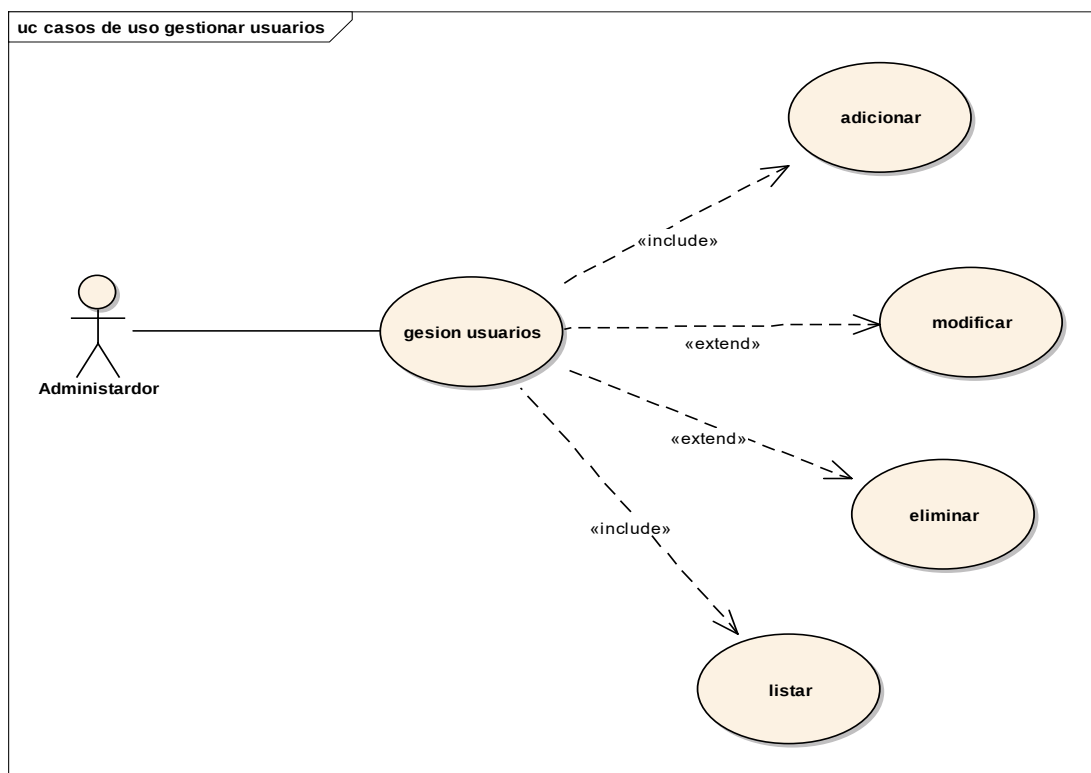
II.1.7.4. DIAGRAMA DE CASOS DE USO DEL SISTEMA



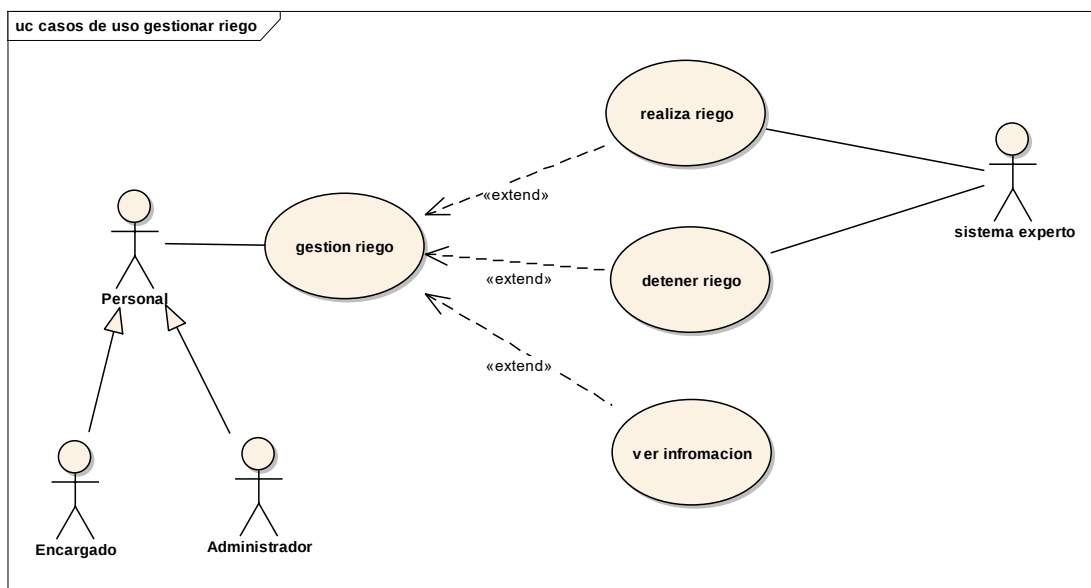
II.1.7.4.1. DIAGRAMA DE CASO DE USO INGRESAR AL SISTEMA



II.1.7.4.2. DIAGRAMA CASO DE USO GESTION USUARIO



II.1.7.4.3. DIAGRAMA DE CASO DE USO GESTION RIEGO



II.1.8. ESPECIFICACION DE CASOS DE USO

II.1.8.1. INTRODUCCION

Las especificaciones de los Casos de Uso es una descripción detallada de los casos de uso identificados en el análisis.

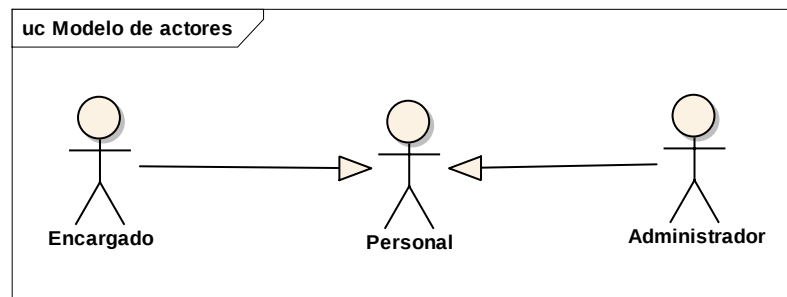
II.1.8.2. PROPOSITO

- ✓ Comprender el funcionamiento de los casos de Uso del Sistema
- ✓ Describir específicamente cada Caso de Uso

II.1.8.3. ALCANCE

- ✓ Describir los procesos internos del sistema en cada Caso de Uso
- ✓ Detallar los flujos de cada caso de uso según lo establecido por la organización.

II.1.8.4. ESPECIFICACION DE LOS ACTORES



II.1.8.4.1. ADMINISTRADOR

Es el encargado del área del sistema tiene el control de todas las funcionalidades que corresponden a esta área; cuenta con los permisos necesarios para operar sobre el área de riego y de los usuarios.

II.1.8.4.2. ENCARGADO

Es el encargado del área riego tiene el control de todas las funcionalidades que corresponden a esta área; sin embargo, no cuenta con los permisos necesarios para operar sobre el área de riego.

II.1.8.5. ESPECIFICACION DE CASOS DE USO

II.1.8.5.1. GESTION DE USUARIOS

II.1.8.5.1.1. VALIDAR USUARIO

Caso de uso	Validar Usuario	
Actores	Usuario	
Tipo	Básico	
Propósito	Verificar si el usuario que intenta ingresar al sistema está registrado o no.	
Descripción	A partir de este caso de uso se podrá ingresar al sistema.	
Precondiciones	- El usuario debe estar registrado - El usuario debe ingresar usuario y la clave - Aceptar ingresar al sistema	
Post Condiciones	Curso normal El sistema muestra la pantalla para ingresar el nombre de usuario y la contraseña que luego de validar nos muestra el contenedor principal del sistema con el menú a un costado.	Alternativas - En caso de que los datos ingresados sean incorrectos, el sistema informará de ello. - Si el usuario no se encuentra registrado no podrá ingresar al sistema.

II.1.8.5.1.2. ADICIONAR USUARIO

Caso de uso	Adicionar usuario
Actores	Administrador
Tipo	Básico
Propósito	Describir como se debe agregar un usuario al sistema.

Descripción	A partir de este caso de uso se podrá registrar un nuevo usuario.	
Precondiciones	- El usuario debe estar registrado	
Post Condiciones	Curso normal Al hacer clic en la opción de Registra usuario en la pantalla de listado de usuarios se redirigirá a una pantalla en la que se llenara los datos de los usuarios, esta pantalla contiene campos obligatorios y al presionar la opción de Guardar se regresa a la pantalla de listar usuarios.	Alternativas - En caso de que los datos ingresados sean incorrectos, el sistema informará de ello. - Si se presionar Cancelar no se guarda la información y se regresa a la pantalla de listar usuarios.

II.1.8.5.1.3. MODIFICAR USUARIO

Caso de uso	Modificar Usuario	
Actores	Administrador	
Tipo	Básico	
Propósito	Permitir que se puedan modificar los datos de un usuario registrado previamente en el sistema.	
Descripción	A partir de este caso de uso se podrá modificar la información de un usuario.	
Precondiciones	- El usuario debe haberse autenticado. - Se debe haber guardado la información de un usuario	
Post Condiciones	Curso normal	Alternativas

	Al hacer clic en la opción de Modificar en la pantalla de listado de usuarios se redirigirá a una pantalla en la que se muestra los campos de información editable, esta pantalla contiene campos obligatorios y al presionar la opción de Guardar se regresa a la pantalla de listar usuarios.	- En caso de que los datos ingresados sean incorrectos, el sistema informará de ello. - Si se presionar Cancelar no se guarda la nueva información y se regresa a la pantalla de listar usuarios manteniendo la información anterior.
--	---	--

II.1.8.5.1.4. ELIMINAR USUARIO

Casos de Uso	Eliminar Usuario	
Actores	Administrador	
Tipo	Básico	
Propósito	Dar de baja a un usuario para que no pueda ingresar al sistema. No se eliminarán sus datos.	
Descripción	A partir de este caso de uso se podrá eliminar la información de un usuario.	
Precondiciones	- El usuario debe haberse autenticado. - Se debe haber guardado la información de un usuario	
Post Condiciones	Curso normal Al hacer clic en la opción de Borrar en la pantalla de listado de usuarios se mostrará una pantalla de tipo pop up en la que se	Alternativas - Si se presionar Cancelar no se produce ningún cambio y no se refresca la información.

	presentará la opción de Aceptar el borrado o Cancelar el borrado, cuando se presionar la opción de Aceptar se actualiza la información de la pantalla listar usuarios.	
--	--	--

II.1.8.5.1.5. LISTAR USUARIO

Caso de uso	Listar usuario	
Actores	Administrador	
Tipo	Básico	
Propósito	Describir el listado de los usuarios registrados en el sistema.	
Descripción	A partir de este caso de uso se podrá listar la información de un usuario.	
Precondiciones	- El usuario debe haberse autenticado. - Se debe haber guardado la información de un usuario	
Post Condiciones	Curso normal Al escoger la opción de listar usuarios en el menú se mostrar la pantalla de listado de usuarios. Esta pantalla muestra el código, nombre completo, el estado, las opciones de borrar, modificar datos, modificar contraseña, ver información y registrar usuario.	Alternativas

II.1.8.5.1.6. ADICIONAR LOGIN Y CLAVE

Caso de uso	Adicionar login y clave
Actores	Adiministrador
Tipo	Extensión.
Propósito	Permite crear login y clave para un usuario del sistema.
Resumen	Este caso de uso hace referencia a la creación de datos de acceso al sistema (login y clave) para los usuarios del sistema.
Precondiciones	El usuario debe haber ingresado al sistema, estar autorizado para gestionar los usuarios, seleccionar un usuario de la lista y haber seleccionado la opción log. y clave en la pantalla gestión usuarios (P-3).

II.1.8.5.1.7. GESTIONAR ROLES

Caso de Uso	Gestionar roles	
Actores	Administrador	
Tipo	Básico	
Propósito	Describir los procesos de roles que realiza el sistema.	
Descripción	A partir de este caso de uso se podrá listar, adicionar, modificar y eliminar la información de un rol.	
Precondiciones	- El usuario debe haberse autenticado.	
Post Condiciones	Curso normal Al escoger la opción de lista de roles en el menú se mostrar la pantalla de listado de roles. Esta pantalla	Alternativas - En caso de seleccionarse la opción Cancelar no se guarda ninguna información o

	muestra el número de rol, código, nombre del rol, el estado, las opciones de borrar, modificar y registrar rol. Si selecciona las opciones de registrar o modificar un rol se re direccionará a una nueva pantalla en la que se debe llenar esta información. Una vez se seleccione guardar o modifica la información se retornará a la pantalla de listado de roles. En el caso de borrar un rol se mostrará una pantalla tipo pop up que actualizará la información de la página listar roles.	modificación y se retornara a la pantalla de listado de roles.
--	---	---

II.1.8.5.1.8. GESTIONAR RIEGO

Caso de Uso	Gestionar riego
Actores	Administrador, Encargado
Tipo	Básico
Propósito	Describir los procesos de riego que realiza el sistema.
Descripción	A partir de este caso de uso se podrá controlar el riego como los distintos aspersores como la bomba de agua.
Precondiciones	- El usuario debe haberse autenticado.

Post Condiciones	Curso normal Al escoger la opción de detener o apagar la bomba controla el encendido y apagado de la bomba ubicada en la plaza. Si selecciona las opciones de encender o apagar los distintos aspersores. Todo esto se lo realizara en caso de emergencia.	Alternativas - En caso de seleccionarse la opción Cancelar no se guarda ninguna información o modificación y se retornara a la pantalla de listado de roles.
---------------------	--	--

II.1.9. DIAGRAMA DE ACTIVIDADES

II.1.9.1. INTRODUCCION

Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación.

II.1.9.2. PROPOSITO

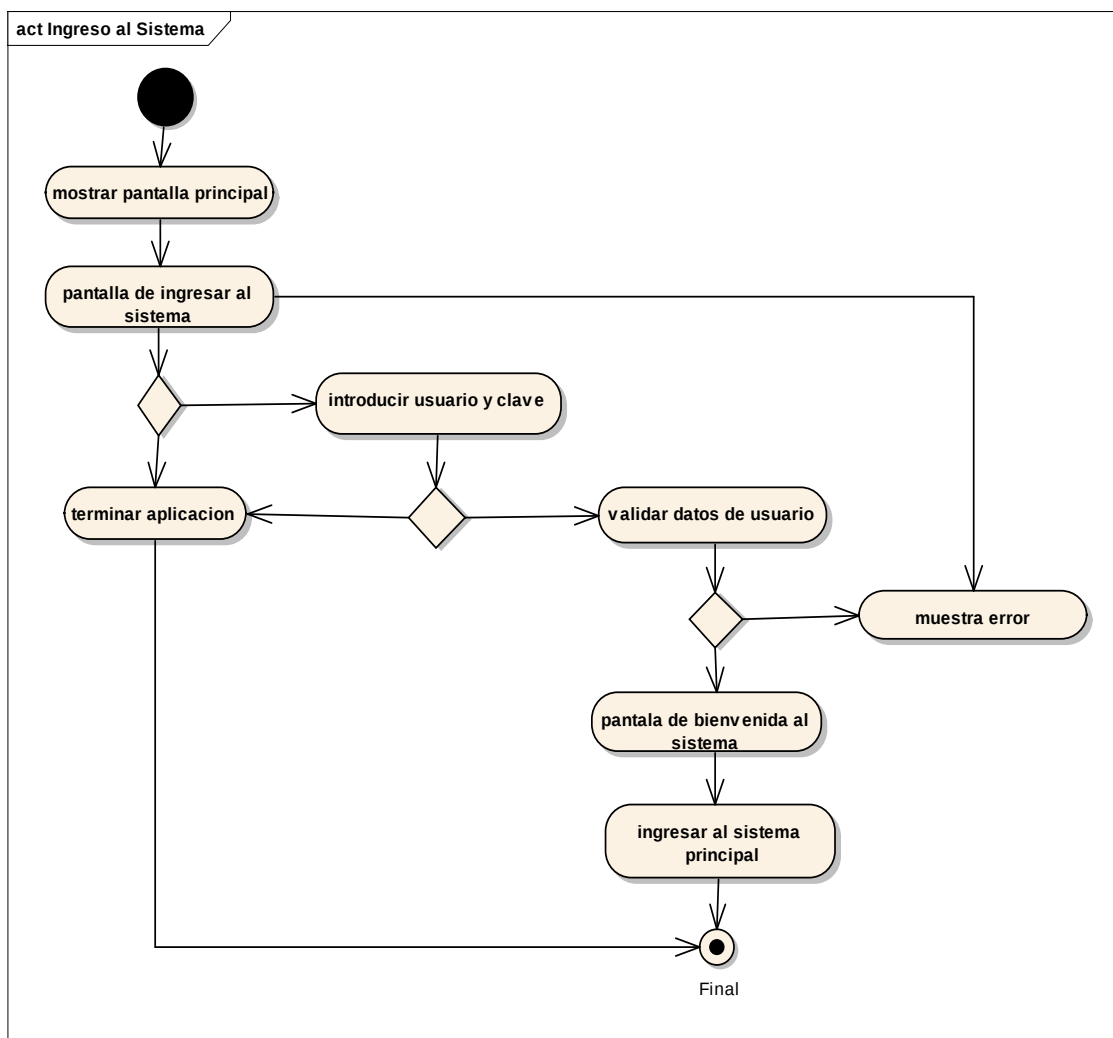
Comprender el funcionamiento de las actividades del Sistema

