



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
E INFORMÁTICOS**

TEMA:

**APLICACIÓN MÓVIL PARA CONSULTA DE TURNOS DE REGADÍO DEL
CANAL DE RIEGO PÍLLARO RAMAL NORTE YATCHIL CENTRAL.**

Trabajo de Graduación. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

SUBLÍNEA DE INVESTIGACIÓN:

Aplicaciones para dispositivos móviles

AUTOR: Edgar Hipólito Quishpe Collaguazo
TUTOR: Ing. Clay Fernando Aldás Flores, Mg

Ambato - Ecuador
Mayo, 2017

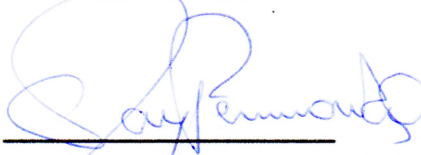
APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

“APLICACIÓN MÓVIL PARA CONSULTA DE TURNOS DE REGADÍO DEL CANAL DE RIEGO PÍLLARO RAMAL NORTE YATCHIL CENTRAL.”, del señor Edgar Hipólito Quishpe Collaguazo, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad Técnica de Ambato

Ambato, Mayo de 2017

EL TUTOR



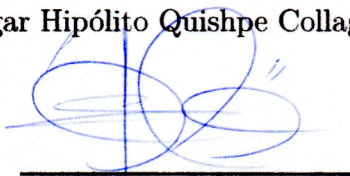
Ing. Clay Fernando Aldás Flores, Mg

AUTORÍA

El presente trabajo de investigación titulado: Aplicación Móvil para la consulta de Turnos de Regadío del Canal de Riego Píllaro Ramal Norte Yatchil Central. Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Mayo de 2017

Edgar Hipólito Quishpe Collaguazo



CC:1718757733

APROBACIÓN COMISIÓN CALIFICADORES

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Franklin Mayorga e Ing. Carlos Núñez, revisó y aprobó el Informe Final del trabajo de graduación titulado “Aplicación Móvil para la consulta de Turnos de Regadío del Canal de Riego Píllaro Ramal Norte Yatchil Central”, presentado por el señor Edgar Hipólito Quishpe Collaguazo de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Elsa Pilar Urrutia Urrutia, Mg



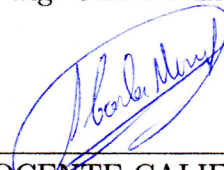
PRESIDENTE DEL TRIBUNAL

Ing. Franklin Mayorga, Mg



DOCENTE CALIFICADOR

Ing. Carlos Núñez, Mg



DOCENTE CALIFICADOR

DEDICATORIA

A Dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad, amor y paciencia que ha tenido conmigo durante todo el transcurso de mi vida.

A mi madre Leticia por haber creído en mi y apoyarme en todo momento, por sus consejos, sus valores, por sus tantas horas dedicadas a pedir a Dios por mi futuro, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre Hipólito por los ejemplos de perseverancia y constancia que lo caracterizan y que ha infundado siempre en toda nuestra familia, por el valor mostrado para salir adelante y el no saber rendir ante cualquier circunstancia difícil. por el enorme sacrificio de alejarse de la familia con el fin de obtener recursos para invertir en mi futuro.

A mi esposa Narciza y mi hijo Mick por ser el motivo para iniciar mis estudios y estar en todo momento brindándome cariño y comprensión, despreocupándose de los problemas del mundo a fin de brindarme espacio para no desconcentrarme de mi educación.

A toda mi familia muchos de los cuales ya no están presente pero siempre incentivaron a que pueda lograr mis sueños de convertirme en un profesional y una mejor persona para así poder ofrecer una vida mejor a los míos.

A todos mis amigos de infancia, su presencia ha sido constante y preocupado sobre el avance de mi carrera profesional.

A Myrian Díaz por ser un gran apoyo y a la vez un gran reto para nunca dejar de estudiar, por enseñar que cuando uno se propone objetivos en la vida, no existe un pretexto para no cumplirlo.

Edgar Hipólito Quishpe Collaguazo.

AGRADECIMIENTO

Agradezco a Dios por ser quien ha propuesto convertirme en la persona que soy, ha bendecido con salud y sabiduría a mis queridos padres, maestros, familia y demás personas que han estado formando parte directa o indirectamente en el desarrollo de mi vida profesional.

A las autoridades de la Universidad Técnica de Ambato, en especial a las autoridades de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, el compromiso y la entrega completa que han invertido desinteresadamente por mejorar continuamente la educación es lo que ha permitido que persista y siga persistiendo la educación de calidad de nuestra querida facultad.

Edgar Hipólito Quishpe Collaguazo

ÍNDICE

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
APROBACIÓN COMISIÓN CALIFICADORA	iv
Dedicatoria	v
Agradecimiento	vi
Introducción	xix
CAPÍTULO 1 El problema	1
1.1 Tema de investigación	1
1.2 Planteamiento del problema	1
1.3 Delimitación	4
1.4 Justificación	4
1.5 Objetivos	5
1.5.1 General	5
1.5.2 Específicos	5
CAPÍTULO 2 Marco teórico	6
2.1 Antecedentes investigativos	6
2.2 Fundamentación teórica	7
2.2.1 El regadío	7
2.2.2 Dispositivos móviles	8
2.2.3 Teléfonos inteligentes (Smartphone)	9
2.2.4 Sistema operativo	10
2.2.5 Android OS	10
2.2.6 Arquitectura de Android	10
2.2.7 Estructura de un proyecto Android	13
2.2.8 Android SDK	14

2.2.9	Dalvik VM	14
2.2.10	Base de datos	14
2.2.11	Motor de base de datos	15
2.2.12	Sistema de gestión de base de datos	15
2.2.13	Web services	15
2.3	Propuesta de solución	16
CAPÍTULO 3 Metodología		18
3.1	Modalidad de la investigación	18
3.1.1	Investigación de campo	18
3.1.2	Investigación Bibliográfica - Documental	18
3.2	Recolección de información	18
3.3	Procesamiento y análisis de datos	19
3.4	Desarrollo del proyecto	19
3.4.1	Metodología	19
3.4.2	Metodología de desarrollo ágil	19
3.4.3	Metodología de desarrollo de aplicaciones móviles	20
3.4.4	Metodología RUP	20
3.4.4.1	Principios de desarrollo	20
3.4.4.2	Ciclo de vida	21
3.4.4.3	Descripción de RUP	23
3.4.4.4	Principales características RUP	23
3.4.4.5	Aspectos de calidad en RUP	27
3.4.5	Metodología SCRUM	27
3.4.5.1	Características de SCRUM	28
3.4.5.2	Roles	29
3.4.5.3	SCRUM de SCRUM	30
3.4.5.4	Reunión de Planificación del Sprint (Sprint Planning Meeting)	31
3.4.5.5	Reunión de Revisión del Sprint (Sprint Review Meeting)	31
3.4.5.6	Retrospectiva del Sprint (Sprint Retrospective)	31
3.4.5.7	Sprint	32
3.4.5.8	Beneficios de SCRUM	32
3.4.5.9	Descripción de SCRUM	32
3.4.5.10	Aspectos de Calidad SCRUM	35
3.4.6	Metodología XP (Extreme Programming)	36
3.4.6.1	Proceso XP	37

3.4.6.2	Fases de desarrollo de la Programación Extrema . . .	38
3.4.7	Comparación	39
3.4.8	Metodología seleccionada	42
CAPÍTULO 4	Desarrollo de la propuesta	43
4.1	Datos informativos	43
4.2	Exploración	44
4.3	Planificación del proyecto	45
4.3.1	Requerimientos	45
4.3.2	Historias de usuario	46
4.3.2.1	Historia de usuario de aplicación móvil	46
4.3.2.2	Historia de usuario de aplicación web	48
4.3.3	Estimación de historias de usuario	50
4.4	Iteraciones	51
4.4.1	Plan de entrega	51
4.4.2	Tareas	52
4.4.2.1	Tareas de aplicación móvil	53
4.4.2.2	Tareas de aplicación web	56
4.4.3	Prototipo	59
4.4.3.1	Diseño de la pantalla de inicio de la aplicación móvil	59
4.4.3.2	Diseño de la pantalla de consulta de la aplicación móvil	60
4.4.3.3	Diseño de la pantalla de inicio de sesión de la aplicación web	60
4.4.3.4	Diseño de la pantalla de administración de la aplicación web	61
4.5	Desarrollo	61
4.5.1	Iteración 1	61
4.5.1.1	Tarea 1 - Adquisición de un dominio y contratación de hosting	61
4.5.1.2	Tarea 2 - Diseño y construcción de la base de datos .	62
4.5.1.3	Tarea 3 - Codificación y desarrollo de interfaz para ingreso a pantalla de consultas de turnos de riego de aplicación móvil	64
4.5.2	Iteración 2	73
4.5.2.1	Tarea 4 - Desarrollo de web service que permita tomar registros desde la base de datos	73
4.5.2.2	Tarea 5 - Desarrollo de la interfaz con los componen- tes necesarios para mostrar los turnos de riego	73