II.1.9.3. ALCANCE

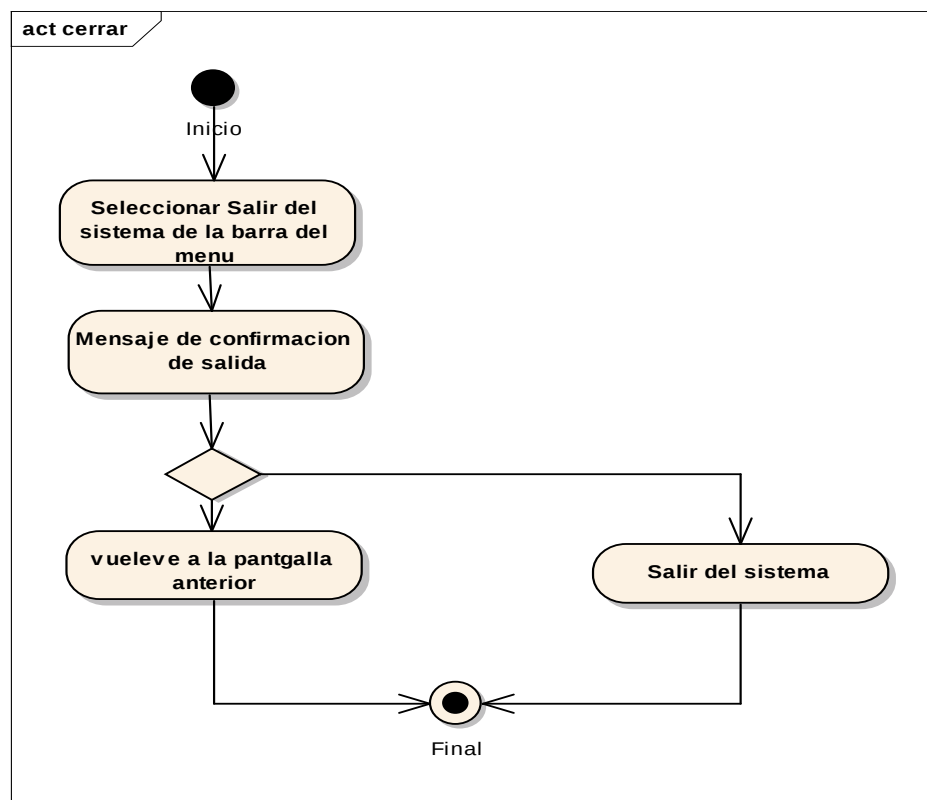
Describir los procesos internos del sistema en cada Actividad del Sistema

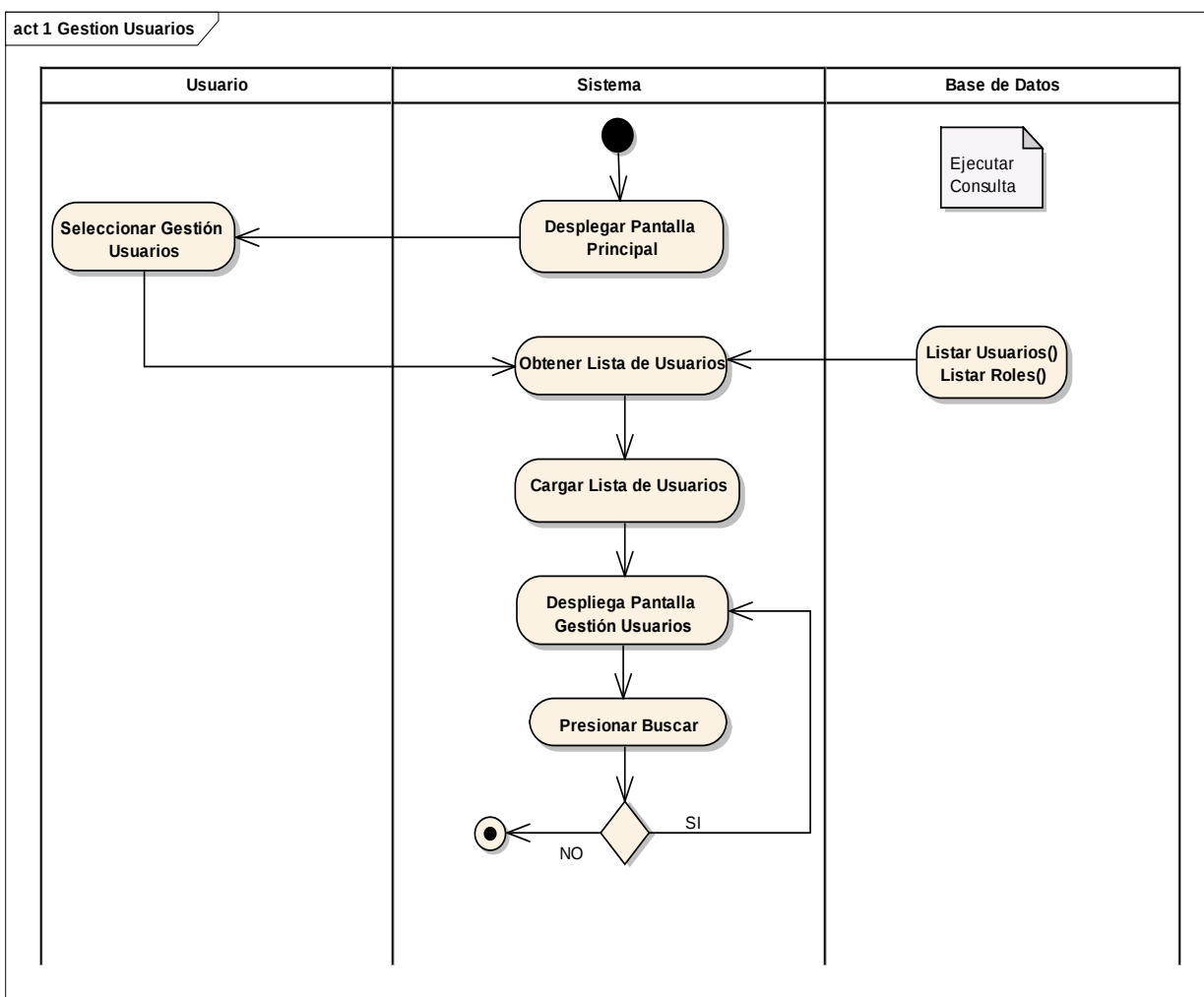
Detallar los flujos de cada actividad según lo establecido por la organización.

II.1.9.4. DIAGRAMA DE ACTIVIDAD INGRESO AL SISTEMA

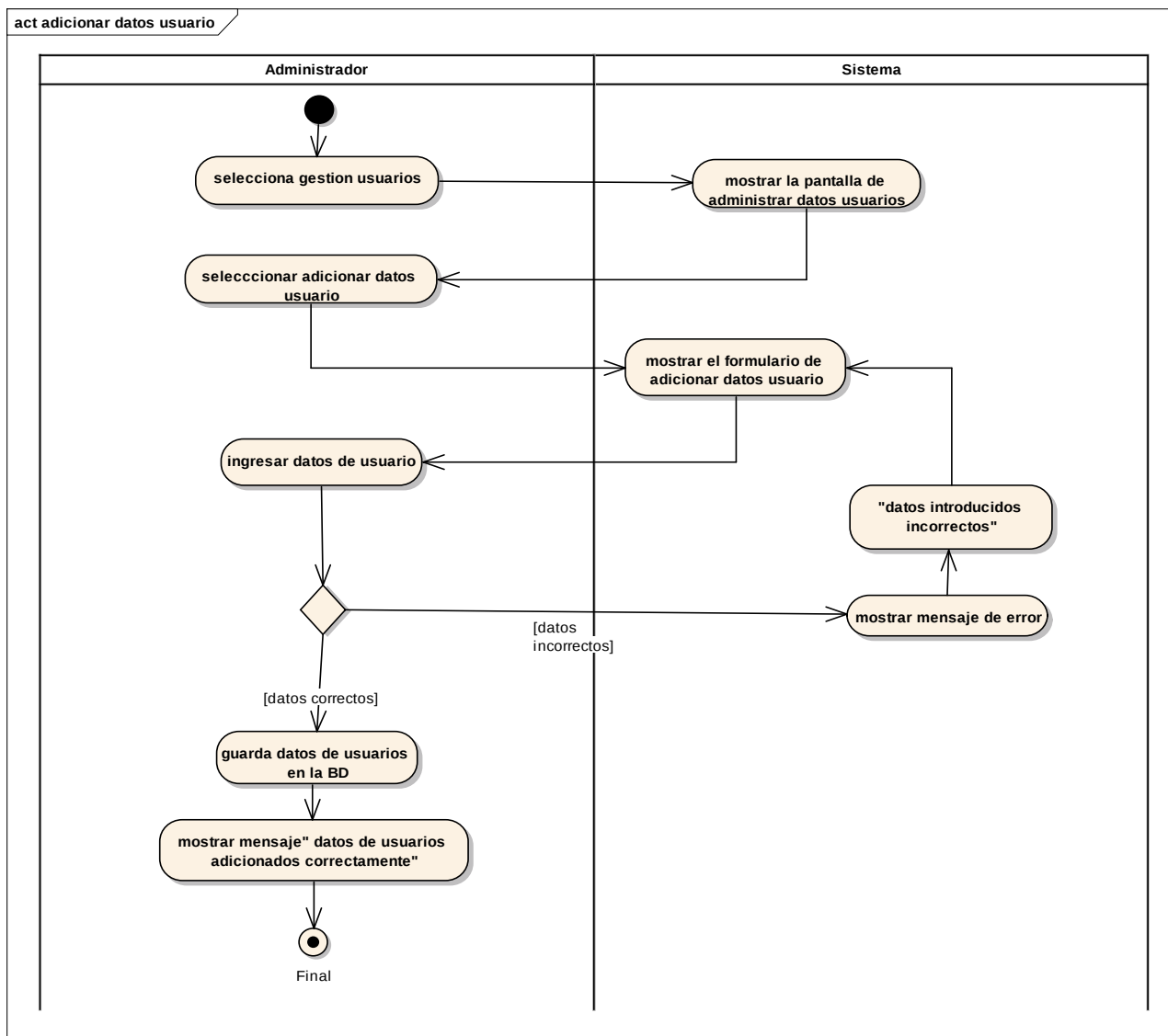


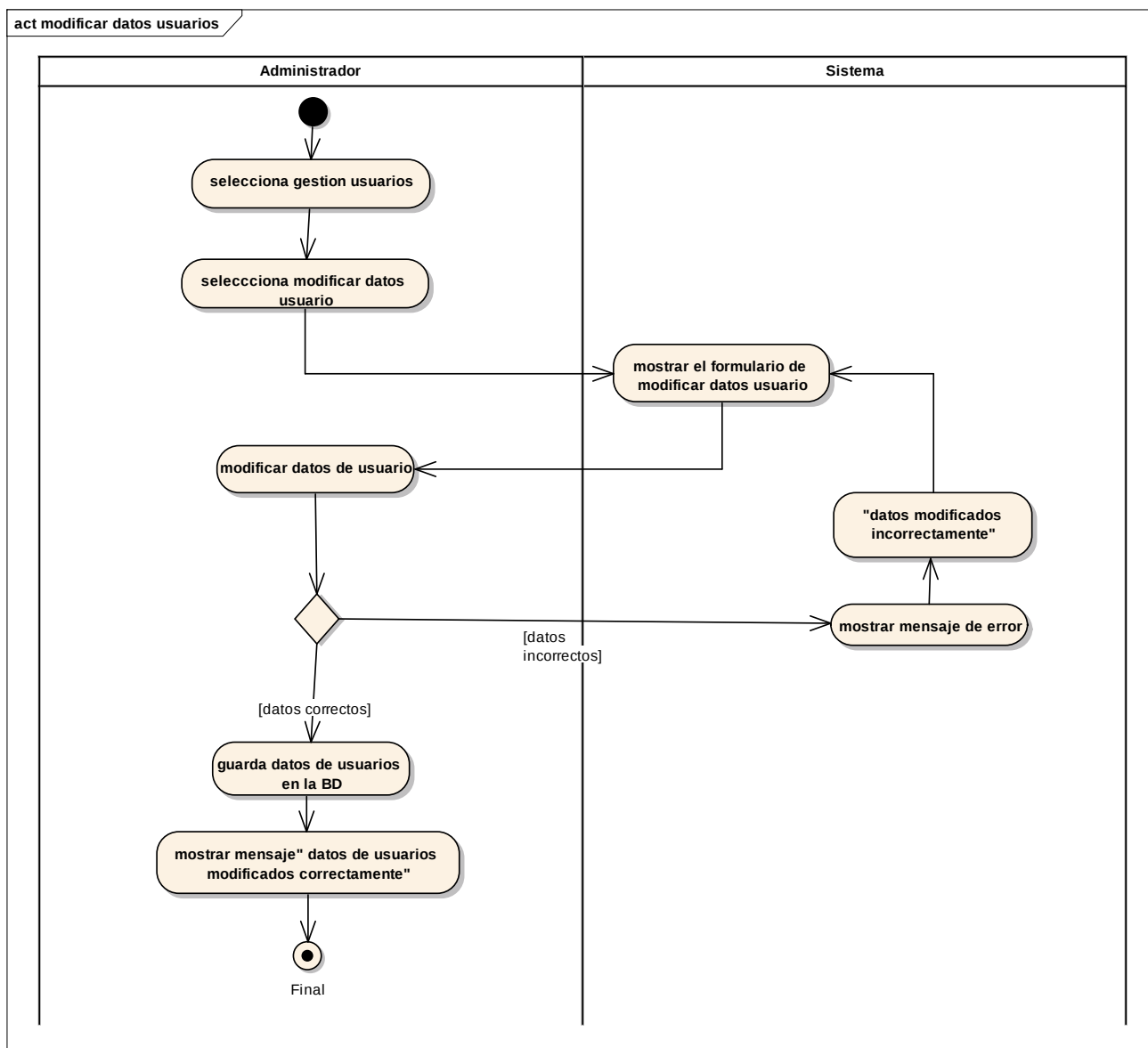
II.1.9.5. DIAGRAMA DE ACTIVIDAD CERRAR CESION



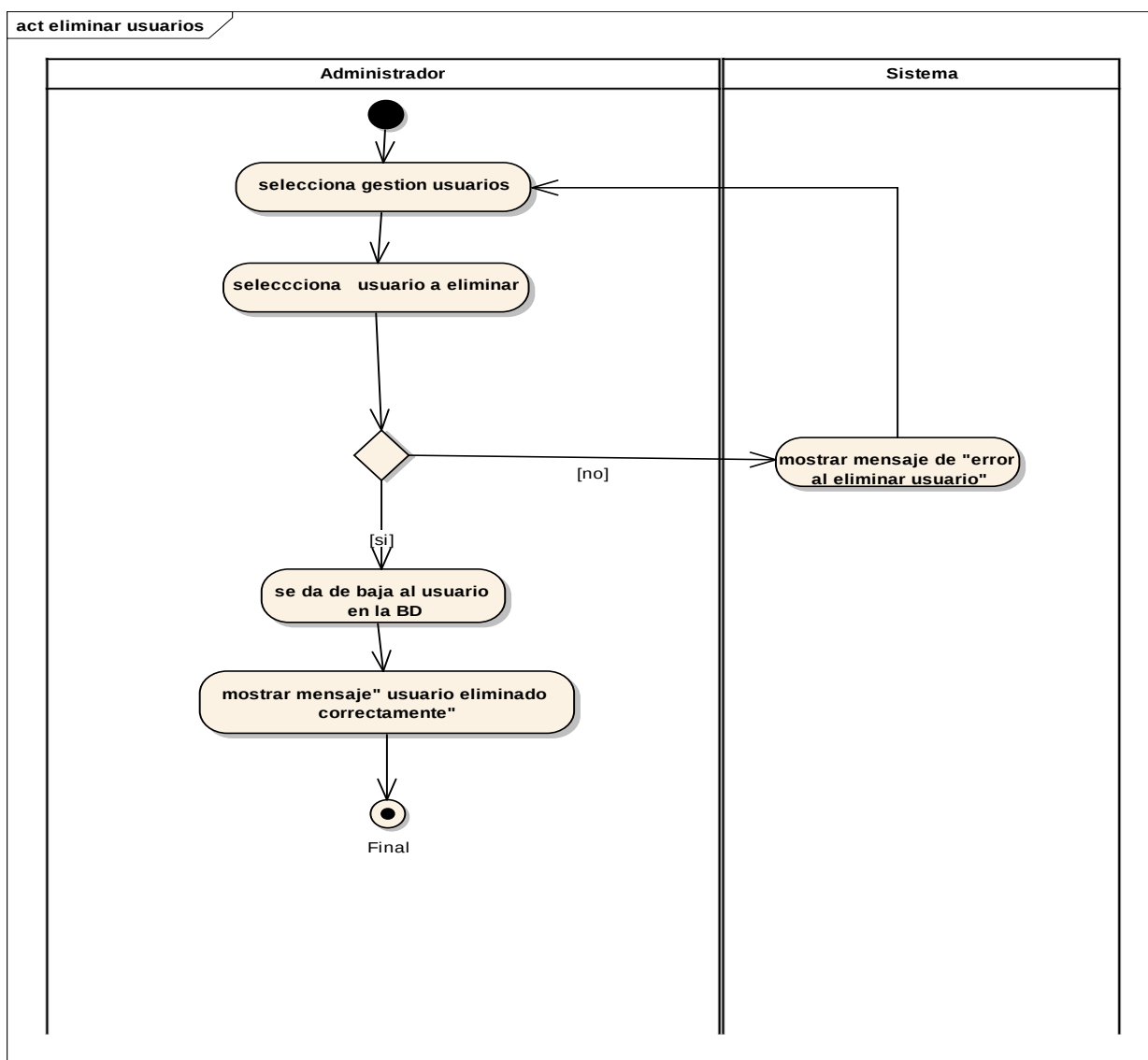
II.1.9.6. DIAGRAMA DE ACTIVIDAD GESTION USUARIOS

II.1.9.7. DIAGRAMA DE ACTIVIDAD ADICONAR USUARIO



II.1.9.8. DIAGRAMA DE ACTIVIDAD MODIFICAR USUARIO

II.1.9.9. DIAGRAMA DE ACTIVIDAD ELIMINAR USUARIO



II.1.10. MODELO DE ANALISIS Y DISEÑO

II.1.10.1. MODELADO DE DIAGRAMA DE CLASES

II.1.10.1.1. INTRODUCCION

El modelado de Diagrama de Clases es un artefacto de la disciplina de Análisis/Diseño en la Metodología Scrum que se está implementando.

II.1.10.1.2. PROPOSITO

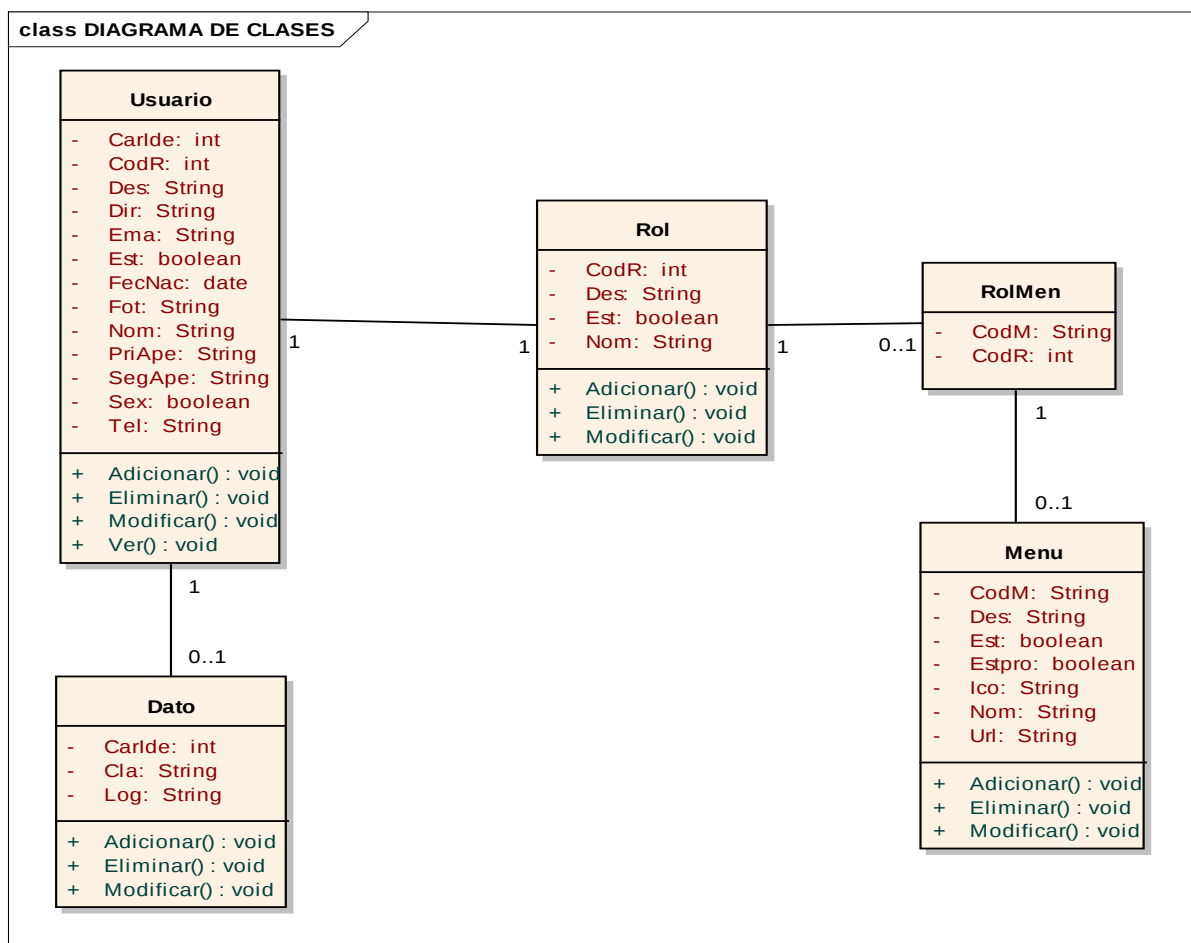
Comprender la estructura del sistema, Identificar clases de análisis y diseño.

II.1.10.1.3. ALCANCE

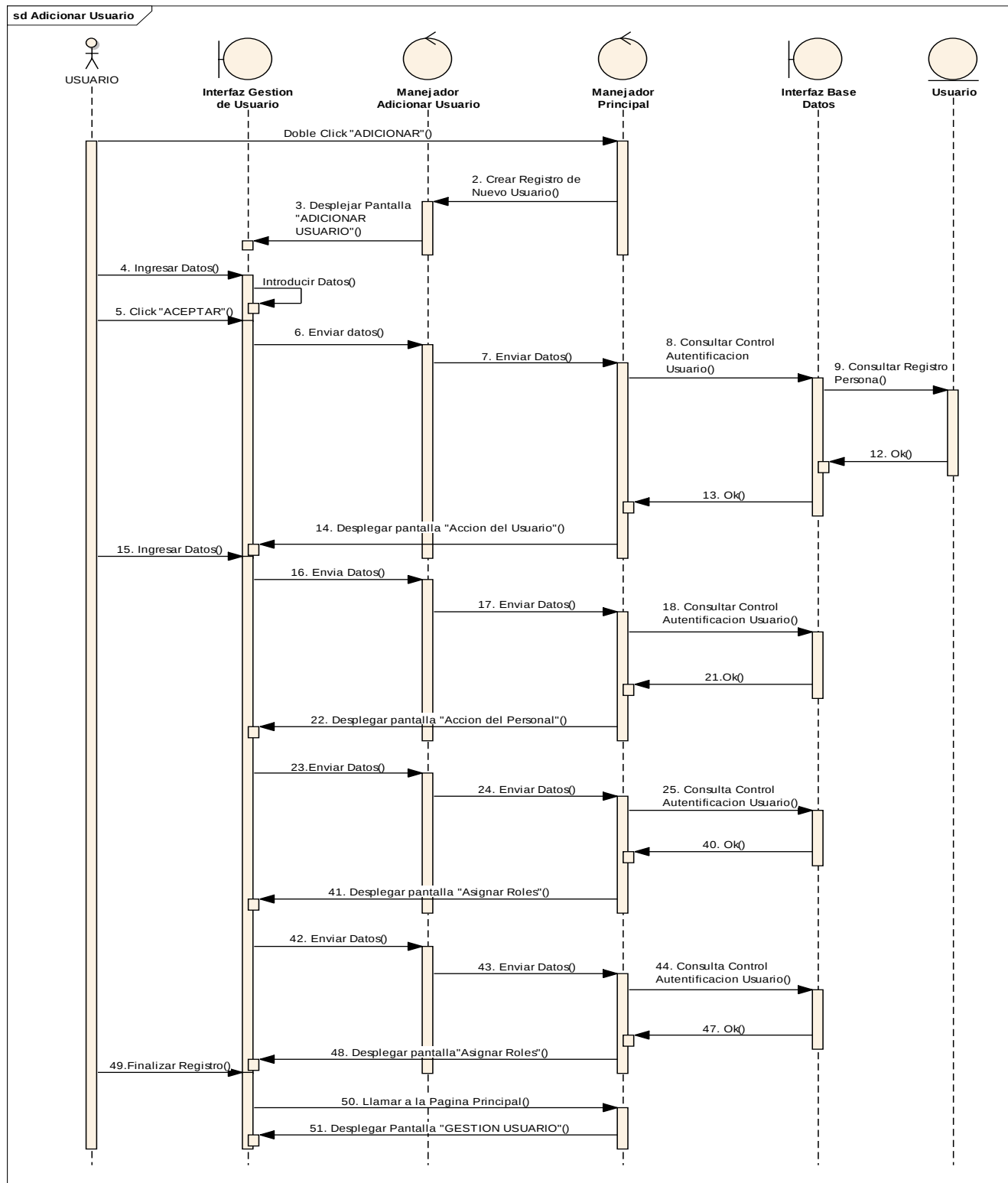
Describir las clases y objetos de diseño del sistema en su segunda iteración

Identificar y definir los objetos del sistema según los objetos del sistema deseado.