4.5.3	Iteración 3	74
4.5.3.1	Tarea 6 - Usar librería iTextG para generar archivo PDF	74
4.5.4	Iteración 4	78
4.5.4.1	Tarea 7 - Adquirir herramienta “PHP Generator for MySQL” e investigar funcionamiento	78
4.5.4.2	Tarea 8 - Crear pantalla de acceso a la aplicación web	79
4.5.5	Iteración 5	83
4.5.5.1	Tarea 9 - Crear módulos de consulta de turnos de riego para la aplicación web	83
4.5.6	Iteración 6	87
4.5.6.1	Tarea 10 - Crear módulos de inserción y actualiza- ción de usuarios para la aplicación web	87
4.5.7	Iteración 7	87
4.5.7.1	Tarea 11 - Crear módulos de inserción y actualiza- ción y eliminación de turnos de riego para la aplica- ción web	87
4.5.8	Iteración 8	87
4.5.8.1	Tarea 12 - Crear un trigger de base de datos que genere los turnos de riego de manera automática . . .	87
4.5.8.2	Tarea 13 - Crear módulos de inserción y eliminación de generación de calendarios	95
4.6	Pruebas	95
4.7	Producción	97
CAPÍTULO 5 Conclusiones y Recomendaciones		99
5.1	CONCLUSIONES	99
5.2	RECOMENDACIONES	100
Bibliografía		101
ANEXOS		106

ÍNDICE DE TABLAS

3.1	Comparativa entre metodologías RUP y XP N° 2	40
3.2	Comparativa entre metodologías RUP y XP N° 1	41
4.1	Software de desarrollo	46
4.2	Servicios	46
4.3	Historia de usuario 1 - Mostrar pantalla inicial de ingreso al sistema .	47
4.4	Historia de usuario 2 - Consultar turnos en la aplicación móvil	47
4.5	Historia de usuario 3 - Generar archivo PDF	48
4.6	Historia de usuario 4 - Acceder a la aplicación web	48
4.7	Historia de usuario 5 - Consultar turnos en la aplicación web	49
4.8	Historia de usuario 6 - Insertar y actualizar usuarios	49
4.9	Historia de usuario 7 - Insertar, actualizar y eliminar turnos	50
4.10	Historia de usuario 8 - Generar calendarios	50
4.11	Puntos de estimación	51
4.12	Avance de iteraciones y plan de entrega	52
4.13	Tarea 1 - Compra de un dominio y contratación de hosting	53
4.14	Tarea 2 - Diseño y construcción de la base de datos	53
4.15	Tarea 3 - Codificación y desarrollo de interfaz para ingreso a pantalla de consultas de turnos de riego de aplicación móvil	54
4.16	Tarea 4 - Desarrollo de web service que permita tomar registros desde la base de datos	54
4.17	Tarea 5 - Desarrollo de la interfaz con los componentes necesarios para mostrar los turnos de riego	55
4.18	Tarea 6 - Usar librería iTextG para generar archivo PDF	55
4.19	Tarea 7 - Adquirir herramienta “PHP Generator for MySQL” e investigar funcionamiento	56
4.20	Tarea 8 - Crear pantalla de acceso a la aplicación web	56
4.21	Tarea 9 - Crear módulos de consulta de turnos de riego para la aplicación web	57
4.22	Tarea 10 - Crear módulos de inserción y actualización de usuarios para la aplicación web	57

4.23	Tarea 11 - Crear módulos de inserción y actualización y eliminación de turnos de riego para la aplicación web	58
4.24	Tarea 12 - Crear un trigger de base de datos que genere los turnos de riego de manera automática	58
4.25	Tarea 13 - Crear módulos de inserción y eliminación de generación de calendarios	59
4.26	Prueba de aceptación - Consultar turnos en la aplicación móvil	97

ÍNDICE DE FIGURAS

2.1	Sistema de riego	8
2.2	Arquitectura Android	11
2.3	Estructura de un proyecto Android	13
2.4	Modelo básico de un web service	16
3.1	Ciclo de vida metodología RUP	22
3.2	Diagrama SCRUM	29
3.3	Diagrama SCRUM	33
3.4	Flujo de características de la Programación Extrema	37
4.1	Compuerta del canal	44
4.2	Diseño de la pantalla de inicio de la aplicación móvil	59
4.3	Diseño de la pantalla de consulta de la aplicación móvil	60
4.4	Diseño de la pantalla de inicio de sesión de la aplicación web	60
4.5	Diseño de la pantalla de administración de la aplicación web	61
4.6	Detalle del dominio	62
4.7	Diseño de la base de datos	63
4.8	Pantalla de inicio de la aplicación móvil	64
4.9	Ciclo de vida clase AsyncTask	65
4.10	Pantalla de consulta de la aplicación móvil	74
4.11	Inicio de sesión de la aplicación web	83
4.12	Conexión a la base de datos	83
4.13	Configurar fuente de datos	84
4.14	Configurar comportamiento de los campos	84
4.15	Configurar apariencia de la página	86
4.16	Configurar seguridad	86
4.17	Algoritmo de calendarización	88
4.18	Pantalla para generar calendarios	95
4.19	Prueba iteración N° 2	96
A.1	Laguna de Pisayambo	107
A.2	Compuerta principal	108

A.3	Canal de riego	108
A.4	Tanque de reserva y acequia secundaria	108
B.1	Descargar archivo	110
B.2	Activar fuentes desconocidos	111
B.3	Instalar	111
B.4	Abrir aplicación	112
B.5	La aplicación	112
B.6	Selección de tomas y módulos	113
B.7	Menú	113
B.8	Lista de turnos	114
B.9	Archivo PDF	114
C.1	Página principal	115
C.2	Página de inicio de sesión	115
C.3	Turnos	116
C.4	Visualizar turnos	116
C.5	Visualizar turnos	117
C.6	Visualizar turnos	117
C.7	Visualizar turnos	118
C.8	Resultado	118

Resumen

Esta tesis tiene como objetivo desarrollar una aplicación móvil que muestre la lista de los turnos de riego del canal de riego de Yatchil Central, de igual forma una aplicación web que permita el manejo de la base de datos que se creará para albergar la información requerida por nuestro algoritmo que generará la calendarización de los turnos de riego de manera automática.

Uno de los sistemas operativos que mas a sobresalido para dispositivos móviles es Android OS desarrollado por Google, siendo un sistema operativo bien difundido, los desarrolladores se han dedicado al desarrollo de aplicaciones que ejecuten sobre esta plataforma. Con este análisis se ha realizado una aplicación de consulta de turnos de riego del canal de riego de Yatchil Central, aplicación que resulta beneficiosa para mas de 450 usuarios que evitarán formar largas colas y pagar un costo de impresión para obtener el turno correspondiente de regadío. Mediante esta aplicación el usuario accederá rápidamente a la información detallada sobre la fecha y hora que le corresponde hacer uso del agua, podrá generar un archivo PDF y guardarlo en su dispositivo para imprimirlo en algún momento si fuera el caso.

Con el afán de facilitar el trabajo a los administradores del canal de riego en lo que se refiere al mantenimiento de la base de datos, se ha realizado una aplicación web con el que a más de realizar consultas, se podrá insertar, actualizar y eliminar registros de las tablas asociadas al proceso de calendarización de los turnos de regadío. Con esto se da la comodidad al administrador de acceder a la base de datos desde cualquier navegador y realizar las modificaciones necesarias para la correcta distribución del agua.

Tanto la consulta y el mantenimiento a través de la aplicación móvil y la aplicación web respectivamente, se lo realiza a una base de datos que contiene todo el registro y procesos necesarios para generar turnos de riego automáticamente. Mediante un trigger de base de datos, se ha solucionado el trabajo repetitivo y complejo de generar turnos de riego que hasta hace poco los administradores lo realizaban manualmente y les tomaba casi una semana para concluir la elaboración de los cinco calendarios que corresponde a todas las tomas de riego con las que cuenta el canal, con este proceso automático al administrador le tomará pocos minutos generar todos los calendarios para todas las tomas.

Abstract

This thesis takes as aim to develop a mobile application that shows the list of the shifts of irrigation of the channel of irrigation of Yatchil Central, in the same way a web application that allows the handling of the database that will be created to lodge information needed by our algorithm that will generate the scheduling of the shifts of irrigation of an automatic way.

One of the operating systems that more has stood out for mobile devices is Android OS developed by Google, being a quite widely used operating system, the developers have devoted themselves to the development of applications that they execute on this platform. With this analysis has been realized an application of consultation of shifts of irrigation of the channel of irrigation of Yatchil Central, application that is beneficial for more than 450 users who is beneficial and to pay an impression cost to obtain the corresponding irrigation shift. By means of this application the user will gain access quickly to the information detailed on the date and hour that is up to him to make use of the water, he will be able to generate a file PDF and keep it in its device to print it in some moment if it was the case.

In an effort to facilitate the work to the managers of the channel of irrigation as for maintenance of the database, a web application has been made with which, in addition to realize consultations, it will be possible to insert, to update and to eliminate records of the tables associated with the scheduling process of irrigation shifts. On this the serviceability one meets to the manager of gaining access to the database from any navigator and realizing modifications necessary for the correct distribution of the water.

So much the consultation and the maintenance through the mobile application and the web application respectively, realizes it to a database that contains the whole record and necessary processes to generate irrigation shifts automatically. By means of a database trigger, the repetitive and complex work of generating irrigation shifts has been solved, until recently the managers did it manually and it took them almost a week to complete the elaboration of the five calendars that corresponds to all the captures of irrigation with which the channel is provided, with this automatic process to the manager it will take a few minutes to generate all the calendars for all the captures.

Glosario de términos

Algoritmo

Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

API

La interfaz de programación de aplicaciones (API) es el conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

IDE

Un entorno de desarrollo integrado, en inglés Integrated Development Environment, es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Kernel

En informática, un núcleo o kernel es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado (conocido también como modo núcleo). Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

Librerías

Es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

Petición

Es un requerimiento o solicitud que le hace un cliente a un servidor.