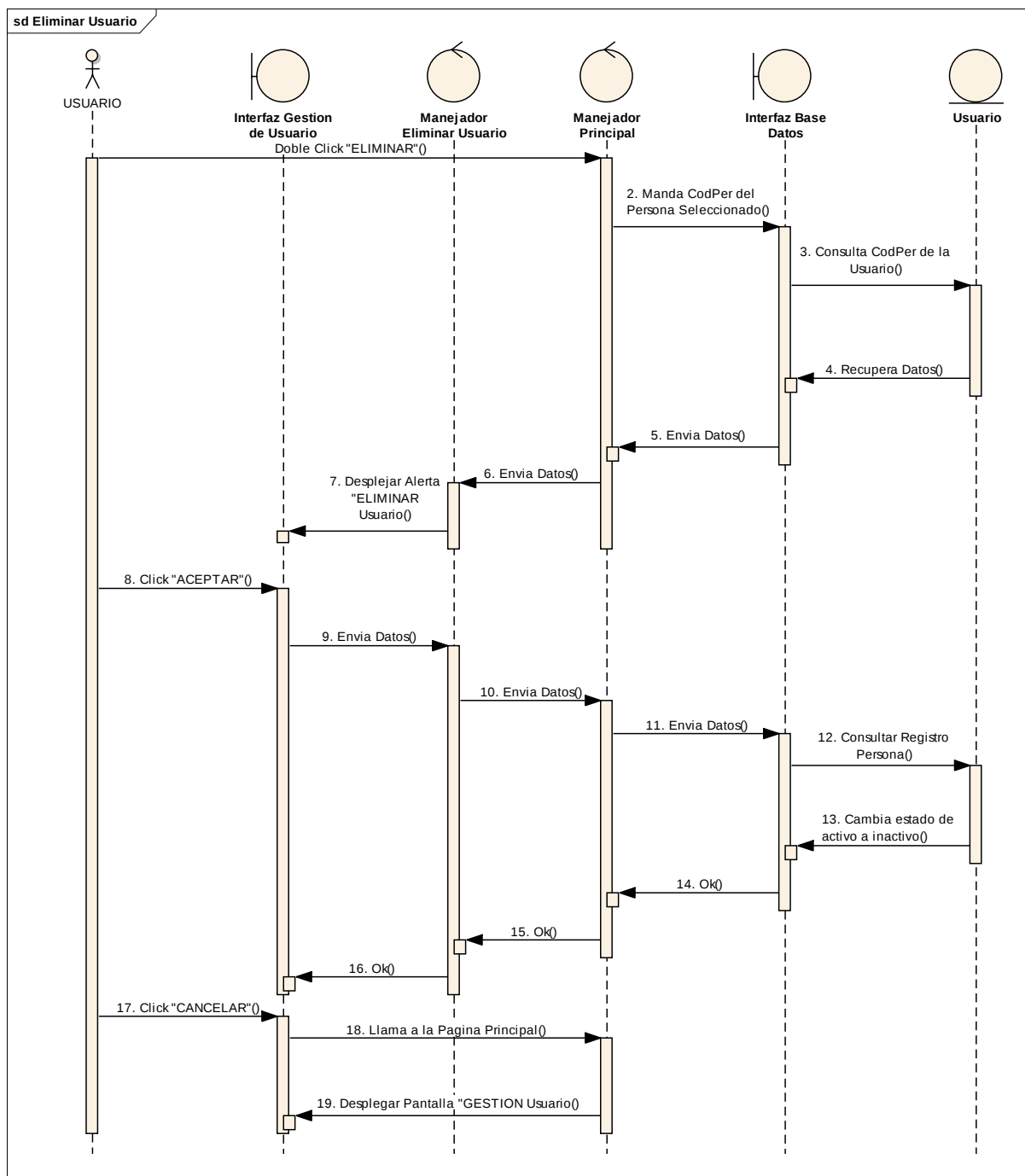
II.1.10.1.4. DIAGRAMA DE CLASES



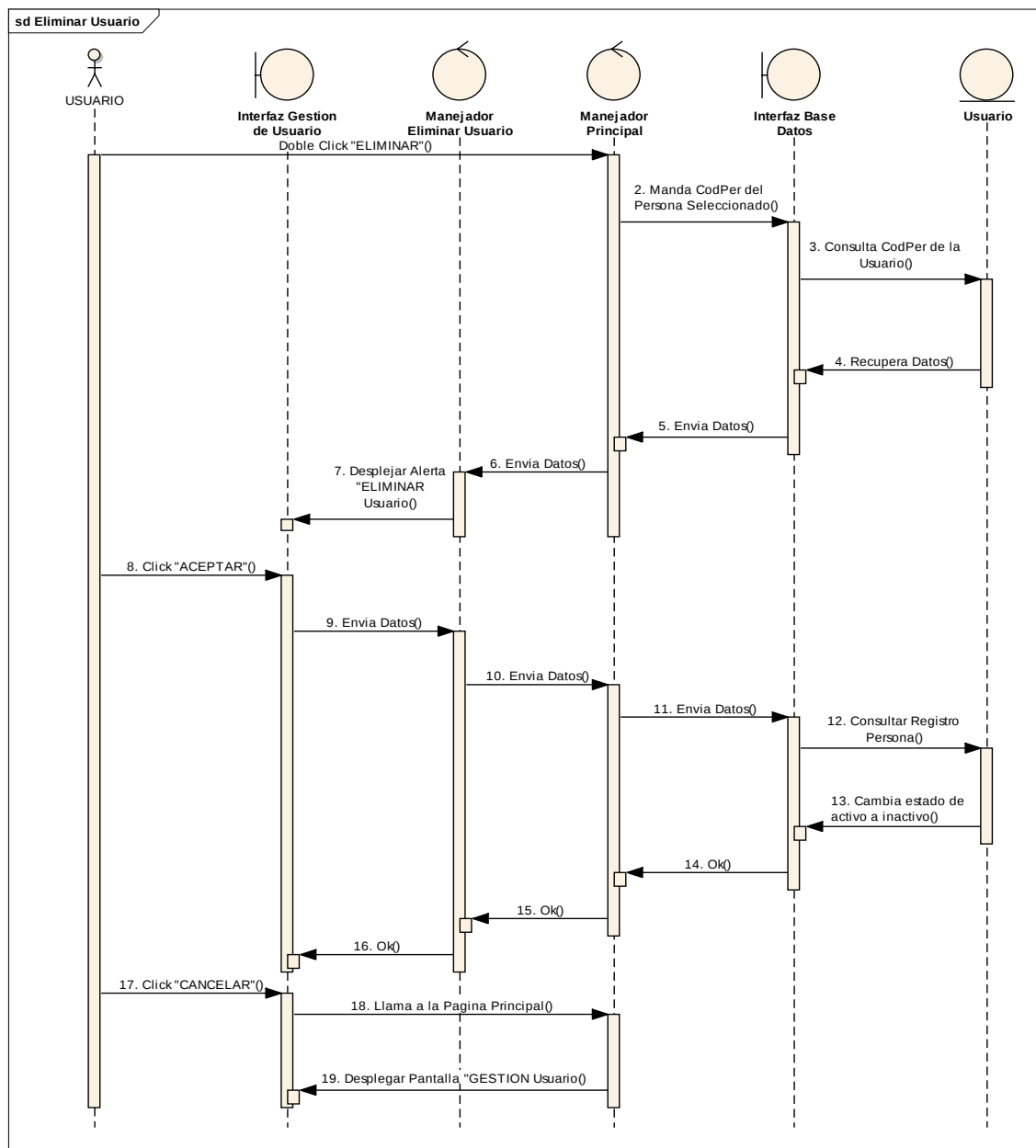
II.1.10.2.2. DIAGRAMA DE SECUENCIA ADICIONAR USUARIO



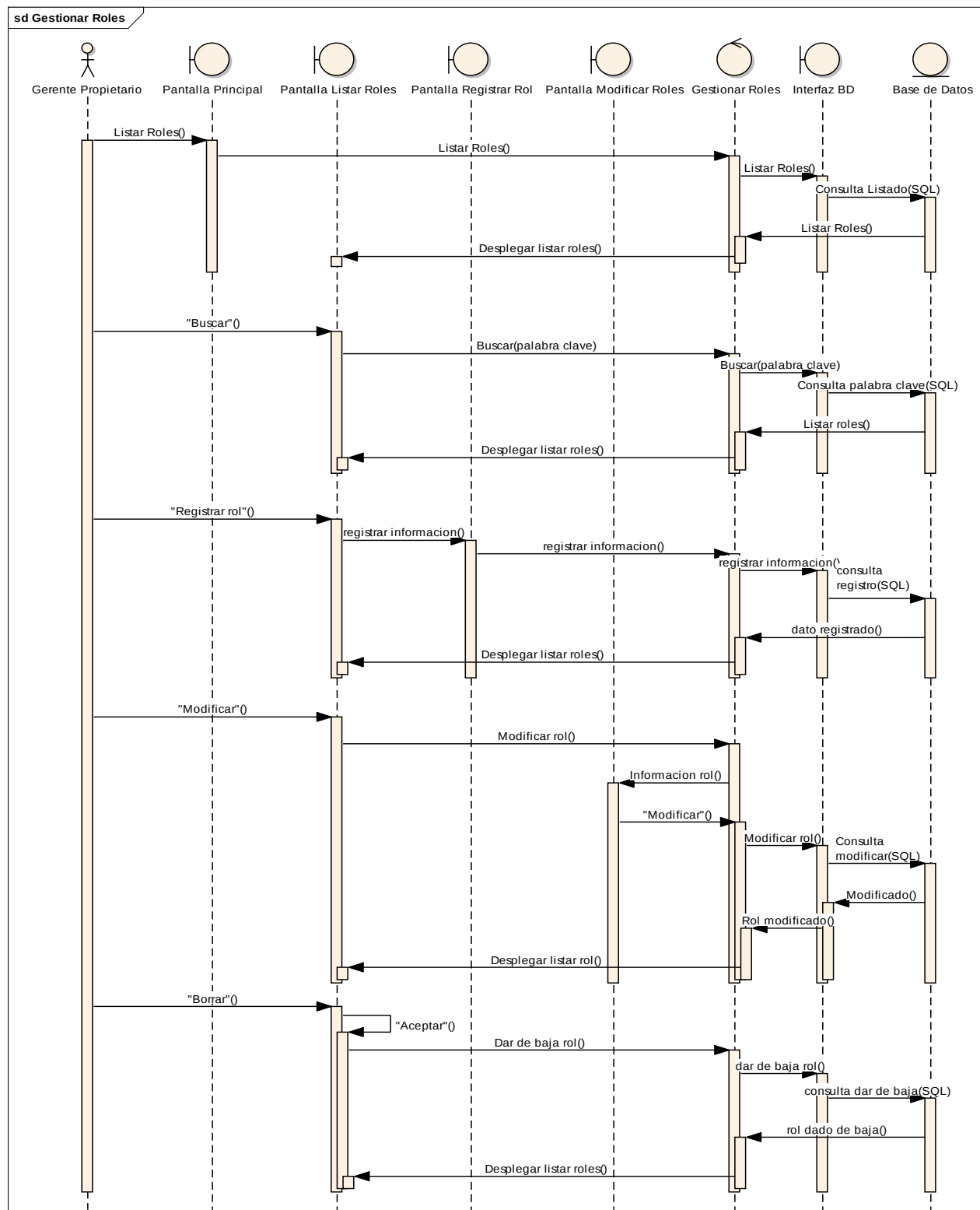
II.1.10.2.3. DIAGRAMA DE SECUENCIA MODIFICAR USUARIO



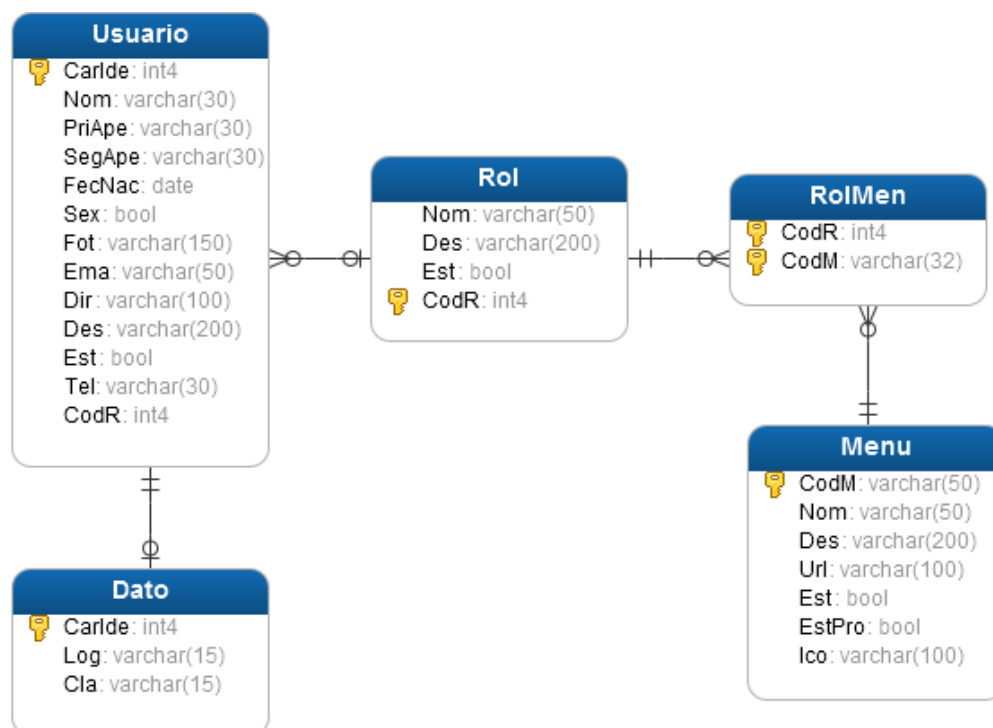
II.1.10.2.4. DIAGRAMA DE SECUENCIA ELIMINAR USUARIO



II.1.10.2.5. DIAGRAMA DE SECUNECIA GESTION ROL



II.1.11. DIAGRAMA ENTIDAD RELACION



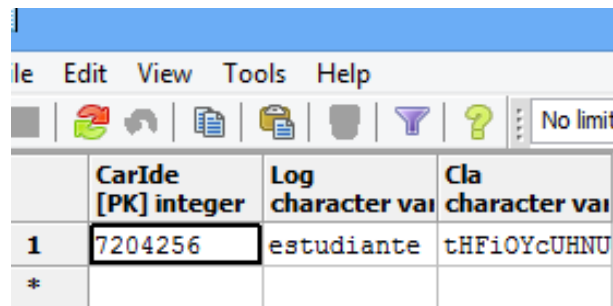
II.1.12. MODELO DE DATOS

II.1.12.1. BASE DE DATOS SQL

II.1.12.2. Creacion de la Base de datos

```
-- Database: "Riego"
-- DROP DATABASE "Riego";
CREATE DATABASE "Riego"
WITH OWNER = postgres
ENCODING = 'UTF8'
TABLESPACE = pg_default
LC_COLLATE = 'Spanish_Spain.1252'
LC_CTYPE = 'Spanish_Spain.1252'
CONNECTION LIMIT = -1;
```

II.1.12.3. Tabla Dato



The screenshot shows a database management interface with a menu bar (File, Edit, View, Tools, Help) and a toolbar. Below the toolbar is a table structure view for a table named 'Dato'. The table has four columns: 'CarIde' (integer, primary key), 'Log' (character varying), 'Cla' (character varying), and an unnamed column. The first row of data shows '1' in the first column, '7204256' in the second column, 'estudiante' in the third column, and 'tHFiOYcUHNU' in the fourth column. A row with an asterisk (*) is shown below the first row of data.

	CarIde [PK] integer	Log character vai	Cla character vai
1	7204256	estudiante	tHFiOYcUHNU
*			

```
-- Table: "Dato"
-- DROP TABLE "Dato";
CREATE TABLE "Dato"
(
    "CarIde" integer NOT NULL,
    "Log" character varying(15),
    "Cla" character varying(15),
    CONSTRAINT "Dato_pkey" PRIMARY KEY ("CarIde"),
```

```

CONSTRAINT "Dato_CarIde_fkey" FOREIGN KEY ("CarIde")
REFERENCES "Usuario" ("CarIde") MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "Dato"
OWNER TO postgres;

```

II.1.12.4. Tabla Menu

	CodM [PK] character	Nom character vai	Des character vai	Url character vai	Est boolean	EstPro boolean	Ico character vai
1	1	Gestion Usu	Se encarga	../usuario/	TRUE	FALSE	fa fa-user
2	4	Informacion	Se encarga	../informac	TRUE	FALSE	fa fa-file-
3	5	Gestion Rol	Se encarga	../rol/gest	TRUE	FALSE	fa fa-suitc
4	6	Gestion Rie	Se encarga	../riego/ge	TRUE	FALSE	fa fa-leaf
*							

```

-- Table: "Menu"
-- DROP TABLE "Menu";
CREATE TABLE "Menu"
(
  "CodM" character varying(50) NOT NULL,
  "Nom" character varying(50),
  "Des" character varying(200),
  "Url" character varying(100),
  "Est" boolean DEFAULT true,
  "EstPro" boolean DEFAULT true,
  "Ico" character varying(100),
  CONSTRAINT "Menu_pkey" PRIMARY KEY ("CodM")

```

```

)
WITH (
    OIDS=FALSE
);
ALTER TABLE "Menu"
    OWNER TO postgres;

```

II.1.12.5. Tabla Rol

	Nom character vari	Des character vari	Est boolean	CodR [PK] integer
1	Administrador	Encargado de	TRUE	1
2	Encargado	Encargado de	TRUE	2
*				

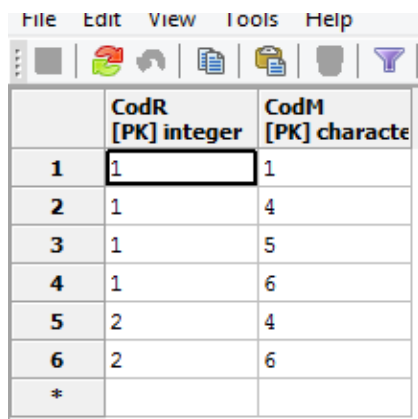
```

-- Table: "Rol"
-- DROP TABLE "Rol";
CREATE TABLE "Rol"
(
    "Nom" character varying(50),
    "Des" character varying(200),
    "Est" boolean DEFAULT true,
    "CodR" serial NOT NULL,
    CONSTRAINT "Rol_pkey" PRIMARY KEY ("CodR")
)
WITH (
    OIDS=FALSE

```

```
);
ALTER TABLE "Rol"
OWNER TO postgres;
```

II.1.12.6. Tabla Rol-Menu



	CodR [PK] integer	CodM [PK] character
1	1	1
2	1	4
3	1	5
4	1	6
5	2	4
6	2	6
*		

```
-- Table: "RolMen"
-- DROP TABLE "RolMen";
CREATE TABLE "RolMen"
(
    "CodR" integer NOT NULL,
    "CodM" character varying(32) NOT NULL,
    CONSTRAINT "RolMen_pkey" PRIMARY KEY ("CodR", "CodM"),
    CONSTRAINT "RolMen_CodM_fkey" FOREIGN KEY ("CodM")
        REFERENCES "Menu" ("CodM") MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "RolMen_CodR_fkey" FOREIGN KEY ("CodR")
        REFERENCES "Rol" ("CodR") MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

```

WITH (
    OIDS=FALSE
);
ALTER TABLE "RolMen"
    OWNER TO postgres;

```

II.1.12.7. Tabla Usuario

Edit Data - PostgreSQL 9.3 (localhost:5432) - Riego - Usuario													
File Edit View Tools Help													
No limit													
	CarIde [PK] integer	Nom character var	PriApe character var	SegApe character var	FecNac date	Sex boolean	Fot character var	Ema character var	Dir character var	Des character var	Est boolean	Tel character var	CodR integer
1	7204256	Ibsen	Copa	Mamani	1993-04-11	TRUE	user.png	paul.copa@h	Av. Oscar A	usuario pri	TRUE	72972778	1
2	7233535	Juan	Torrez	Vega	1860-06-11	TRUE	user.png	juan_torrez	av colon	solo encarg	TRUE	71198064	2
*													

```

-- Table: "Usuario"
-- DROP TABLE "Usuario";
CREATE TABLE "Usuario"
(
    "CarIde" integer NOT NULL,
    "Nom" character varying(30),
    "PriApe" character varying(30),
    "SegApe" character varying(30),
    "FecNac" date,
    "Sex" boolean,
    "Fot" character varying(150),
    "Ema" character varying(50),
    "Dir" character varying(100),
    "Des" character varying(200),
    "Est" boolean DEFAULT true,

```

```
"Tel" character varying(30),
"CodR" integer,
CONSTRAINT "Usuario_pkey" PRIMARY KEY ("CarIde"),
CONSTRAINT "Usuario_CodR_fkey" FOREIGN KEY ("CodR")
    REFERENCES "Rol" ("CodR") MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
);
ALTER TABLE "Usuario"
    OWNER TO postgres;
```

II.1.13. DICCIONARIO DE DATOS

Tabla Usuario

USUARIO						
NOMBRE	TIPO	TAM.	NULO	PK	FK	DESCRIPCION
CarIde	Entero		no	si		cedula de identidad del usuario
nombre	Texto	30	no			nombre del usuario
Ap	Texto	30	si			apellido paterno del usuario
Am	Texto	30	si			apellido materno del usuario
Fecnac	Date		no			fecha de nacimiento del usuario
Sexo	Booleano		no			genero del usuario
Foto	Texto	150	no			foto del usuario
Email	Texto	50	no			correo electrónico del usuario
dirección	Texto	100	no			dirección del usuario
descripción	Texto	200	no			descripción del usuario
Estado	Booleano		no			estado del usuario
teléfono	Texto	30	no			teléfono del usuario
CodRol	Entero		no	si	si	Código de rol

Tabla Rol

ROL						
NOMBRE	TIPO	TAM.	NULO	PK	FK	DESCRIPCION
nombre	Texto	50	no			nombre del rol
descripcion	Texto	200	no			descripción del rol
Estado	Booleano		no			estado del rol
Codr	Serie		no	si		código del rol

Tabla Rol-Menu

ROLMEN						
NOMBRE	TIPO	TAM.	NULO	PK	FK	DESCRIPCION
Codr	Entero		no	si	si	código de rol
Codm	Texto	30	no	si	si	código de menú

Tabla Dato

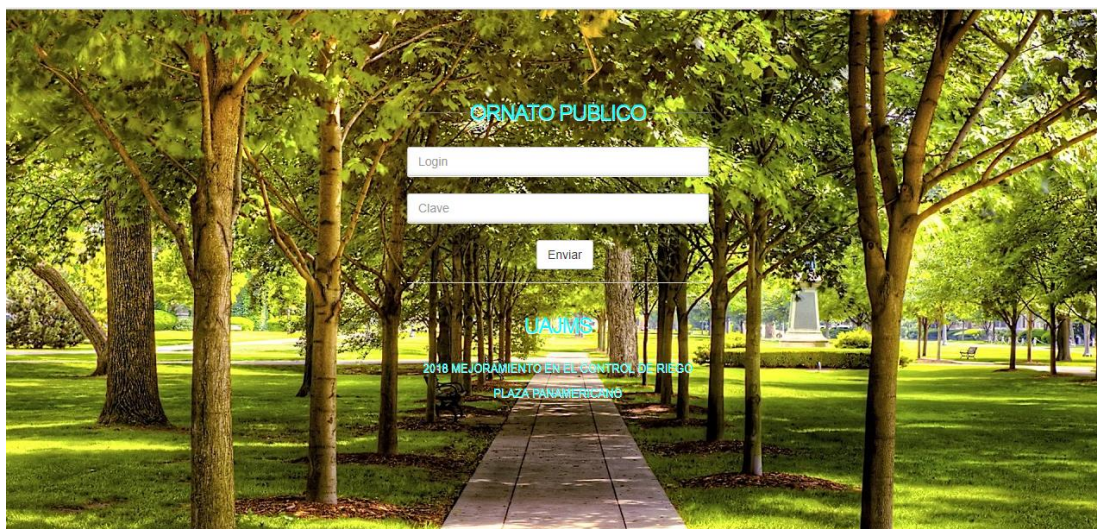
DATO						
NOMBRE	TIPO	TAM.	NULO	PK	FK	DESCRIPCION
CarIde	Entero		no	si		Carnet de identidad
Login	Texto	15	no			nombre de usuario
Clave	Texto	15	no			contraseña del usuario

Tabla Menu

MENU						
NOMBRE	TIPO	TAM.	NULO	PK	FK	DESCRIPCION
Codm	texto	50	no	si		código del menú
Nombre	texto	50	no			nombre del menú
descripción	texto	200				descripción del menú
Codpad	texto	50	no		si	código del proceso
url	texto	100	no			dirección del menú
Estado	booleano		no			estado del menú

II.1.14. DISEÑO DE PANTALLAS DE USUARIO

II.1.14.1. PANTALLA PRINCIPAL



En esta pantalla el usuario debe proporcionar sus datos de acceso como ser su usuario y clave.

II.1.14.2. PANTALLA GESTION USUARIOS

Carnet	Nombres y Apellidos	Borrar clave	login/clave	Modificar	Eliminar	Ver
7204256	Ibsen Copa M.					
7233535	Juan Torrez V.					

Pantalla con la lista de todos los usuarios activos, registrados en el sistema excepto el usuario actual logueado.

II.1.14.3. PANTALLA ADICIONAR USUARIO

The screenshot shows the 'Nuevo Usuario' (New User) form within the 'Ornato Publico' application. The interface features a green top header with the application name and user information, and a dark blue sidebar with navigation links. The form itself is titled 'Nuevo Usuario' and contains several input fields for user registration. The fields are: 'Carnet de Identidad' (a single-line text input), 'Nombre' (a single-line text input), 'Primer Apellido' (a single-line text input), 'Segundo apellido' (a single-line text input), 'Fecha de Nacimientos' (a date picker), 'Genero' (radio buttons for 'M' and 'F'), 'Rol' (a dropdown menu currently set to 'Administrador'), 'Foto' (a file upload button labeled 'Seleccionar archivo' and a text area showing 'Ningún archivo seleccionado'), 'Direccion' (a single-line text input), 'Email *' (a single-line text input with an asterisk indicating it is required), 'Telefono *' (a single-line text input with an asterisk indicating it is required), and 'Descripcion' (a multi-line text area). At the bottom of the form, there are two buttons: 'Cancelar' (red) and 'Guardar' (green).

Ornato Publico

Administrador Ibsen Copa M.

Informacion

Gestion Roles

Gestion Riego

Gestion Usuarios

Nuevo Usuario

Carnet de Identidad

Nombre

Primer Apellido

Segundo apellido

Fecha de Nacimientos

Genero :

M: ☒ F: ☐

Rol: Administrador

Foto: Seleccionar archivo Ningún archivo seleccionado

Direccion

Email * :

Telefono * :

Descripcion :

Cancelar Guardar

Pantalla con el formulario para agregar un nuevo usuario, el administrador debe proporcionar los datos como ser nombre, teléfono, clave y rol, los demás campos son opcionales.

II.1.14.4. PANTALLA MODIFICAR USUARIO

MODIFICAR USUARIO

Carnet de Identidad: 7204256

Nombre: Ibsen

Primer Apellido: Copa

Segundo apellido: Mamani

Fecha de Nacimientos: 11/04/1993

Genero :
M: ☒ F: ☐

Rol: Administrador ▼

Foto anterior:

Direccion: Av. Oscar Alfaro

Email * : paul.copa@hotmail.com

Telefono * : 72972778

Descripcion : usuario principal

Cancelar Guardar

Pantalla con el formulario para modificar a un usuario de la lista, el administrador puede modificar los datos como ser nombre, teléfono y rol, los demás campos son opcionales.

II.1.14.5. PANTALLA ASIGNAR LOGIN Y CLAVE A USUARIO

Crear clave

Usuario: Juan Torrez V.

Login:

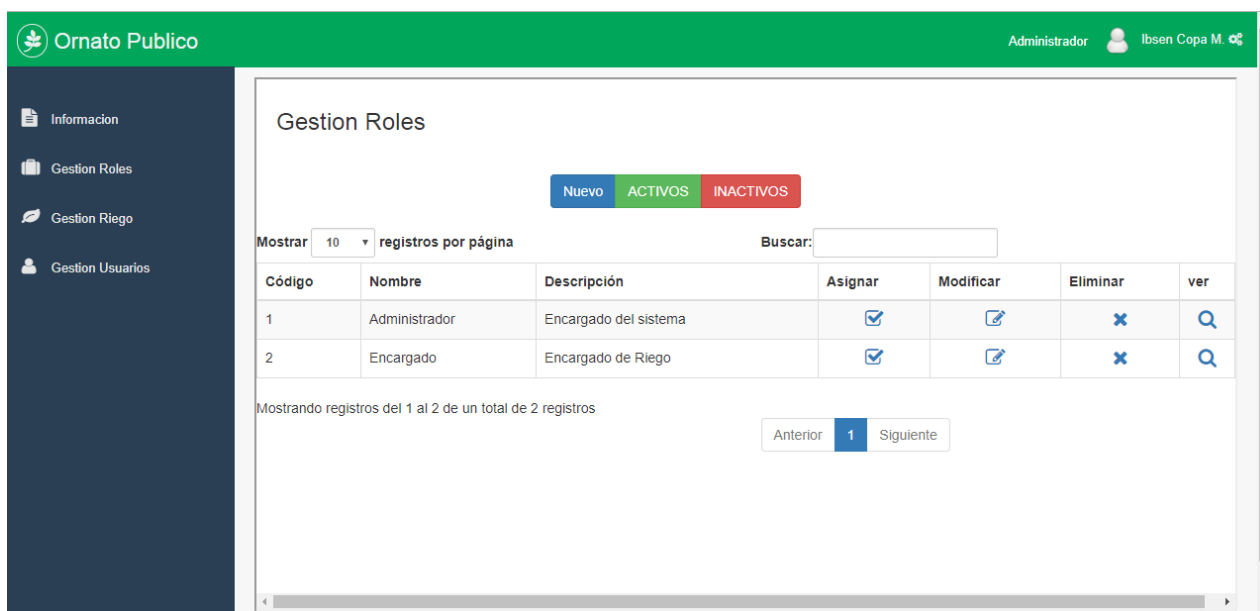
Clave:

Repetir clave:

Cancelar Guardar

Pantalla con el formulario para asignar a un usuario de la lista, un login y una contraseña para que este pueda ingresar al sistema.

II.1.14.6. PANTALLA GESTION ROLES



Ornato Publico

Administrador Ibsen Copa M.

Informacion

Gestion Roles

Gestion Riego

Gestion Usuarios

Gestion Roles

Nuevo ACTIVOS INACTIVOS

Mostrar 10 registros por página

Buscar:

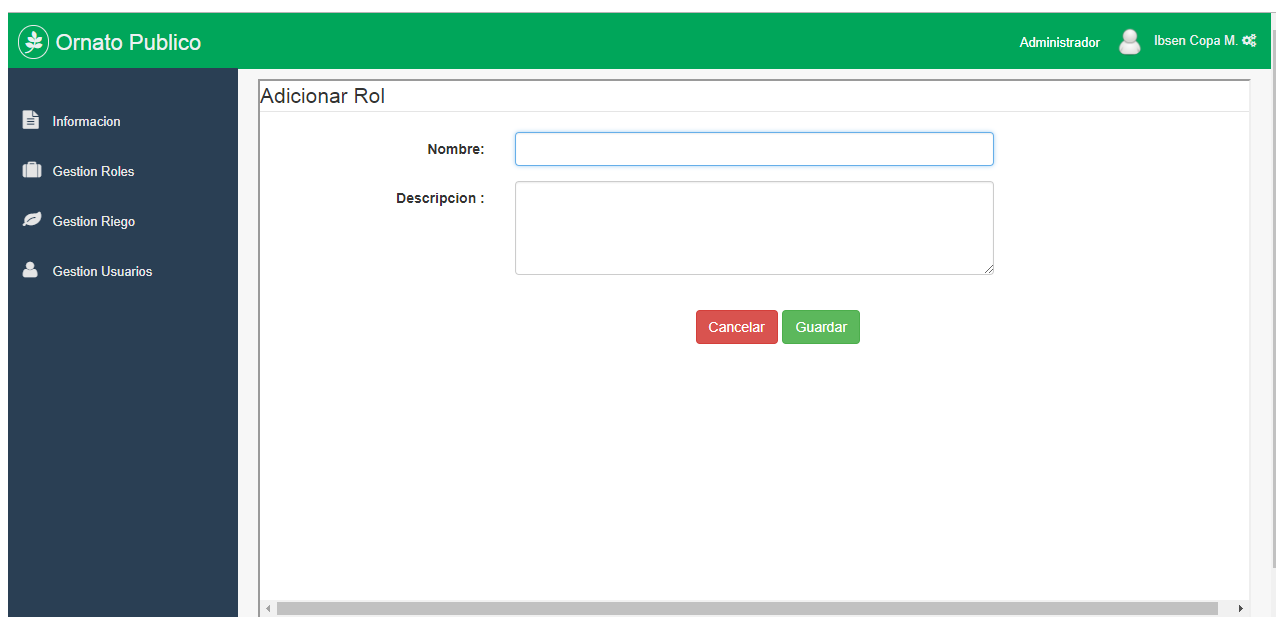
Código	Nombre	Descripción	Asignar	Modificar	Eliminar	ver
1	Administrador	Encargado del sistema	☒			
2	Encargado	Encargado de Riego	☒			

Mostrando registros del 1 al 2 de un total de 2 registros

Anterior 1 Siguiente

Pantalla donde se visualizan los roles del sistema, solo se tiene dos roles en el sistema y cada usuario solo puede tener un rol asignado.

II.1.14.7. PANTALLA ADICIONAR ROL



Ornato Publico