Sesión

Es la duración de una conexión hecha desde el cliente hacia el servidor.

SDK

Un Kit de Desarrollo de Software o SDK (siglas en inglés de Software Development Kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de software crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etcétera.

Trigger (base de datos)

Son objetos que se asocian con tablas y se almacenan en la base de datos. Su nombre se deriva por el comportamiento que presentan en su funcionamiento, ya que se ejecutan cuando sucede algún evento sobre las tablas a las que se encuentra asociado. Los eventos que hacen que se ejecute un trigger son las operaciones de inserción (INSERT), borrado (DELETE) o actualización (UPDATE), ya que modifican los datos de una tabla.

INTRODUCCIÓN

El desarrollo del presente proyecto denominado “APLICACIÓN MÓVIL PARA CONSULTA DE TURNOS DE REGADÍO DEL CANAL DE RIEGO Píllaro RAMAL NORTE Yatchil CENTRAL” consta de cinco capítulos divididos sistemáticamente de la siguiente forma:

CAPÍTULO I “EL PROBLEMA”, identifica el problema que actualmente tienen los moradores de Yatchil Central al querer hacer uso del agua para regar en los sembríos. De igual forma aquí se detalla la justificación y los objetivos que se van a cumplir en la presente investigación.

CAPÍTULO II “MARCO TEÓRICO”, realiza una búsqueda de investigaciones previas que sirvan de soporte para el desarrollo del proyecto.

CAPÍTULO III “METODOLOGÍA”, determina el tipo de investigación que ha sido desarrollada, el procesos que ha encaminado a la recolección de la información y su respectivo análisis, las etapas de desarrollo disciplinado por el marco metodológico.

CAPÍTULO IV “DESARROLLO DE LA PROPUESTA”, muestra el trabajo extenso realizado para completar el desarrollo de la aplicación móvil, aplicación web, base de datos y todo los procesos y recursos que han sido necesario crear para la puesta en marcha del proyecto.

CAPÍTULO V “CONCLUSIONES Y RECOMENDACIONES”, expone de forma clara y concisa las consideraciones mas relevantes que se han obtenido el investigador al finalizar el proyecto, además se indican recomendaciones que servirán de apoyo para el desarrollo de futuras investigaciones relacionadas.

CAPÍTULO 1

El problema

1.1. Tema de investigación

Aplicación móvil para consulta de turnos de regadío del canal de riego Píllaro ramal norte Yatchil Central.

1.2. Planteamiento del problema

Los últimos años han sido muy significativos en cuanto a la utilización de aplicaciones móviles se refiere. Con el avance de la tecnología y la accesibilidad que tiene casi toda persona para manipular un dispositivo móvil, muchos sistemas informáticos han dado un salto hacia las plataformas móviles que mediante una aplicación permiten hacer las mismas operaciones que se puede realizar en un computador de escritorio pero con la ventaja de que el lugar no es un limitante de su utilización.

Al rededor del mundo, en países desarrollados, los dispositivos móviles de gama alta son muy comunes en un hogar económicamente promedio. Estos dispositivos permiten realizar tareas mucho más complejas en tiempos más cortos, esto hace que las personas dependan de su dispositivo para realizar las principales actividades bancarias, servicios básicos, etc.

Un estudio encargado por la tecnológica unisys a la consultora idg connect, que ha medido el nivel de implantación en el ámbito empresarial de los programas y estrategias destinados a fomentar el uso de las tecnologías móviles entre los empleados y a llegar a los clientes de una forma más eficaz, concluye que los dispositivos móviles son más productivos y generan mayores entradas en las empresas creando nuevos canales con sus clientes, mejorando su interactividad, incluso aumentando su catálogo de productos y servicios. “El 84 % de las empresas que aplican la tecnología móvil afirman haber aumentado la productividad en el último año gracias

a las políticas de movilidad, mientras que el 75 % asegura también haber generado más ingresos. Otro aspecto analizado en el estudio ha sido el retorno de la inversión (ROI). Según los datos obtenidos, el 79 % de las empresas encuestadas declaran haber podido medir el impacto de las estrategias móviles de forma positiva, y el 65 % incluso afirma contar con un proceso formal y tecnológico destinado a este fin.”[1]

Ecuador trata de avanzar a la par de la tecnología en el mundo, aun cuando conseguir un terminal de última tecnología por medio de una operadora de telefonía móvil no está al alcance de el bolsillo de todo Ecuatoriano, por medio de comercios electrónicos como “mercado libre” y “OLX”, es posible conseguir terminales de gama alta a precios razonables ya que estos evaden los pagos de aranceles y otros impuestos que obliga la ley, de esta forma podemos sacar mucho provecho a las variadas apps existentes en las tiendas en línea. Una de las aplicaciones importantes que se puede encontrar en el play store de google, es la aplicación Produbanco móvil que permite realizar las principales actividades transaccionales de banca.

Según las últimas estadísticas de INEC (Instituto Nacional de Estadística y Censos) realizado en diciembre de 2013 aplicado a 21.768 hogares en todas las regiones del país, afirman que el 16,9 % de las personas que poseen telefonía celular poseen teléfono inteligente, esto sería más de 1,2 millones de individuos, esto representa un número muy considerable de dispositivos móviles que tienen la capacidad de navegar por Internet y correr aplicaciones que permita realizar consultas a plataformas de servicios ofrecido por entidades municipales u otras organizaciones privadas y/o públicos. [2]

Con el objetivo de acercar los servicios a la mayoría de sus clientes, organizaciones públicas y sobre todo las privadas se suman a proveer ciertos módulos de transacciones para dispositivos móviles. Así podemos encontrar aplicaciones de banca móvil que proporcionan las entidades bancarias más conocidas como es Produbanco Grupo Promerica, Banco de Pichincha, Banco Internacional por citar algunos. Por ejemplo Produbanco Grupo Promerica ofrece una aplicación gratuita con la cual el cliente puede acceder a su cuenta y realizar consulta de saldos, conocer fechas y valores de pago de tarjetas de crédito, consultar movimientos de cuentas y tarjetas de crédito, realizar pago de tarjetas propias (del banco) y de terceros, realizar transferencias a cuentas propias y de terceros, hacer pagos de servicios básicos y recargar tiempo aire.[3]

En Tungurahua el sector publico no se ha preocupado lo suficiente en proveer servicios de consultas básicas de tramites municipales o pagos de servicios a través de aplicaciones móviles, sin embargo la Universidad Técnica de Ambato y la Dirección de Tecnología de Información y Comunicación DITIC, ha creado una aplicación que permite a sus estudiantes revisar su información personal, calificaciones actuales y de períodos anteriores, horario de clases y de matrículas desde su teléfono celular con plataforma Android.[4]

En el sector de Yatchil Central, el uso de smartphones de igual forma ha incrementado en los últimos años, esto gracias a los avances tecnológicos y la competitividad de empresas que han abaratado el costo de los equipos. Un smartphone no es completamente funcional si no cuenta con el acceso a Internet pero gracias a las arriesgadas decisiones de empresas proveedoras de Internet que han anclado sus nodos repetidores en las cercanías del sector, se ha podido proveer de este servicio, siendo así Yatchil Central un lugar que cuenta con todo lo necesario para poder hacer uso de los servicios que puede ofrecer cualquier entidad por medio del Internet.

La administración de regadío del canal de riego Píllaro ramal norte Yatchil central, a través de un proceso laborioso que realiza la directiva trimestralmente, provee de turnos de regadío a cada uno de los usuarios quienes deben hacer largas colas para poder recibir un papel impreso con los turnos correspondientes. Los procesos se llevan a cabo muy tradicionalmente sobre hojas de calculo de excel e incluso a mano alzada sobre hojas cuadriculadas, esto hace muy ineficiente el proceso y muy larga la espera del cliente.

La preocupación de responder con mayor rapidez y eficiencia a los clientes, genera la necesidad de hacer uso de las nuevas herramientas que provee la tecnología. La instalación administrativa cuenta con una amplia infraestructura física y equipamiento tecnológico adecuado para proveer servicios a través de Internet, lamentablemente la directiva del canal de riego no cuentan con el conocimiento adecuado en el manejo de la tecnologías de información, lo que da la oportunidad a que el problema lo resuelva un aspirante a titulación universitaria. La entidad será pionera en emprendimiento del uso de tecnologías web y móvil para satisfacer requerimientos exigentes del cliente en actividades que tiene que ver con el uso adecuado del agua de regadío en zonas rurales.

1.3. Delimitación

Área académica: Software.

Línea de investigación: Desarrollo de software.

Sublínea de investigación: Aplicaciones para dispositivos móviles.

Delimitación espacial: Sector Yatchil Central, Parroquia San Andrés, Canton Píllaro.

Delimitación temporal: Esta investigación tendrá una duración de seis meses a partir de la aprobación del perfil de tesis.

1.4. Justificación

El problema actual que enfrentan los administradores y usuarios del canal de riego de Yatchil Central, es la ausencia de un proceso automático que genere los calendarios de riego, actualmente se realiza el proceso de calendarización de forma manual y toma al rededor de una semana el desarrollarlo. Es evidente que esto a la vez produce problemas por la espera, costes de impresión, imprecisión de turnos, entre otros. El siguiente trabajo de investigación es necesario e importante para resolver este problema optimizando así todo el proceso que permita la entrega de turnos de regadío a tiempo, preciso y sin costo. El usuario haciendo uso de una aplicación móvil, podrá consultar los turnos de riego correspondientes a el, esto le tomará a penas unos minutos y como podrá guardar los datos en el celular, no le hará falta gastar en la impresión. Por otra parte el administrador mediante una aplicación web podrá elaborar los calendarios con el ahorro de casi una semana de su tiempo, ahorro económico y ahorro de esfuerzo humano.

La app se convertirá rápidamente en un recurso novedoso que las demás juntas de riego tomanan como ejemplo para solucionar los mismos problemas que se encuentran en tales administraciones.

El trabajo resultará muy útil ya que beneficiará al rededor de 450 personas que cada tres meses deben concurrir a las oficinas de la junta de administración del canal de riego para obtener sus turnos respectivos. Una vez implantado el sistema, los beneficiarios podrán hacer la consulta de sus turnos desde cualquier lugar haciendo uso de un dispositivo móvil que cuente con salida a Internet.

Gracias a los conocimientos obtenidos, los recursos informáticos para el desarrollo y la disponibilidad de dispositivos móviles inteligentes en casi todos los hogares del

sector Yatchil Central, resulta factible realizar este trabajo que como su principal utilidad será facilitar la obtención de turnos de riego de al rededor de 500 personas.

1.5. Objetivos

1.5.1. General

- Desarrollar una aplicación móvil para la consulta de turnos de regadío del canal de riego Píllaro ramal norte Yatchil Central.

1.5.2. Específicos

- Realizar el levantamiento de información de los procesos relacionados con las actividades de calendarización de los turnos de riego para el desarrollo de la aplicación.
- Desarrollar el algoritmo que permita de forma optima y automática generar la calendarización de los turnos de regadío.
- Diseñar e implementar el módulo de administración y los web services necesarios para el control y mantenimiento del sistema utilizando una plataforma web.
- Implantar una aplicación móvil para la consulta de turnos de regadío del canal de riego Píllaro ramal norte Yatchil central que se ejecute sobre plataforma Android.

CAPÍTULO 2

Marco teórico

2.1. Antecedentes investigativos

En Internet se puede encontrar proyectos sobre aplicaciones móviles parecidos al propuesto, con antecedentes investigativos muy valiosos de los cuales se podrá extraer información que beneficiara al avance del presente trabajo. Entre los trabajos similares que destacan para servir como investigaciones previas destacan los siguientes:

Oscar Efrén Acosta Mayorga en su trabajo Aplicación móvil bajo la plataforma Android para la gestión de calificaciones en la Unidad Educativa "Augusto Nicolás Martínez", utilizo la metodología ágil de desarrollo de software “Extreme Programming” (XP) del cual menciona ser un conjunto de valores, principios y prácticas que permite el rápido análisis, diseño, desarrollo y pruebas, necesarias para un correcto funcionamiento de la aplicación móvil, además permite una interacción y entrega de incrementales del producto hacia los usuarios pudiendo interactuar con cambios que puedan surgir durante el proceso de desarrollo. Utiliza software gratuito que permite crear aplicaciones con altos estándares de calidad, hace uso de Android Studio bajo el lenguaje de programación java para el desarrollo de la app, por otra parte el desarrollo del web service lo hace a través de eclipse kepler el cual proporciona las librerías necesarias para que las aplicaciones clientes puedan acceder por medio de un formato de intercambio de información como json. El web service desarrollado en php ha sido hospedado en un servidor de Internet, en cuanto al manejo de la base de datos, phpmyadmin y sqliteman han sido los apropiados para este proyecto gracias a que son independientes de la plataforma en donde se localizan.[5]

Villarruel Chico Miguel Boberto es su tesis Sistema de gestión para historias clínicas bajo la plataforma Android orientado a los médicos del condominio del hospital Millennium, hace uso de la metodología XP (Extreme Programming) gracias a que

permite desarrollar apps en base a pruebas y errores, además de permitir una programación ordenada y en base a las expectativas del cliente, estableciendo las mejores prácticas de ingeniería de software en el desarrollo de proyectos. La base de datos es gestionada a nivel local a través de el motor de base de datos incorporado de Android (sqlite), de esta forma todo el desarrollo de la app es realizada en Android Studio 1.2.1.1, Android sdk built tools 22.0.1, sdk platform api 21 revisión 2. Gran parte del proyecto se ha dedicado para el estudio del uso de j2me java 2 micro edition, que es una colección certificada de apis para el desarrollo de aplicaciones.[6]

Gamboa Saffa Diego Leonardo en su trabajo de tesis Aplicación móvil para el control de notas de los estudiantes de la Universidad Técnica de Ambato utilizando la plataforma Android, a través de las 6 fases del ciclo de vida Extreme Programming project, desarrollo una aplicación móvil para el control de notas de los estudiantes de la uta, realiza un web service asp.net que ofrece el servicio para consumir datos.[7]

En cuanto a la automatización de turnos de riego, no se encuentra trabajos similares pero se puede encontrar investigaciones previas sobre formas de repartición de turnos de regadío de agua, el riego libre es propio de sistemas de distribución con pocas reglas y normas sobre el uso del agua y permite que los usuarios puedan hacer uso de la cantidad necesaria de agua durante el tiempo necesario, sin embargo en la mayoría de los sistemas de riego en los andes, existe el sistema de distribución según turnos.[8]

2.2. Fundamentación teórica

2.2.1. El regadío

Uno de los mayores consumidores de los recursos hídricos es el agricultor llegando a hacer uso de mas del 70 % del agua del planeta, la necesidad de el uso del agua de regadío depende en mayor parte del clima presente, en verano hace muy necesario que el agricultor aplique regadío para que sus cultivos tengan éxito. En este punto resulta muy complicado repartir el liquido vital a las tantas personas con necesidades comunes, para esto muchos países optan por implementar modelos de repartición de agua, esto hace que sea justo y ordenado el regadío evitando así el desorden y confusión absoluta entre los beneficiarios y personal encargado de la administración.

Los métodos de riego más comunes son por inundación y aspersión, para esto se puede necesitar mucha infraestructura como canales (Ver figura: 2.1), reservorios,

tubería, etc.

[9][10]



Figura 2.1: Sistema de riego
Elaborado por: Edgar Hipólito Quishpe Collaguazo

2.2.2. Dispositivos móviles

Un dispositivo móvil es un equipo electrónico que cuenta con características de procesamiento, capacidades de memoria y conexión permanente diseñado para una función específica pero que puede realizar otros trabajos generales. Este tipo de terminal no solo es un teléfono inteligente sino que se convierte en un computador que podría sin problemas estar en las mismas categorías que los computadores de alto rendimiento como por ejemplo: pc portátiles, tabletas, entre otros. Con todas estas posibilidades hace muy difícil estandarizar las aplicaciones para que funcione en todos los dispositivos y en todas las plataformas. La principal dificultad es que los dispositivos poseen arquitecturas de hardware y software lo que resulta un reto para cualquier desarrollador realizar una aplicación multiplataforma.

Entre las características mas sobresalientes de los dispositivos móviles están:

- Son pequeños.
- Capacidad de procesar información.
- Memoria RAM.
- Conexión permanente.

Existen varios tipos de dispositivos móviles que dependiendo de las necesidades prestan funciones diferentes al usuario:

- Teléfonos móviles.
- Organizadores y asistentes personales digitales (personal digital assistant).
- Web-enabled phones.
- Two-way pagers.
- Smartphones.
- Handheld pc.
- Tablet pc.
- Tablets.
- Libros electrónicos (e-books).[11]

2.2.3. Teléfonos inteligentes (Smartphone)

Un smartphone combina la localización de un GPS con el sistema de mensajería de correo electrónico así también con reconocimientos de voz y una cámara potente o realizar búsqueda en Internet al mismo tiempo que se escucha música. El teléfono inteligente hace que todas las posibilidades de una pc se traslade al bolsillo del usuario y ofrece muchas más prestaciones y en todo momento gracias a su fácil portabilidad.

Posee las principales características de un dispositivo móvil siendo su principal función el permitir comunicación con el mundo a través de Internet o el servicio de telefonía.[12]

Para su funcionamiento a demás de el hardware que poseen, necesitan un sistema operativo que administre los recursos del equipo así como permita interactuar al usuario con las aplicaciones.

2.2.4. Sistema operativo

El sistema operativo es un conjunto de programas informáticos enfocados a la administración de los recursos que ofrece un computador, es el encargado de realizar un vínculo entre los recursos hardware, el usuario y las aplicaciones finales.

Existen varios tipos de sistemas operativos, las clasificaciones son extensas dependiendo de la arquitectura del procesador, por citar un ejemplo el optar por uno u otro dependerá del trabajo al que esta destinado. [13]

2.2.5. Android OS

Android es un sistema operativo basado en el núcleo de linux creado inicialmente por Android Inc. y fue absorbido mas tarde por google en unión con open handset alliance, un consorcio experto en hardware, software y telecomunicaciones.

Gracias a que su plataforma es completamente abierta, su capacidad de adaptación a cualquier dispositivo, aplicaciones desarrolladas en java, pantalla adaptable, alta calidad de gráficos y procesamiento, la filosofía de siempre conectado a Internet, entre otros hace que Android lidere el mercado estando presente en la mayoría de los dispositivos móviles.[14]

2.2.6. Arquitectura de Android

Esta formado de cuatro capas siendo todas libres. (Ver figura: 2.2)

Aquí una breve explicación de lo mas relevante de la arquitectura.

Kernel de linux

El núcleo actúa con las capas de la arquitectura y el hardware del dispositivo, para acceder al núcleo se debe hacerlo mediante el uso de librerías que se dispone en las capas superiores. Cada dispositivo que es añadido al dispositivo móvil, lleva un controlador dentro del kernel lo que permite poder usar el dispositivo mediante el uso de software.