Administrador Ibsen Copa M.

Informacion

Gestion Roles

Gestion Riego

Gestion Usuarios

Adicionar Rol

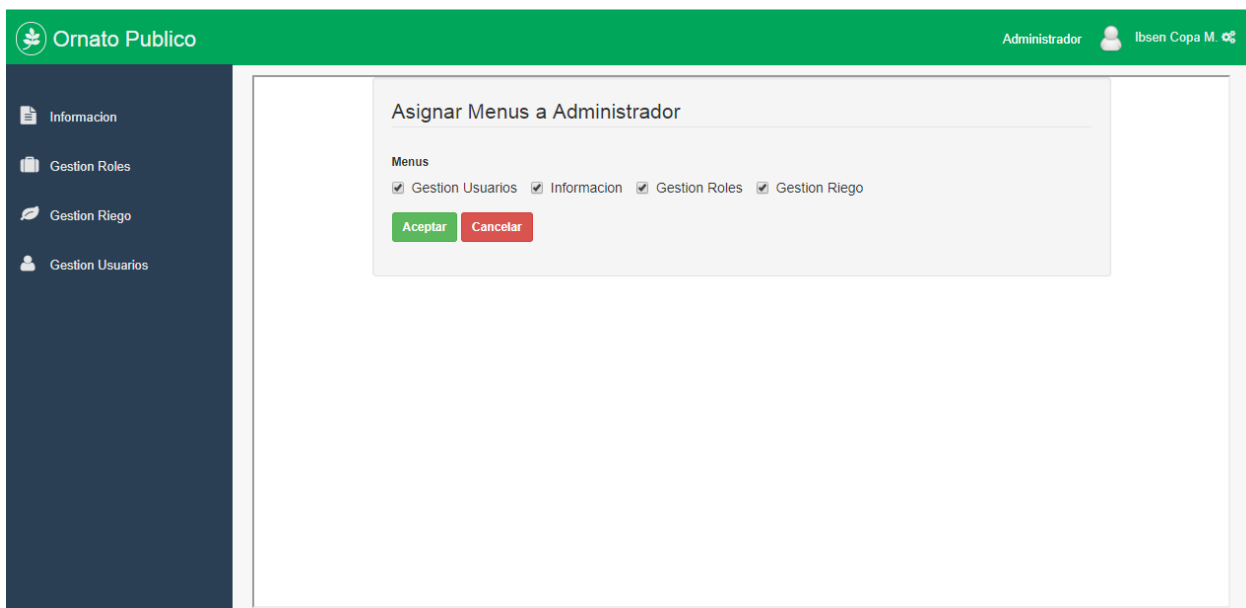
Nombre:

Descripcion :

Cancelar Guardar

Pantalla con el formulario para agregar un nuevo rol, el administrador debe proporcionar los datos como ser nombre y descripción.

II.1.14.8. PANTALLA ASIGNAR MENUS A USUARIO



Pantalla con el menú de opciones para asignar los menús a un rol seleccionado como también se puede desasignar.

II.1.14.9. PANTALLA VER ROL

The screenshot displays the 'Ver Rol' (View Role) interface within the 'Ornato Publico' system. The interface features a green header bar with the system name and a dark blue sidebar with navigation options. The main content area shows the details of a selected role.

Ornato Publico (Header)

Administrador (User)

Ver Rol (Section Title)

Nombre: Administrador

Descripcion : Encargado del sistema

Volver (Button)

Pantalla con el formulario que nos permite ver el nombre del rol seleccionado con su determinada descripción.

II.1.14.10. PANTALA GESTION RIEGO

Omato Público

Administrador

Información

Gestión Roles

Gestión Riego

Gestión Usuarios

Gestión de Riego

☐ Apagar Bomba

Area # 1	Area # 2	Area # 3	Area # 4	Area # 5	Area # 6	Area # 7	Area # 8
☐ On ☐ Off	☐ On ☐ Off	☐ On ☐ Off	☐ On ☐ Off	☐ On ☐ Off	☐ On ☐ Off	☐ On ☐ Off	☐ On ☐ Off

Diagrama de un sistema de riego con 8 áreas numeradas (1 a 8) y una bomba central. El diagrama muestra las líneas de riego y las áreas de cultivo. El texto "BARRIO PANAMERICANO" es visible en el diagrama.

Pantalla con distintos botones relacionados con los sensores de humedad que nos permiten realizar el riego a distintas áreas verdes como también poder detener el riego apagando la bomba de agua en caso de emergencia.

II.1.14.11. PANTALLA GESTION INFORMACION

 Ornato Publico

Administrador  Ibsen Copa M. 

 Informacion

 Gestion Roles

 Gestion Riego

 Gestion Usuarios

Gestión de Informacion

Informacion de los Sensores


Area verde 1
Tempratura:


Area verde 2
Tempratura:


Area verde 3
Tempratura:


Area verde 4
Tempratura:


Area verde 5
Tempratura:


Area verde 6
Tempratura:


Area verde 7
Tempratura:


Area verde 8
Tempratura:

Informacion del Pozo

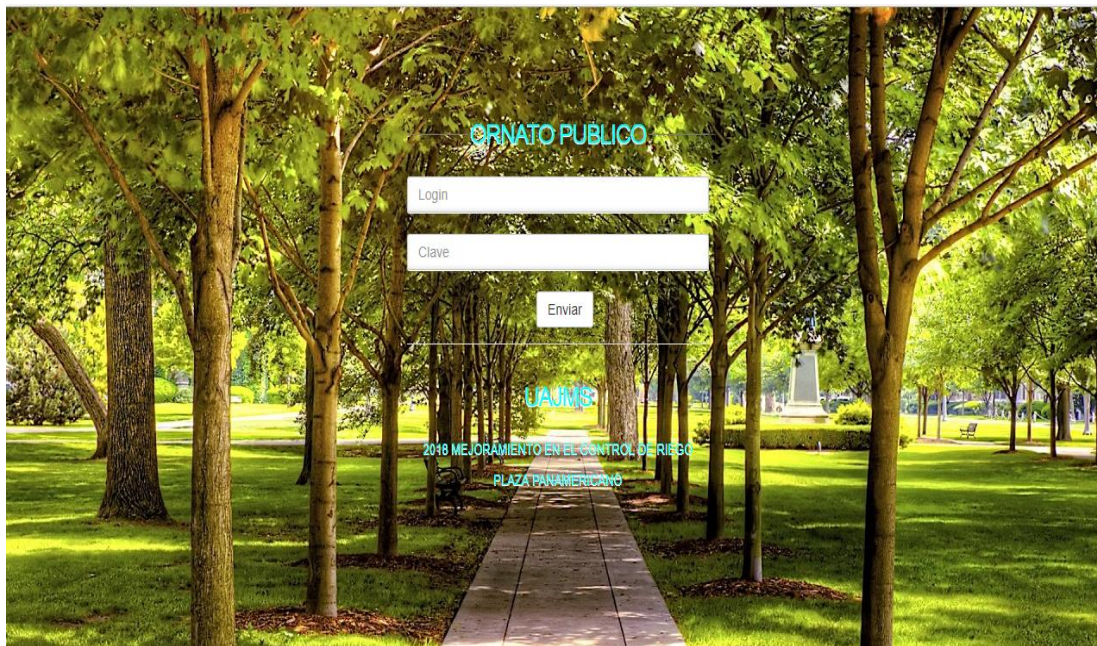

Pozo
Nivel:

Imprimir

Pantalla con la información de todos los sensores de humedad ubicados en las áreas verdes tanto como el nivel de agua en el pozo.

II.1.15. CASOS DE PRUEBA

II.1.15.1. INGRESAR AL SISTEMA



Condición	Tipo	Condición válida	Condición Inválida
Usuario	Valor	1. Cadena alfanumérica de cinco a quince caracteres.	2. Cadena menor a cinco caracteres. 3. Cadena mayor a quince caracteres.
Clave	Valor	1. Cadena alfanumérica de cinco a quince caracteres.	2. Cadena menor a cinco caracteres. 3. Cadena mayor a quince caracteres.

II.1.15.2. REGISTRO DE USUARIOS

Ornato Publico Administrador Susan Cope M

Nuevo Usuario

Carnet de Identidad

Nombre

Primer Apellido

Segundo apellido

Fecha de Nacimientos

Genero :
 M ☐ F ☐ O ☐

Rol

Foto Ningún archivo seleccionado

Direccion

Email * :

Telefono * :

Descripción :

Condición	Tipo	Condición válida	Condición Inválida
Cedula de Identidad	Valor	1. numero de 7 u 8 dígitos	2. número menor a 7 dígitos 3. número mayor a 8 dígitos 4. Cualquier carácter no numérico
Nombre	Rango	1. Cadena que no sobrepase los 25 caracteres.	2. En blanco. 3. Cadena > a 30 caracteres. 4. Cadena menor a 3 caracteres.
Apellido Paterno	Rango	1. Cadena que no sobrepase los 25 caracteres.	2. En blanco. 3. Cadena > a 25 caracteres.
Apellido Materno	Lógico Si está es Rango	1. Cadena <= a 25 caracteres.	2. Cadena > a 25 caracteres.
Email	Lógico Si está es Rango	1. Cadena que no sobre pase los 30 caracteres	2. Cadena > a 30 caracteres
Teléfono	Rango	1. 1000000 <= Teléfono <= 9999999	2. Cadena no numérica 3. T < 1000000

			4. T >9999999
Seleccionar Rol	Conjunto de valores	1. Solo selecciona un valor.	2. No selecciona ninguno

II.1.15.3. MODIFICAR INFORMACION USUARIO

The screenshot shows the 'MODIFICAR USUARIO' form in the Ornato Publico system. The form is displayed in a web browser window. The header bar is green and contains the text 'Ornato Publico' and 'Administrador'. The left sidebar is dark blue and contains the text 'Informacion', 'Gestion Roles', 'Gestion Riego', and 'Gestion Usuarios'. The main content area is white and contains the form fields. The fields are: Cedula de Identidad (7204256), Nombre (Itzen), Primer Apellido (Copa), Segundo apellido (Mamani), Fecha de Nacimiento (11/04/1993), Genero (M), Rol (Administrador), Foto anterior (placeholder image), Direccion (Av. Oscar Alfaro), Email (paul.copa@hotmail.com), Telefono (72572778), and Descripcion (usuario principal). At the bottom of the form are two buttons: 'Cancelar' (red) and 'Guardar' (green).

Condición	Tipo	Condición válida	Condición Inválida
Cedula de Identidad	Valor	1. numero de 7 u 8 dígitos	2. número menor a 7 dígitos 3. número mayor a 8 dígitos 4. Cualquier carácter no numérico

Nombre	Rango	1. Cadena que no sobrepase los 25 caracteres.	2. En blanco. 3. Cadena > a 30 caracteres. 4. Cadena menor a 3 caracteres.
Apellido Paterno	Rango	1. Cadena que no sobrepase los 25 caracteres.	2. En blanco. 3. Cadena > a 25 caracteres.
Apellido Materno	Lógico Si está es Rango	1. Cadena <= a 25 caracteres.	2. Cadena > a 25 caracteres.
Email	Lógico Si está es Rango	2. Cadena que no sobre pase los 30 caracteres	2. Cadena > a 30 caracteres
Teléfono	Rango	1. 1000000 <= Teléfono <= 9999999	2. Cadena no numérica 3. T < 1000000 4. T > 9999999