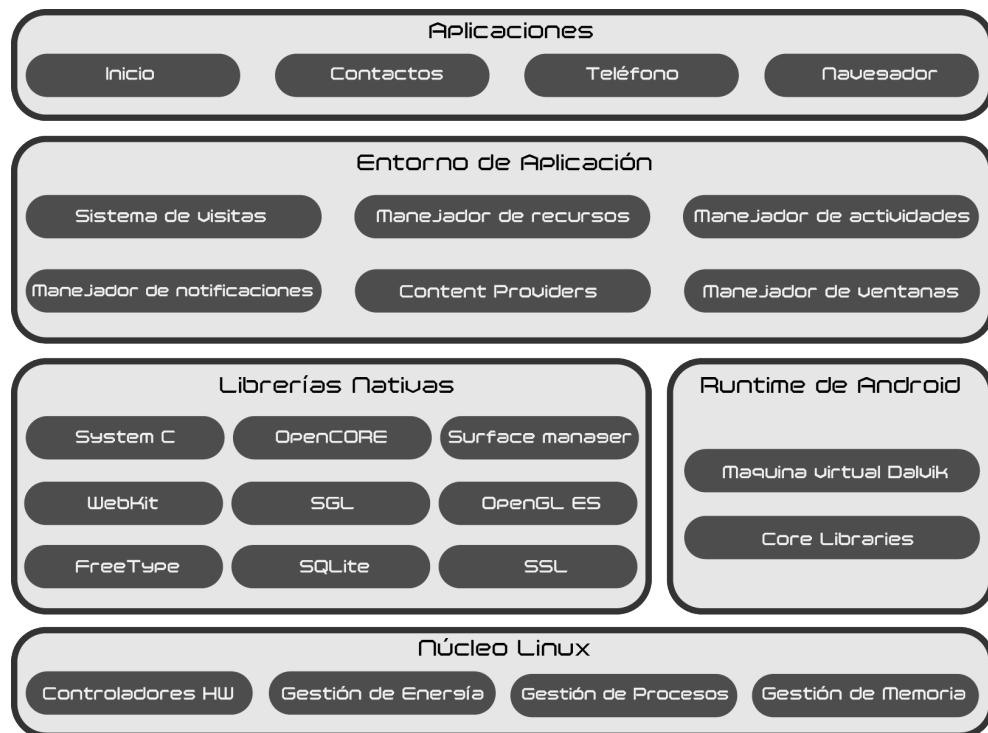


Figura 2.2: Arquitectura Android
 Elaborado por: Edgar Hipólito Quishpe Collaguazo
 Fuente: [14]

Bibliotecas

Estas bibliotecas están escritas en c o c++ y compiladas para la arquitectura hardware específica del teléfono, antes de ser comercializado el dispositivo, el fabricante instala las librerías con el objetivo de evitar codificar cada vez que una aplicación realice tareas que se repiten con frecuencia.

Entre las bibliotecas que se incluye habitualmente en un terminal que va a ser comercializado, tenemos los siguientes:

- **Gestor de Superficies (Surface Manager):** es el encargado de componer las imágenes que se muestra en la pantalla.
- **SGL (Scalable Graphics Library):** es el motor gráfico 2d de Android, se encarga de presentar elementos en dos dimensiones.
- **Opengl (Opengl for Embedded Systems):** es el motor gráfico 3d de Android.
- **Bibliotecas multimedia:** permiten visualizar, reproducir e incluso grabar numerosos formatos de imagen, vídeo y audio.
- **Webkit:** es el motor que utiliza el navegador para acceder a la web.

- **SSL (Secure Sockets Layer):** encargado de proporcionar seguridad cuando el usuario accede al Internet.
- **Freetype:** permite mostrar fuentes tipográficas, tanto basadas en mapas de bits como vectoriales. SQLite: motor de bases de datos relacionales, disponible para todas las aplicaciones.
- **SQLite:** motor de bases de datos relacionales, disponible para todas las aplicaciones.
- **Biblioteca C de sistema (LIBC):** está basada en la implementación de berkeley software distribution (bsd), pero optimizada para sistemas linux embebidos. Proporciona funcionalidad básica para la ejecución de las aplicaciones.[15]

Entorno de ejecución

El componente principal de el entorno de ejecución es la maquina virtual dalvik y dado que esta formado por librerías con las funcionalidades habituales de java y especificas de Android, no es considerada como una capa.

Framework de aplicaciones

Esta formada por servicios y clases que utilizan las aplicaciones de forma directa para realizar las funciones.

Aplicaciones

Aquí se encuentran las aplicaciones propias, las administradas por java y las que instala el usuario en su dispositivo. También está la aplicación principal del sistema: inicio (home), también llamada a veces lanzador (launcher), porque es la que permite ejecutar otras aplicaciones proporcionando la lista de aplicaciones instaladas y mostrando diferentes escritorios donde se pueden colocar accesos directos a aplicaciones o incluso pequeñas aplicaciones incrustadas o widgets, que son también aplicaciones de esta capa.[15]

La capa de aplicaciones es donde va a correr nuestras apps instaladas en la maquina, todos correrán sobre la maquina de dalvik quien se encargara de ofrecer la seguridad necesaria.[14]

2.2.7. Estructura de un proyecto Android

Los IDEs (Entorno de Desarrollo Integrado, en inglés Integrated Development Environment) especializados para el desarrollo de aplicaciones para Android nos ayudan en la creación del espacio de trabajo con contenido listo para compilar. Para el desarrollo del presente trabajo se ha usado el IDE de desarrollo Android Studio.

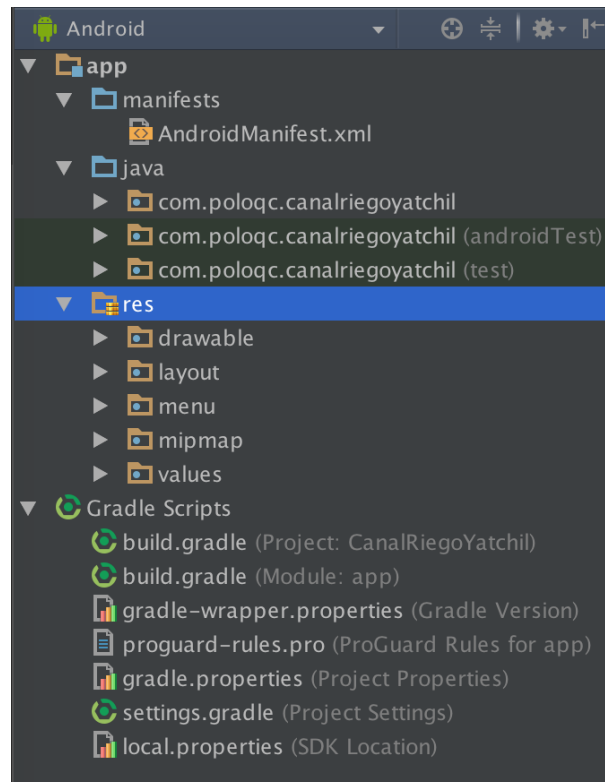


Figura 2.3: Estructura de un proyecto Android
Elaborado por: Edgar Hipólito Quishpe Collaguazo
Fuente: [15]

Se detalla a continuación los componentes mas importantes para entender la estructura de un proyecto Android con el IDE de desarrollo Android Studio.

Activity: son el componente visual de una aplicación, es decir forman la interfaz y se podría decir que son las pantallas o ventanas que ve el usuario.

Service: son componentes sin interfaz gráfica, se ejecutan en segundo plano y realizan múltiples acciones.

Content provider: es la forma que tiene Android de compartir datos entre aplicaciones, compartiendo datos sin la necesidad de dar de detalles sobre su almacenamiento.

Broadcast reciver: este componente se encarga de detectar y reaccionar a los

eventos generales como pueden ser: batería baja, entrada de llamada, es decir es el encargado de recibir las llamadas que son enviados a cualquier aplicación que esté “escuchando”.

Las aplicaciones hechas en Android están basadas en componentes, estos son los elementos fundamentales con el que se construye un proyecto. Una aplicación final resulta de la combinación de uno o más componentes declarados dentro del fichero *Androidmanifest.xml* y es en este fichero donde se definen los componentes y se otorga permisos requeridos por la aplicación así como se define los recursos y librerías que va utilizar la app.[16]

2.2.8. Android SDK

Sdk son las siglas software development kit, esto viene a ser un kit de desarrollo de software para aplicaciones móviles que se ejecuta sobre la plataforma de Android. Con él podremos desarrollar aplicaciones y ejecutar un emulador del sistema Android de la versión que sea. Todas las aplicaciones Android se desarrollan en lenguaje java con este kit.

2.2.9. Dalvik VM

Dalvik es la maquina virtual que utiliza Android, este nombre fue seleccionado por su diseñador y escritor “dan bornstein” en honor a un pueblo de pescadores de eyjafjörður (islandia), donde vivieron algunos de sus antepasados. Dalvik solo interpreta y ejecuta solo los archivos .dex. a este formato se le optimizado principalmente para que ahorre memoria, lo cual consigue delegando en el kernel la gestión de hilos (multithreading), de memoria y de procesos. [17]

2.2.10. Base de datos

Es una forma sistemática, ordenada, estructurada de almacenar datos, incluye todos los mecanismos para poder compartir estos datos con diferentes usuarios en una red, entre sus características principales están:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.

- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.[18]

2.2.11. Motor de base de datos

Aunque no se lo ve instalado en el equipo, el motor es el que manipula directamente la base de datos, el motor es el que maneja lenguajes de programación y procedimientos almacenados, el motor es el que programa eventos, salidas, códigos ejecutables.[18, 19]

2.2.12. Sistema de gestión de base de datos

Compuesta por un lenguaje de definición de datos (ddl) y un lenguaje de manipulación de datos (dml), los sgbd son un tipo de software muy específico para un motor de bd, diseñado para servir de interfaz entre el usuario y la base de datos.[19]

2.2.13. Web services

El web service es el proceso en el que un usuario puede realizar peticiones de servicios creados por terceros o propios que se encuentre en cualquier parte del mundo, haciendo uso de protocolos y estándares universales de Internet. El cliente que consume el servicio no deberá preocuparse de la plataforma en la que va a utilizar ni el lenguaje que va a decodificar el servicio. (Ver figura: 2.4)

Existen dos formas de crear un web wervice:

- SOAP: basados en xml , aprovecha el puerto 80 para salir, crea una capa adicional de servicio.
- REST: no crea una capa adicional de servicio, describe un conjunto de principios de la arquitectura.

Estándares necesarios:

- XML: formato estándar para los datos que se van a intercambiar.
- SOAP: protocolos sobre los que se intercambia los web services.

- WSDL: contiene los requisitos funcionales necesarios para establecer la comunicación con los servicios web. Cuando se usa un web service, se lee un archivo wsdl que contiene los parámetros que tiene el web service.
- UDDI: protocolo para publicar los web service y saber que servicios web están disponibles.[14, 20]

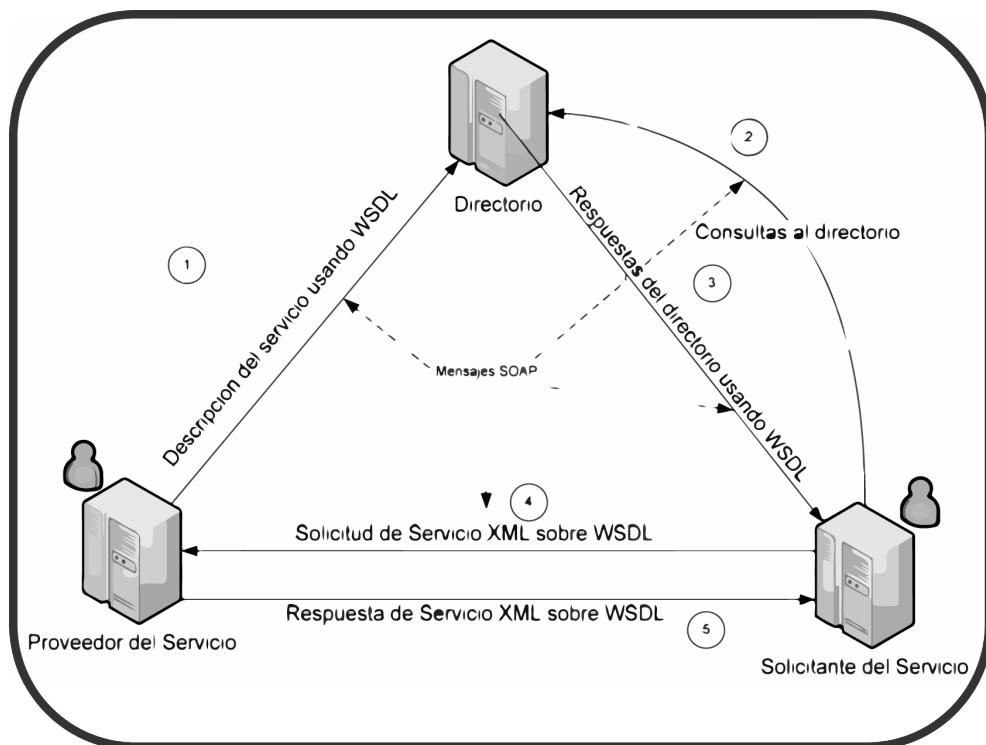


Figura 2.4: Modelo básico de un web service
Elaborado por: Edgar Hipólito Quishpe Collaguazo
Fuente: [20]

2.3. Propuesta de solución

El presente trabajo aportará solución a la comunidad agricultora, específicamente a quienes hacen uso del canal de riego Píllaro ramal norte Yatchil Central.

Se iniciará el proyecto realizando el levantamiento de información y su análisis respectivo para desarrollar una optimizada base de datos, relacionando todas las entidades que involucren el proceso de calendarización de turnos de regadío. Esto resolverá el problema de turnos duplicados y falta de precisión de horas asignadas en función del área registrado de cada cliente.

Para eliminar el trabajo laborioso que la directiva debe realizar trimestralmente ideando una forma de proveer los turnos de regadío, se automatizará el proceso de calendarización mediante un trigger de base de datos. Igualmente se facilitará el trabajo del administrador permitiendo realizar el manejo y mantenimiento de la base de datos a través de una aplicación web. Por medio de este sistema también se podrá realizar consultas de turnos de riego, revisar los días asignados para la limpieza del canal, días de reuniones, entre otros.

Los web services permitirá que la aplicación móvil intercambie información con la base de datos lo que permitirá que el usuario del canal de riego realice consultas de los turnos de regadío a través de una aplicación móvil con conexión a Internet, el resultado de la consulta proporcionará el día, la hora y la cantidad de horas que debe hacer uso del líquido vital. De esta forma los usuarios evitaban las largas colas, el traslado innecesario y el costo de impresión que en la actualidad debe asumir el usuario siendo el proceso llevado de forma manual.

CAPÍTULO 3

Metodología

3.1. Modalidad de la investigación

3.1.1. Investigación de campo

Para el presente trabajo, se realizara una investigación de campo para determinar las divisiones de los terrenos, áreas y propietarios respectivos, acequias secundarias que reparten el agua del canal, tanques de almacenamiento, por citar las mas importantes. En este proceso se contara con el apoyo de la Directiva del Canal de Riego de Yatchil Central y personal encargado de la distribución de los turnos en la actualidad.

3.1.2. Investigación Bibliográfica - Documental

En cuanto al desarrollo de la aplicación, este trabajo conlleva a la realización de una investigación Bibliográfica-documental debido a que se deberá investigar sobre herramientas, métodos y técnicas actuales que garanticen una funcionalidad extendida de la App.

3.2. Recolección de información

La obtención de información válida para el proceso de esta investigación, dependerá del tipo de investigación que conlleva ejecutar este proyecto, así entonces mediante la investigación de campo, se lograra recolectar toda la información relacionada con los componentes y características del Canal de Riego Píllaro Ramal Norte Yatchil Central . La información que se obtenga será de mucha importancia para poder diseñar y codificar funciones o procedimientos que determine la mejor solución a los problemas de calendarización.

La técnica de recolección de información que mayormente permitirá recolectar los datos será la observación con la que se podrá captar de forma sistemática los fenó-

menos relevantes para esta investigación.

Para la investigación Bibliográfica - Documental, se obtendrá información de Internet haciendo uso de motores de búsquedas avanzada como Google Académico, estos hallazgos aportaran soluciones a problemas que se presenten en la etapa de diseño y desarrollo de la Aplicación.

3.3. Procesamiento y análisis de datos

Cuando se disponga de toda la información en las manos, se procederá a formalizar los datos adquiridos eliminando la ambigüedad y redundancia. También se procederá a tabular la información a fin de realizar comparaciones con resultados anteriores que nos permita ver lo necesario que resulta la realización de este trabajo de tesis.

Los datos recolectados sobre áreas de terreno, usuarios, acequias, serán la clave para el diseño y desarrollo de la base de datos donde se ingresara todos estos datos que posteriormente servirán para relacionar en la tabla de calendarios.

3.4. Desarrollo del proyecto

3.4.1. Metodología

Las metodologías de desarrollo de software se refiere a un marco o estructura de guía y soporte que define el desarrollador para crear un software que sea desarrollado con una planeación y control durante todo el proceso de elaboración de un proyecto.

3.4.2. Metodología de desarrollo ágil

Las metodologías ágiles son muy adecuadas principalmente para el desarrollo de aplicaciones móviles:

En primer lugar, la alta volatilidad del entorno hace que constantemente el equipo de desarrollo se deba adaptar a nuevos terminales, cambios en la plataforma o en el entorno de desarrollo. Un ritmo cambiante que requiere una alta respuesta al cambio más que al seguimiento de un plan concreto.

Como se ha indicado, al menos actualmente, los equipos de desarrollo móvil suelen estar integrados por pocas personas, no más de ocho o diez desarrolladores entorno a un misma aplicación o incluso, un único desarrollador. Las interacciones en el proceso

y las herramientas son más controlables y es posible una fluida comunicación entre los miembros del equipo.

Una aplicación se suele realizar en periodos de desarrollo cortos entorno a un mes a seis meses. Con el propósito de una realimentación rápida es posibles realizar varias actualizaciones de un aplicación según se van entregando funcionalidades. Un tema demandados por los usuarios en los distintos markets de aplicaciones que aprecian la frecuente mejora de la aplicación para mantenerla viva, e incluso ayuda a su propio marketing al tener más visibilidad cada vez que se realiza la actualización.

3.4.3. Metodología de desarrollo de aplicaciones móviles

Existen muchas metodologías enfocadas al desarrollo móvil y en vista de que cada metodología tiene sus pros y contras, analizaremos algunas de ellas y compararemos las mas acercadas al desarrollo móvil con el fin de obtener un estructura que nos permita desarrollar nuestra propuesta de forma ágil y garantizada.