Seleccionar Rol	Conjunto de valores	1. Solo selecciona un valor.	2. No selecciona ninguno

II.1.15.4. ASIGNAR LOGIN Y CONTRASEÑA A USUARIO

The screenshot shows the 'Ornato Publico' web application interface. The top navigation bar is green with the logo and name 'Ornato Publico' on the left, and the user role 'Administrador' and profile 'Ibsen Copa M.' on the right. A dark blue sidebar on the left contains menu items: 'Informacion', 'Gestion Roles', 'Gestion Riego', and 'Gestion Usuarios'. The main content area is titled 'Crear clave' and displays the user 'Juan Torrez V.' with the following fields:

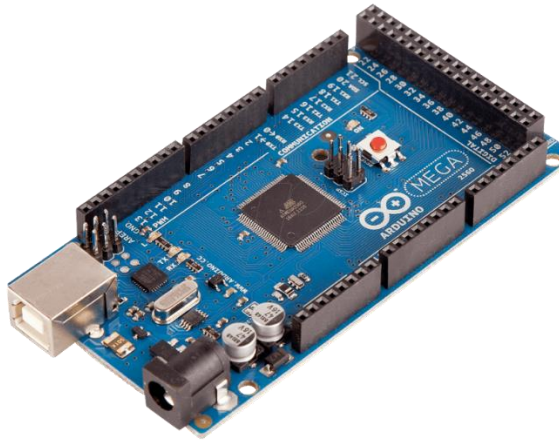
- Usuario:** Juan Torrez V.
- Login:** [Empty text input field]
- Clave:** [Empty password input field]
- Repetir clave:** [Empty password input field]

At the bottom of the form are two buttons: a red 'Cancelar' button and a green 'Guardar' button.

Condición	Tipo	Condición válida	Condición Inválida
Nombre de Usuario	Valor	1. Cadena alfanumérica de 25 caracteres.	2. Cadena > a 25 caracteres. 3. Cadena en blanco.
Contraseña	Valor	1. Cadena alfanumérica de cinco a quince caracteres.	2. Cadena menor a cinco caracteres. 3. Cadena mayor a quince caracteres. 4. Cadena en blanco.
Repetir contraseña	Valor	1. Cadena alfanumérica de cinco a quince caracteres que coincida con el campo anterior.	2. Cadena menor a cinco caracteres. 3. Cadena mayor a quince caracteres. 4. Cadena en blanco. 5. Cadena distinta al campo anterior.

II.1.16. DESCRIPCION DE LOS COMPONENTES UTILIZADOS EN EL PROYECTO PARA LA MAQUETA

II.1.16.1. ARDUINO MEGA 2560 R3



El Arduino Mega es una tarjeta basada en el ATmega2560. Tiene 54 pines digitales de I/O (14 de los cuales pueden ser usados como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos seriales en hardware), un cristal oscilador a 16MHz, conexión USB, jack para conexión a la electricidad, un header ICSP y un botón de reset. Cuenta con todo lo necesario para dar soporte a este microcontrolador; simplemente conéctalo a una computadora mediante un cable usb.

Características:

- Microcontrolador ATmega2560
- Voltaje de entrada: 7 a 12V
- 54 pines digitales I/O (14 salidas PWM)
- 16 entradas analógicas
- 256k de memoria flash
- Velocidad de reloj 16MHz

II.1.16.2. SENSOR DE HUMEDAD DE SUELO



Este sensor de humedad puede leer la cantidad de humedad presente en el suelo que lo rodea. Se trata de una herramienta indispensable para el monitoreo de humedad de tierra.

Este sensor utiliza las dos sondas para pasar corriente a través del suelo, y luego lee la resistencia para obtener el nivel de humedad. Más agua hace que el suelo conduzca la electricidad con mayor facilidad (menos resistencia), mientras que el suelo seco es un mal conductor de la electricidad (más resistencia).

Características

- Sensibilidad ajustable ajustando el potenciómetro digital (en azul).
- Voltaje de operación: 3.3V ~ 5V
- Modo de salida dual, salida digital y salida analógica más precisa.
- Agujeros de montaje para una fácil instalación.
- Dimensiones PCB: 30mm * 16mm
- Dimensiones de sonda: 60mm * 30mm
- Indicador de energía. Indicador alimentación (rojo) e indicador de salida de conmutación digital (verde).
- El módulo tiene un amplificador LM393.

II.1.16.3. SENSOR DE DISTANCIA



El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de arranque y medir la anchura del pulso de retorno. De muy pequeño tamaño, el HC-SR04 se destaca por su bajo consumo, gran precisión y bajo precio por lo que está reemplazando a los sensores polaroid en los robots más recientes.

De fácil uso y programación con las placas de Arduino y microcontroladores.

Características

- Dimensiones del circuito: 43 x 20 x 17 mm
- Tensión de alimentación: 5 Vcc
- Frecuencia de trabajo: 40 KHz
- Rango máximo: 4.5 m
- Rango mínimo: 1.7 cm
- Duración mínima del pulso de disparo (nivel TTL): 10 μ S.
- Duración del pulso eco de salida (nivel TTL): 100-25000 μ S.
- Tiempo mínimo de espera entre una medida y el inicio de otra 20 mS.

II.1.16.4. ELECTROVALVULA



Válvula solenoide, su principal función es abrir o cerrar el paso de cualquier componente líquido, su voltaje de activación es de 12V y su consumo en corriente es de 200mA.

Características:

- 100% Nuevo y de alta calidad
- Material: Metal + plástico
- color principal: Blanco
- Voltaje: 12V DC
- Dimensiones: 85 * 60 * 42 mm / 3.35" * 2.36" * 1.65"
- Conector: diametro exterior 20 mm / 0.79" , diámetro interior: 14 mm / 0.55"
- Tipo de valvula: Diafragma(Operated by Servo)
- Modo de operacion: Normalmente cerrado
- Temperatura de Fluidos: 1-100 °C
- Presion 1-116psi
- Facil de usar en plantas purificadoras de agua
- Equipos de osmosis
- Sistemas de Riego

II.1.16.5. BOMBA DE AGUA



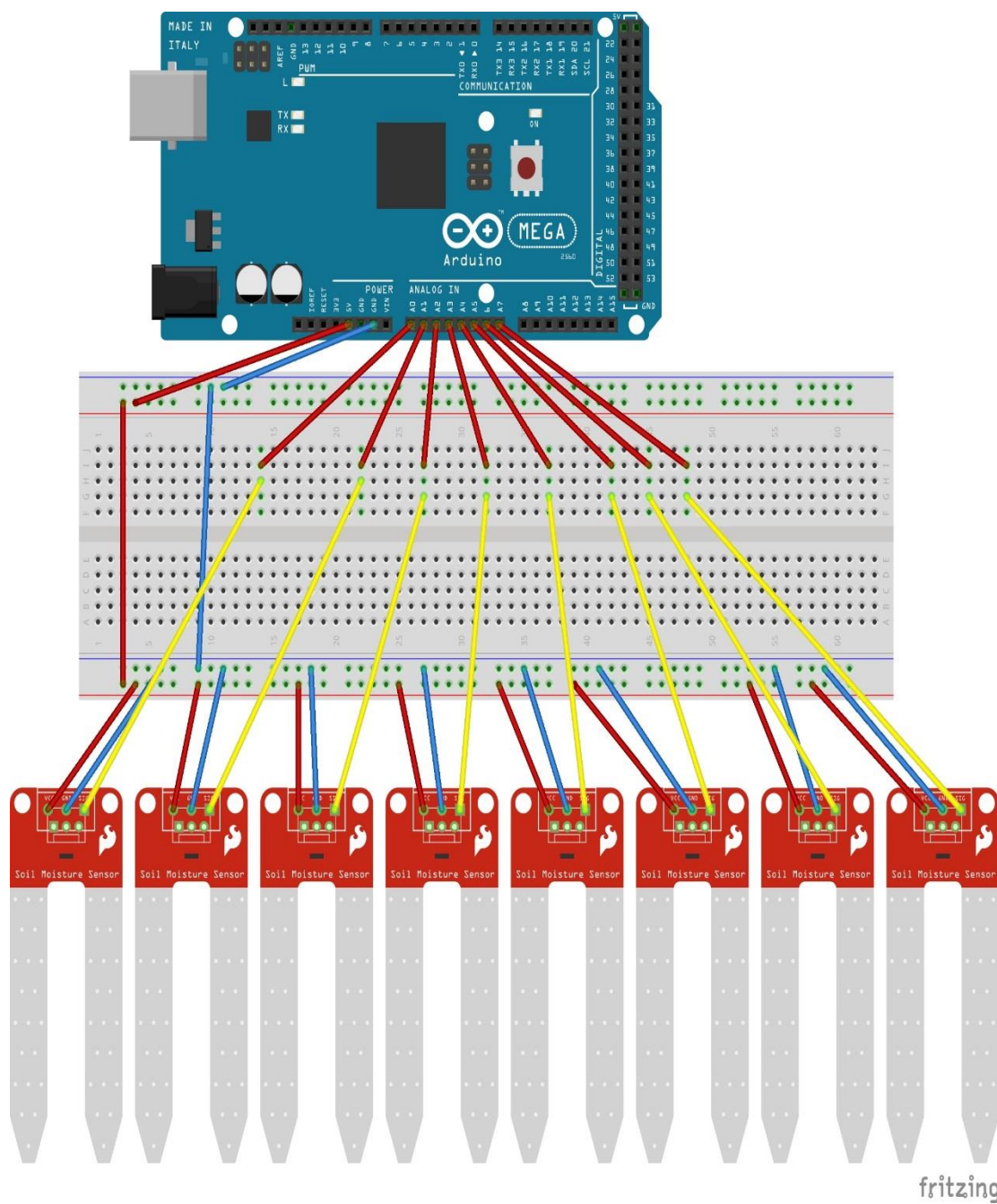
Mini bomba de agua sumergible ideal para sistemas de riego, fuentes de agua artificiales, sistemas de enfriamiento y sistemas de circulación de agua

Características:

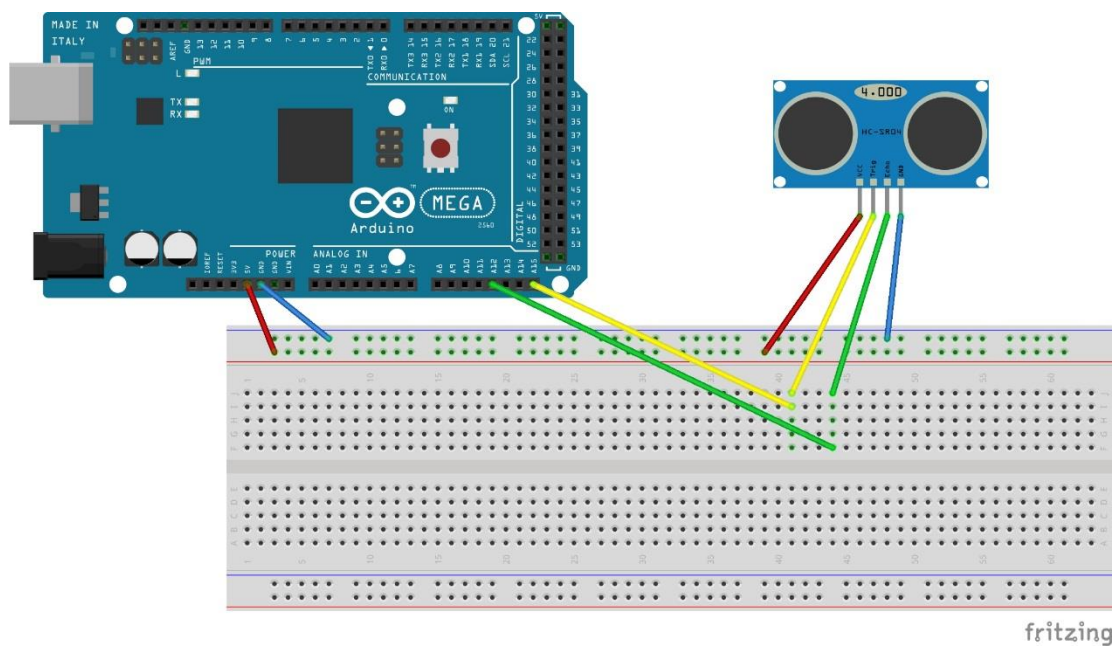
- Voltaje de alimentación: 2.5 a 6 V
- Corriente: 130 a 220 mA
- Flujo 80-120 litros por hora
- Potencia 0.4 - 1.5W
- Altura máxima 110 cm

II.1.17. CONEXIÓN DE LOS COMPONENTES DE ARDUINO

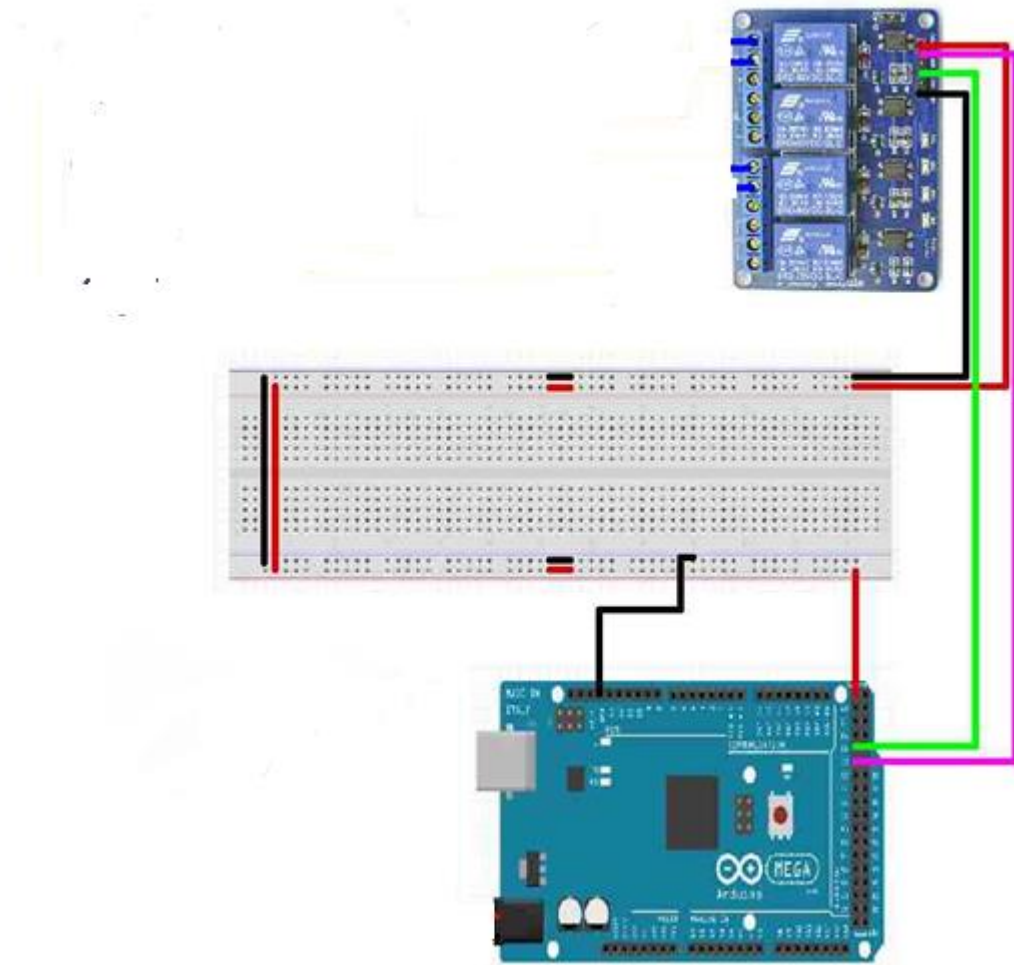
II.1.17.1. Conexión de los Sensores de Humedad de Suelo



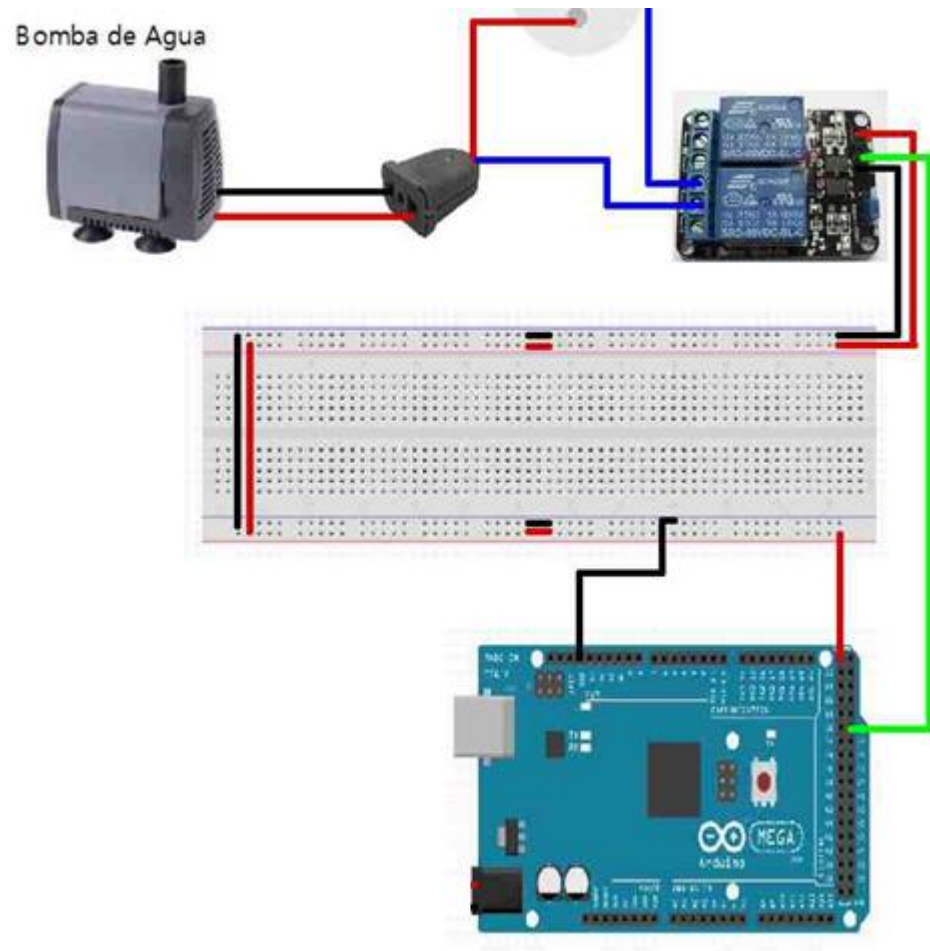
II.1.17.2. Conexión del Sensor de Ultrasonido o Sensor de Distancia



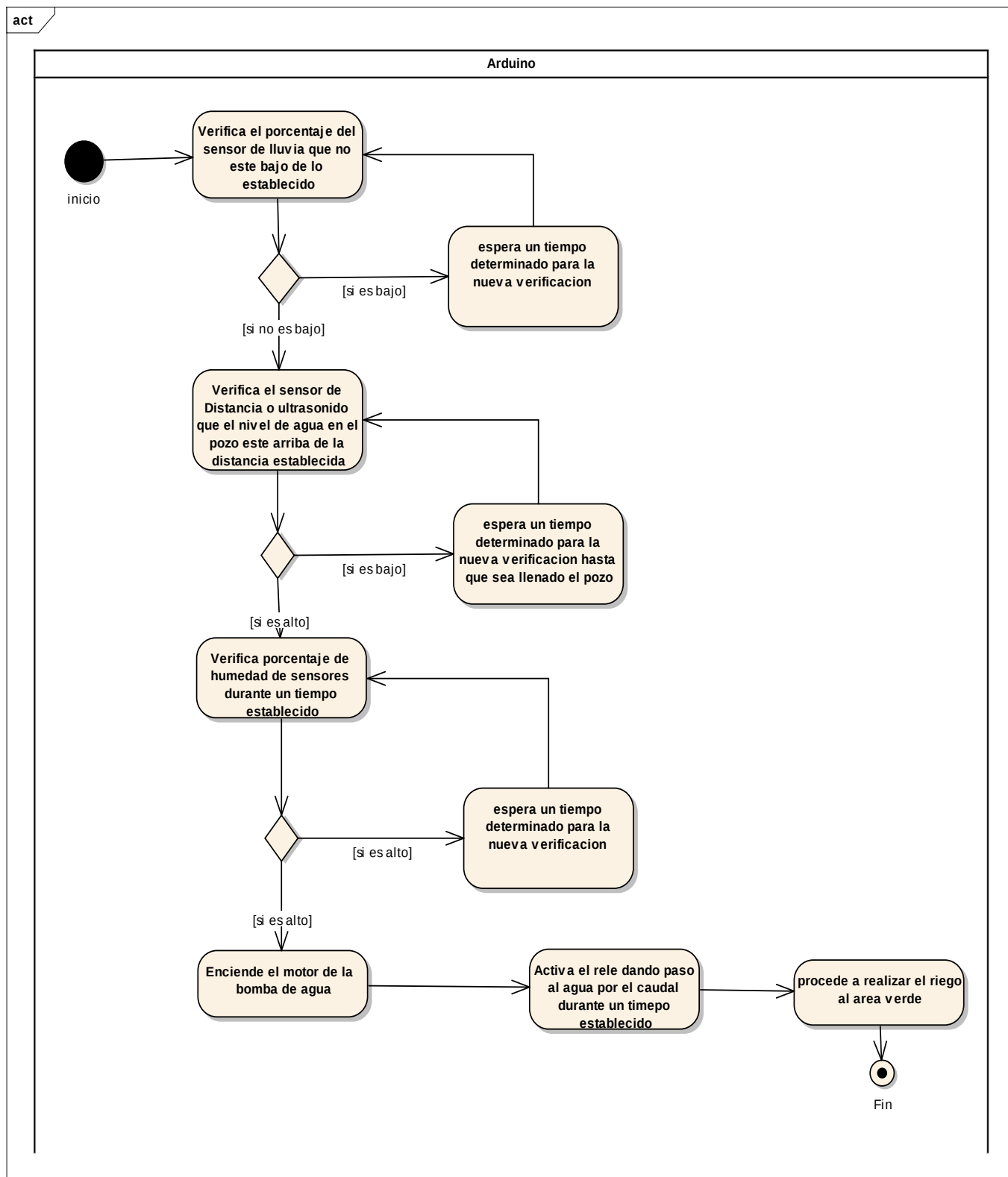
II.1.17.3. Conexión de los Relay's con el Arduino



II.1.17.4. Conexión de La Bomba de Agua con el Arduino



II.1.18. DIAGRAMA DE ACTIVIDAD DEL SOFTWARE DE ARDUINO



II.1.19. ESQUEMA DEL SOFTWARE DE ARDUINO