El análisis de las siguientes metodologías variantes que se enfocan al desarrollo móvil, nos va ha permitir seleccionar la mejor opción para lograr un trabajo con terminado de alta calidad que responda de manera optima a las exigencias del usuario inexperto y logre ser beneficioso para los cientos de personas que serán las que utilicen la aplicación móvil.

3.4.4. Metodología RUP

Rational Unified Process (RUP). Este es un proceso de desarrollo de software desarrollado y mantenido por Rational Software actualmente propiedad de IBM. Junto al acoplamiento de Unified Modeling Language (UML), forman la mejor metodología para el análisis, diseño implementación y documentación de sistemas orientados a objetos.

3.4.4.1. Principios de desarrollo

El Rational Unified Process proporciona a cada miembro del equipo con las directrices, plantillas y herramientas mentores necesarios para todo el equipo.

El RUP está basado en 6 principios clave que son los siguientes:

- **Adaptar el proceso**

El proceso deberá adaptarse a las necesidades del cliente, es muy importante interactuar con el individuo.

- **Equilibrar prioridades**

Los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un equilibrio que satisfaga los deseos de todos.

- **Demostrar valor iterativamente**

Los proyectos se entregan, aunque sea de modo interno, en cada etapa se analiza la opinión de los inversores, la estabilidad y la calidad del producto y se refina los riesgos y la dirección del proyecto.

- **Colaboración entre equipos**

Como el desarrollo no lo realiza una única persona, debe haber una buena relación entre los miembros del equipo.

- **Elevar el nivel de abstracción**

Motiva al uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación, sin saber con certeza lo que hay que codificar

- **Enfocarse a la calidad**

El control de calidad no se debe realizar al final de cada iteración, sino en cada instante de la producción.[21]

3.4.4.2. Ciclo de vida

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.(Ver figura: 3.1)

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto,

la eliminación de los riesgos críticos, y al establecimiento de una baseline (Línea Base) de la arquitectura.

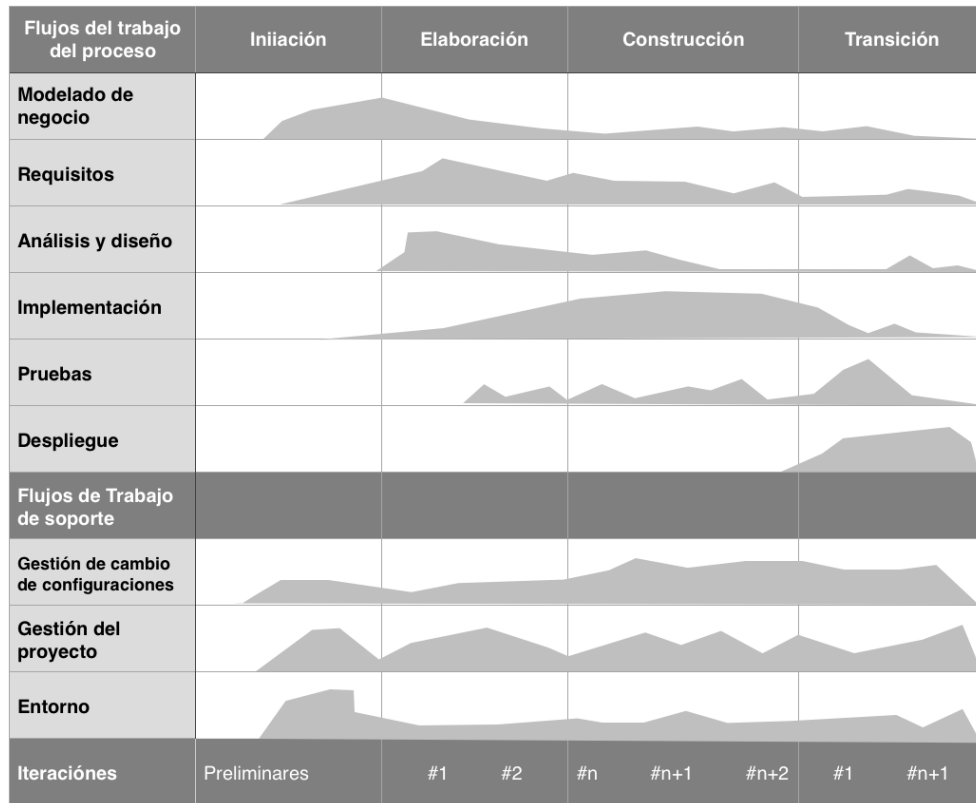


Figura 3.1: Ciclo de vida metodología RUP
Elaborado por: Edgar Hipólito Quishpe Collaguazo
Fuente: [21]

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se seleccionan algunos Casos de Uso, se refinan su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan iteraciones hasta que se termine la implementación de

la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero dependiendo de la fase el esfuerzo dedicado a una disciplina varía.[22]

3.4.4.3. Descripción de RUP

El RUP es un conjunto de metodologías adaptables a las necesidades de cada empresa. Proporciona un enfoque disciplinado para la concesión de tareas y responsabilidades dentro de un equipo de desarrollo, con el objetivo de asegurar la producción de software de calidad que satisfaga al usuario final.

Es un proceso de Ingeniería de Software. Proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo.

Su objetivo es asegurar la producción de alta calidad software que satisfaga las necesidades de sus usuarios finales, dentro de un horario predecible y presupuesto. [23, 24]

3.4.4.4. Principales características RUP

- Forma disciplinada de asignar tareas y responsabilidades.
- Implementa las mejores prácticas en ingeniería de software.
- Implementa el desarrollo Iterativo.
- Administra los requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad de software.

El RUP se caracteriza por ser iterativo e incremental, Incluye productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y

roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP define grupos de roles, agrupados por participación en actividades relacionadas. Estos grupos son:

Analistas:

- Analista de procesos de negocio.
- Diseñador del negocio.
- Analista de sistema.
- Especificador de requisitos.

Desarrolladores:

- Arquitecto de software.
- Diseñador de interfaz de usuario.
- Diseñador de cápsulas.
- Diseñador de base de datos.
- Implementador.
- Integrador.

Gestores:

- Jefe de proyecto.
- Jefe de control de cambios.
- Jefe de configuración.
- Jefe de pruebas.
- Jefe de despliegue.
- Ingeniero de procesos.
- Revisor de gestión del proyecto.
- Gestor de pruebas.

Apoyo:

- Documentador técnico.
- Administrador de sistema.
- Especialista en herramientas.
- Desarrollador de cursos.
- Artista gráfico.

Especialistas en pruebas:

- Especialista en Pruebas (tester).
- Analista de pruebas.
- Diseñador de pruebas

Fases:

- Establece oportunidad y alcance.
- Identifica las entidades externas o actores.
- Identifica los casos de uso.

RUP establece su disciplina por medio de 2 aspectos que son:

Proceso:

- Modelación.
- Requisitos.
- Análisis y Diseño.
- Implementación.
- Pruebas.
- Despliegue.

Soporte:

- Gestión de cambio.
- Gestión de proyecto.

- Entorno.

Para cada una de sus facetas realiza una serie de artefactos que ayudan a comprender de mejor manera el análisis y el diseño.

Artefactos

Inicio:

- Documentación.
- Diagramas de caso de uso.
- Especificación de requisitos.
- Diagrama de requisitos.

Elaboración

Documento que trabaja con las siguientes vistas:

Vista Lógica:

- Diagrama de clases.
- Modelo E-R (Si el sistema así lo requiere).

Vista de Implementación:

- Diagrama de Secuencia.
- Diagrama de estados.
- Diagrama de Colaboración.

Vista Conceptual:

- Modelo de dominio

Vista física:

- Mapa de comportamiento a nivel de hardware.
- Diseño y desarrollo de casos de uso, o flujos de casos de uso arquitectónicos.
- Pruebas de los casos de uso desarrollados.

Construcción:

- Especificación de requisitos faltantes.

- Diseño y desarrollo de casos de uso y/o flujos de acuerdo con la planeación iterativa.
- Pruebas de los casos de uso desarrollados, y pruebas de regresión según sea el caso.

Transición:

- Pruebas finales de aceptación.
- Puesta en producción.
- Estabilización.

[23, 24]

3.4.4.5. Aspectos de calidad en RUP

Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. Cada actividad incluye su verificación; es decir, todo aquello que se haga, no se habrá terminado, hasta que se verifique el cumplimiento de lo que se debía hacer.

A diferencia de otras metodologías, el control de calidad no debe realizarse al final, sino en todos los aspectos de la producción.

El rendimiento pobre de las aplicaciones y la escasa fiabilidad son factores comunes que privan desastrosamente la aceptabilidad de las aplicaciones de software de hoy en día. Por lo tanto, la calidad debe ser revisada con respecto a los requisitos basados en la fiabilidad, funcionalidad, rendimiento de las aplicaciones y el rendimiento del sistema.

Rational Unified Process nos ayuda cumplir con la calidad en la planificación, diseño, implementación, ejecución y evaluación de estas pruebas.

La evaluación de la calidad se construye en el proceso, en todas las actividades, la participación de todos los participantes, con el objetivo de proporcionar al cliente un software de calidad.

3.4.5. Metodología SCRUM

En términos sencillos, Scrum es una metodología más que permite llevar a cabo el desarrollo de un proyecto, esta metodología está dentro de las categorías ágiles, es decir permite realizar proyectos de manera rápida.