```

#include <NewPing.h>

#define TRIGGER_PIN 53 // pin del sensor de ultrasonido trigger.
#define ECHO_PIN 52 // pin del sensor de ultrasonido echo.
#define MAX_DISTANCE 25 // distancia maxima

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);

// pines de los sensores
const int sensor1 = A0;
const int sensor2 = A1;
const int sensor3 = A2;
const int sensor4 = A3;
const int sensor5 = A8;

// declarando variables valor de sensores
int valorSensor1 = 0;
int valorSensor2 = 0;
int valorSensor3 = 0;
int valorSensor4 = 0;
int SensorLluvia = 0;
int SensorDistancia = 0;

// declarando valores para los relays
const int relay1 = 3;
const int relay2 = 4;
const int relay3 = 5;
const int relay4 = 6;

// valor para la bomba
const int Bomba = 2;
const int ledpozo = 12;

void setup() {
    // declarando los pines de salida
    pinMode(relay1, OUTPUT);

```

```

pinMode(relay2, OUTPUT);
pinMode(relay3, OUTPUT);
pinMode(relay4, OUTPUT);
pinMode(ledpozo, OUTPUT);
// declarando la bomba como salida

pinMode(Bomba, OUTPUT);
Serial.begin(9600);
}
void loop() {
  unsigned int uS = sonar.ping();
  int g=(uS / US_ROUNDTRIP_CM);

  // leyendo el valor de los sensores y convirtiendolos a porcentaje:
  valorSensor1 = map(analogRead(sensor1),0,1023,0,100);
  valorSensor2 = map(analogRead(sensor2),0,1023,0,100);
  valorSensor3 = map(analogRead(sensor3),0,1023,0,100);
  valorSensor4 = map(analogRead(sensor4),0,1023,0,100);

  SensorDistancia=digitalRead(g);
  SensorLluvia = map(analogRead(sensor5),0,1023,0,100);

  /*0-300 SUELO SECO
  300- 700 SUELO HUMEDO
  700- 1000 SUELO MOJADO
  */

  Serial.println(valorSensor1);
  Serial.println(valorSensor2);
  Serial.println(valorSensor3);

```

```

Serial.println(valorSensor4);
Serial.println(SensorDistancia);
//if(SensorLluvia <= 90){
digitalWrite(ledpozo, HIGH);
//delay(180000); // 3 minutos
//}
//else{
//digitalWrite(ledpozo,LOW);
//if(sendis>=25){
//Serial.print(sendis);
//}else{
//CONDIOCIONES PARA EL RIEGO AUTOMATICO
if(valorSensor1<=40){
  digitalWrite(relay1, HIGH);
}
if(valorSensor2<=40){
  digitalWrite(relay2, HIGH);
}
if(valorSensor3<=40){
  digitalWrite(relay3, HIGH);
}
if(valorSensor4<=40){
  digitalWrite(relay4, HIGH);
}
if(valorSensor1<=40 || valorSensor2<=40 || valorSensor3<=40 ||
valorSensor4<=40){
  digitalWrite(Bomba, HIGH);
}
  delay(2000);
// turno de apagar la bomba

```

```

digitalWrite(Bomba, LOW);
// apaga los relays off
digitalWrite(relay1, LOW);
digitalWrite(relay2, LOW);
digitalWrite(relay3, LOW);
digitalWrite(relay4, LOW);

// 5 minutos realiza el proceso
delay(5000);
//COTROL CON EL SISTEMA WEB
if(Serial.available()>0){

int dato=Serial.read();
if(dato=='A'){  EncenderArea1();  }
if(dato=='a'){  ApagarArea1();  }
if(dato=='B'){  EncenderArea2();  }
if(dato=='b'){  ApagarArea2();  }
if(dato=='C'){  EncenderArea3();  }
if(dato=='c'){  ApagarArea3();  }
if(dato=='D'){  EncenderArea4();  }
if(dato=='d'){  ApagarArea4();  }
if(dato=='I'){  ApagarBomba();  }
    }
}

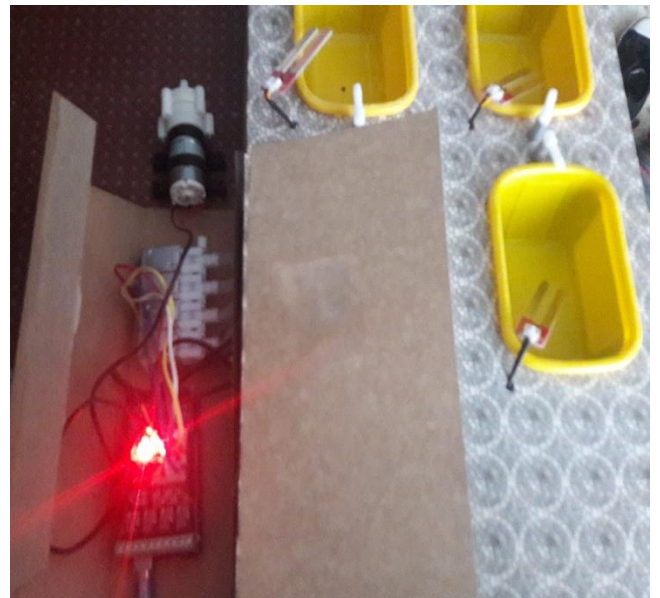
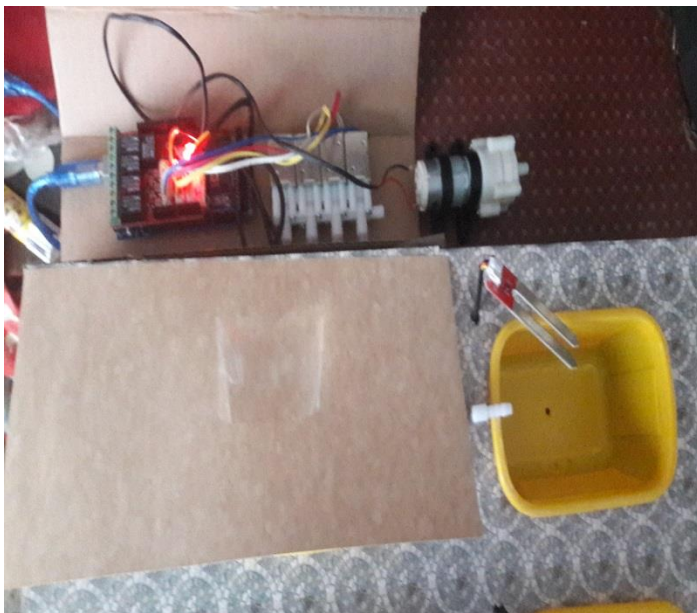
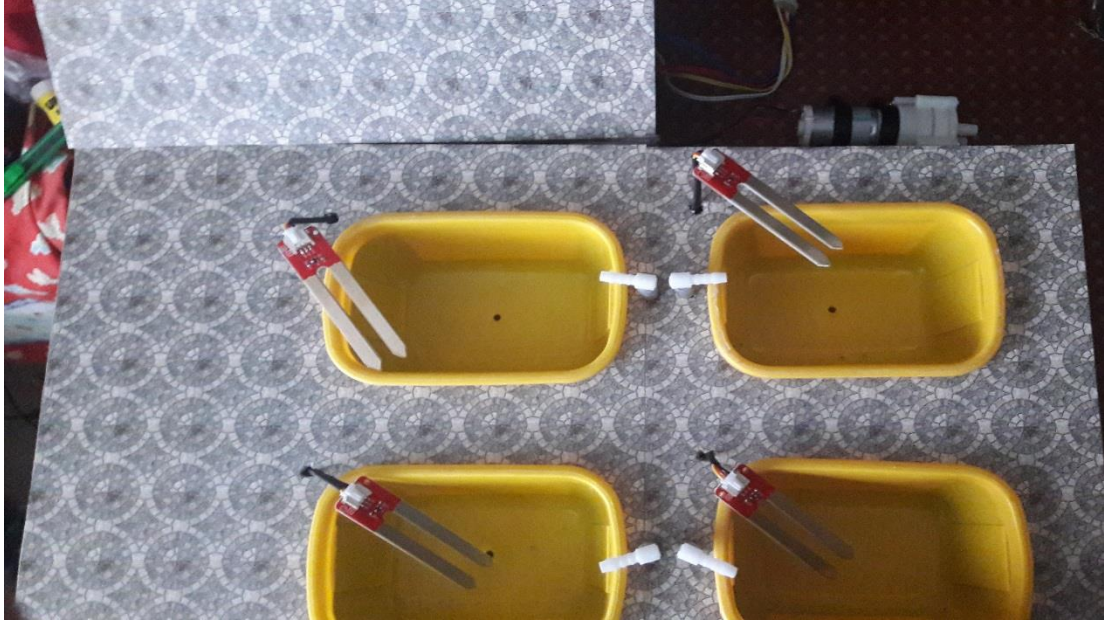
//Funciones para el control del Sistema Web

void EncenderArea1(){
    digitalWrite(Bomba,HIGH);
    digitalWrite(relay1,HIGH); }
void ApagarArea1(){

```

```
    digitalWrite(Bomba,LOW);
    digitalWrite(relay1,LOW);
}
void EncenderArea2(){
    digitalWrite(Bomba,HIGH);
    digitalWrite(relay2,HIGH);
}
void ApagarArea2(){
    digitalWrite(Bomba,LOW);
    digitalWrite(relay2,LOW);
}
void EncenderArea3(){
    digitalWrite(Bomba,HIGH);
    digitalWrite(relay3,HIGH);
}
void ApagarArea3(){
    digitalWrite(Bomba,LOW);
    digitalWrite(relay3,LOW);
}
void EncenderArea4(){
    digitalWrite(Bomba,HIGH);
    digitalWrite(relay4,HIGH);
}
void ApagarArea4(){
    digitalWrite(Bomba,LOW);
    digitalWrite(relay4,LOW);
}
void ApagarBomba(){
    digitalWrite(Bomba,LOW);
}
```

II.1.20. DISEÑO DE LA MAQUETA





II.2. COMPONENTE 2: SOCIALIZACION DEL PROYECTO

II.2.1. CAPACITACION AL PERSONAL ENCARGADO DE LA ADMINISTRACION Y MANTENIMIENTO DEL PROYECTO

II.2.1.1. INTRODUCCION

De acuerdo a las necesidades actuales del mundo laboral y profesional, los términos de Capacitación y Sistemas de Información están modificando la forma de trabajo de las empresas o instituciones; los sistemas de información ayudan a acelerar los procesos, por tanto, las organizaciones que los adoptan en sus funciones logran ventajas competitivas respecto a las demás. Pero si el personal no está debidamente capacitado, están mejoras en la plaza serán poco palpables.

La capacitación se refiere a las metodologías que se usan para proporcionar a las personas dentro de la empresa, las habilidades precisas que necesitan para realizar su trabajo de una manera más eficaz, esto contempla desde pequeños cursos que le permitan al usuario entender el funcionamiento básico del sistema, hasta la capacitación más profunda y avanzada a bases de prácticas y material didáctico entre otros elementos.

II.2.1.1.1. PROPOSITO

Proporcionar al personal de Ornato Público una guía importante en el uso del sistema propuestos, la importancia de su utilización y el uso correcto del sistema.

II.2.1.1.2. OBJETIVOS

- ✓ Capacitar a los administradores del sistema de gestión en el manejo y las funcionalidades que este ofrece.

II.2.1.2. ALCANCES Y LIMITACIONES

II.2.1.2.1. ALCANCES

La Capacitación se realizará solo a las personas que se encargaran de operar el sistema; está dirigida al Administrador y las personas que cumplen el rol de personal; se realizara una explicación de las funciones del sistema web de control y monitorización.

II.2.1.2.2. LMITACIONES

No se capacitará a todo el personal de la empresa.

II.2.2. PROPUESTA DE CAPACITACION

II.2.2.1.1. METODOS DE CAPACITACION UTILIZADOS

Considerando que el personal cuenta con poco tiempo de disponibilidad y ya cuenta con conocimientos en el manejo de las TICs; en la realización de la capacitación acerca del uso del Sistema Informático, se hará uso de técnicas aplicadas en el sitio de trabajo, más propiamente la técnica conocida como *capacitación en el puesto*. Esta técnica contempla que una persona aprenda una responsabilidad mediante su desempeño real en el Sistema Informático, aplicando esta metodología la capacitación puede realizarse en instalaciones de la empresa.

II.2.2.1.2. PLAN DE CAPACITACION

El contenido de la capacitación estará dividido en dos secciones que facilitaran el aprendizaje y consulta del material:

Sección I: Área de Software.

Esta parte está orientada al gerente propietario de la empresa, se profundizará en las utilidades que se brinda respecto al sistema de gestión.

Sección II: Área Hardware.

En esta última sección al igual que la anterior se explica paso a paso las opciones a las que puede realizar el personal con respecto al sistema con sus componentes.

II.2.2.1.3. MATERIAL DE CAPACITACION

Los materiales didácticos que se utilizaran son los siguientes:

- Manual de Usuario
- Computadoras Personales, Presentaciones
- Maqueta prototipo

II.2.2.1.4. DESARROLLO DE CAPACITACIÓN

El taller de Capacitación del Proyecto utilizando un sistema web y la plataforma Arduino se basó en el siguiente programa en ejecución:

Fecha	Tema	Tiempo de desarrollo	Preguntas	Tiempo de participación
07/03/2017	Conceptos centrales de lo que es la Robótica e introducción al Perfil del Proyecto	De 15:00 pm a 15:30pm	10 minutos	
08/03/2017	Dar a conocer el árbol de problemas que se plantea solucionar.	De 17:50 pm a 18:10pm	10 minutos	
08/03/2017	Retroalimentación de los objetivos específicos, alcances, descripción y fundamentación del proyecto.	De 18:00 pm a 18:45pm	10 minutos	
08/03/2017	Refrigerio	De 18:45 pm a 19:00 pm		
08/03/2017	Demostración del uso del Sistema tanto hardware y software desde el servidor web usando el ordenador.	De 19:00pm a 19:20 pm	15 minutos	
	Demostración del uso del Sistema en una maqueta.	De 19:20pm a 19:30	10min	
			TOTAL	4:30 Hrs.

En la primera parte se desarrolló una serie de conceptos básicos que permitan entender los temas centrales del proyecto como el control de Riego, actuadores y sensores de del Sistema.

En la segunda parte se desarrolló la presentación del perfil del proyecto realizando la socialización del árbol de problemas y posteriormente la retroalimentación de los objetivos especificación, gracias al árbol de objetivos existente en el perfil del proyecto, los alcances, la descripción y fundamentación del proyecto

Finalmente, en la tercera parte se comenzó a demostrar el sistema utilizando el servidor web a todos los participantes de la capacitación.

Nombre de la organización donde se realizó la replica	Total de número de alumnos asistentes	N° de hombres	N° de mujeres
	10	9	1

Cabe resaltar que la participación de los asistentes fue muy activa ya que se notó un interés positivo hacia el tema del proyecto que se estaba socializando con un importante rol de preguntas y cuestionamientos además de cierta curiosidad de parte de los asistentes por conocer más acerca de la utilización del sistema.

II.2.3. ANEXOS SOBRE LA CAPACITACION

II.2.3.1.1.1. PRIMERA CAPACITACION





II.2.3.1.1.2. SEGUNDA CAPACITACION AL ADMINISTRADOR DEL SISTEMA



III.1. CONCLUSIONES Y RECOMENDACIONES

III.1.1.CONCLUSIONES

- ✓ Se logró terminar con el prototipo para el proyecto “Control de riego automático para la plaza “Panamericano” a través de un Sistema de Gestión y la Plataforma Arduino”.
- ✓ Se utilizó la norma IEEE 830 para la determinación de requerimientos de la organización que llegó a contribuir a desarrollar un software de calidad.
- ✓ Se aplicó panama_hitek para realizar la conexión arduino_java.
- ✓ Se logró estructurar sistema web mediante el Framework Spring.
- ✓ Se logró validar los datos del sistema realizando pruebas de Caja Negra.
- ✓ Se logró la capacitación al sector interesado en el proyecto de Riego Automático haciendo el uso del manual de usuarios, manual de instalación y una exposición demostrativa y práctica.
- ✓ No se logró concluir con el proyecto Riego Automatizado para la plaza “San José” por falta de equipamiento multifuncional de la universidad.

III.1.2. RECOMENDACIONES

- ✓ Utilizar el sistema desarrollado según se indica en los manuales de usuario del sistema documentado por el director del presente proyecto.
- ✓ Se recomienda ampliar los conocimientos en el uso de dispositivos Arduino de manera que se aproveche todas las ventajas y comodidades que presenta el Sistema.
- ✓ Es necesario que el administrador del sistema este correctamente capacitado para la utilización de este Sistema.
- ✓ Es recomendable utilizar este sistema solo para una sola plaza ya que es realizada solo para ella tomando en cuenta la capacidad de sensores y actuadores soportados por la placa Arduino especificados en este documento.
- ✓ Es necesario tener las especificaciones de los componentes y materiales reales a utilizar en la plaza.