3.4.5.1. Características de SCRUM

Es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Los roles más importantes son: ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (interesados externos o internos), y el Team que incluye a los desarrolladores.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo), el equipo crea un incremento de software potencialmente entregable, el conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint.¹ Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Las características más marcadas que se logran notar en Scrum serían: gestión regular de las expectativas del cliente, resultados anticipados, flexibilidad y adaptación, retorno de inversión, mitigación de riesgos, productividad y calidad, alineamiento entre cliente y equipo, por último equipo motivado. Cada uno de estos puntos mencionados hace que el Scrum sea utilizado de manera regular en un conjunto de buenas prácticas para el trabajo en equipo y de esa manera obtener resultados posibles.[25]

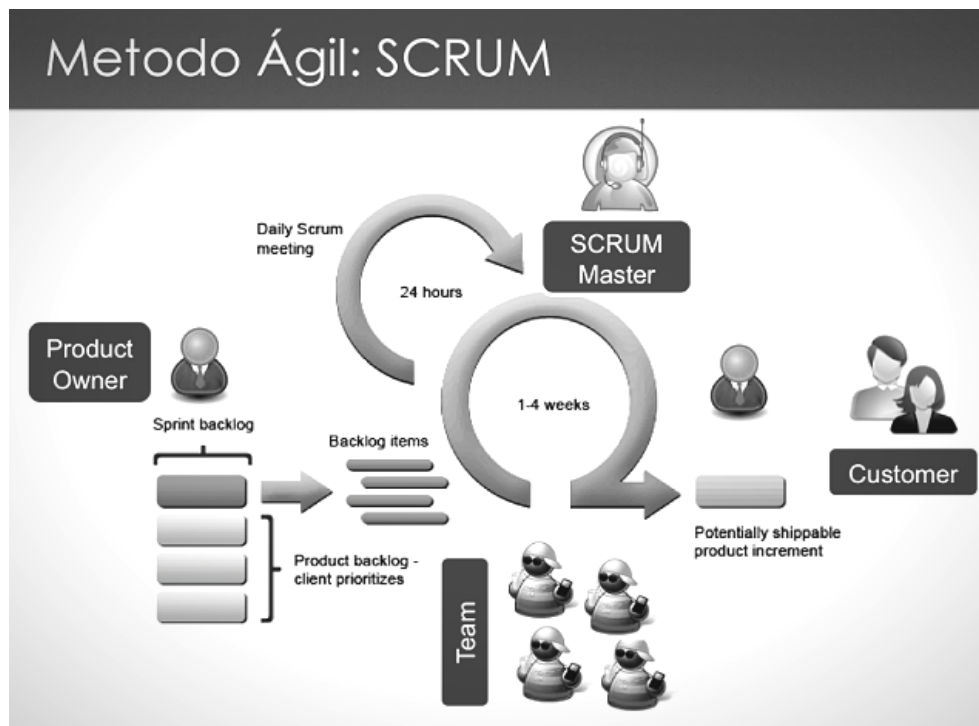


Figura 3.2: Diagrama SCRUM
Elaborado por: Edgar Hipólito Quishpe Collaguazo
Fuente: [26, 27]

3.4.5.2. Roles

■ Roles Principales

Product Owner: El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog.

ScrumMaster (o Facilitador): El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.

Equipo de desarrollo: El equipo tiene la responsabilidad de entregar el producto. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

- **Roles Auxiliares**

Stakeholders (Clientes, Proveedores, Vendedores, etc): Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirán el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

Administradores (Managers): Es la gente que establece el ambiente para el desarrollo del producto.

Reuniones en Scrum

Daily Scrum o Stand-up meeting: Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama daily standup o Stand-up meeting. El scrum tiene unas guías específicas:

- La reunión comienza puntualmente a su hora.
- Todos son bienvenidos, pero sólo los involucrados en el proyecto pueden hablar.
- La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.

Durante la reunión: cada miembro del equipo contesta a tres preguntas:

- ¿Qué has hecho desde ayer?
- ¿Qué es lo que harás hasta la reunión de mañana?
- ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar estos impedimentos).[27, 25]

3.4.5.3. SCRUM de SCRUM

Cada día normalmente después del “Daily Scrum”:

- Estas reuniones permiten a los grupos de equipos discutir su trabajo, enfocándose especialmente en áreas de solapamiento e integración.
- Asiste una persona asignada por cada equipo.

La agenda será la misma que la del Daily Scrum, añadiendo además las siguientes cuatro preguntas:

- ¿Qué ha hecho tu equipo desde nuestra última reunión?
- ¿Qué hará tu equipo antes que nos volvamos a reunir?
- ¿Hay algo que demora o estorba a tu equipo?
- ¿Estás a punto de poner algo en el camino del otro equipo?[27, 25]

3.4.5.4. Reunión de Planificación del Sprint (Sprint Planning Meeting)

Al inicio del ciclo Sprint (cada 15 o 30 días), una “Reunión de Planificación del Sprint” se lleva a cabo:

- Seleccionar qué trabajo se hará.
- Preparar, con el equipo completo, el Sprint Backlog que detalla el tiempo que tomará hacer el trabajo.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.
- Ocho horas como límite.

Al final del ciclo Sprint, dos reuniones se llevarán a cabo: la “Reunión de Revisión del Sprint” y la “Retrospectiva del Sprint”. [27, 25]

3.4.5.5. Reunión de Revisión del Sprint (Sprint Review Meeting)

- Revisar el trabajo que fue completado y no completado.
- Presentar el trabajo completado a los interesados (alias “demo”) .
- El trabajo incompleto no puede ser demostrado.
- Cuatro horas como límite.[27, 25]

3.4.5.6. Retrospectiva del Sprint (Sprint Retrospective)

Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.[27, 25]

3.4.5.7. Sprint

El Sprint es el período en el cual se lleva a cabo el trabajo en sí. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y el resultado obtenido es un producto potencialmente entregable al cliente. La constancia permite la concentración y mejora la productividad del equipo de trabajo.[27, 25]

3.4.5.8. Beneficios de SCRUM

- Flexibilidad a cambios. Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado.
- El marco de trabajo está diseñada para adecuarse a las nuevas exigencias que implican proyectos complejos.
- Reducción del Time to Market. El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- Mayor calidad del software. El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad.
- Mayor productividad. Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- Maximiza el retorno de la inversión (ROI). Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- Predicciones de tiempos. A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.

3.4.5.9. Descripción de SCRUM

Con la metodología Scrum el cliente se entusiasma y se compromete con el proyecto dado que lo ve crecer iteración a iteración. Asimismo le permite en cualquier

momento realinear el software con los objetivos de negocio de su empresa, ya que puede introducir cambios funcionales o de prioridad en el inicio de cada nueva iteración sin ningún problema.

Esta metódica de trabajo promueve la innovación, motivación y compromiso del equipo que forma parte del proyecto, por lo que los profesionales encuentran un ámbito propicio para desarrollar sus capacidades.

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

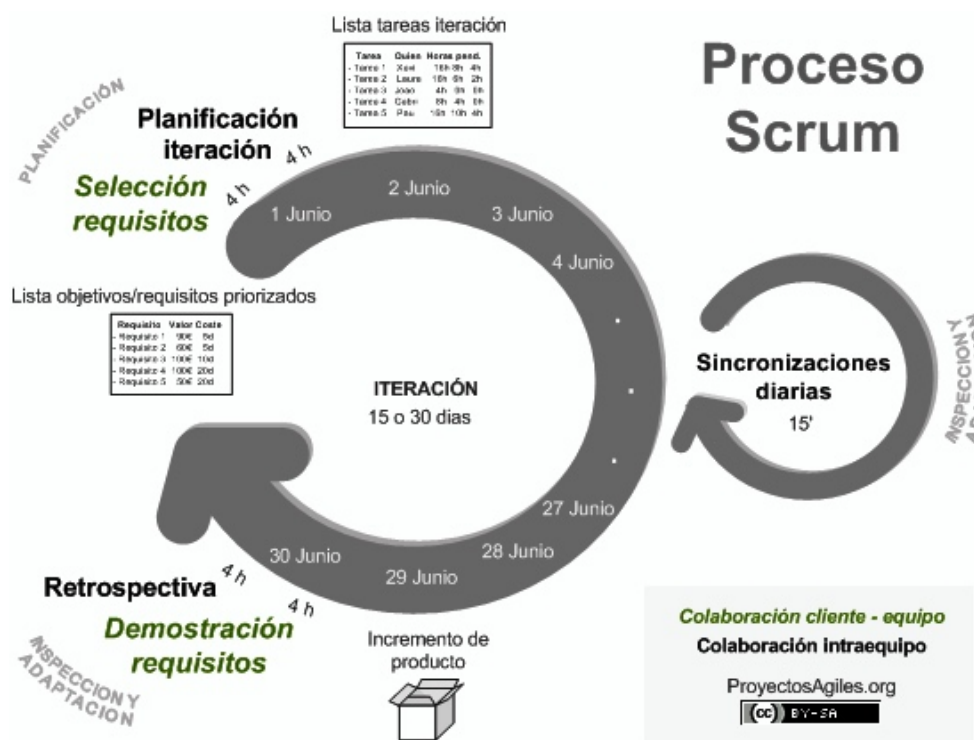


Figura 3.3: Diagrama SCRUM
 Elaborado por: proyectosagiles.org
 Fuente: [28]

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la replanificación de objetivos que rea-

liza al inicio de cada iteración.

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 (4 horas máximo). 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 (4 horas máximo). 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 autoasignan las tareas.

Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). 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 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.

Inspección y adaptación

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

1. Demostración (4 horas máximo). 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.
2. Retrospectiva (4 horas máximo). 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.[28]

3.4.5.10. Aspectos de Calidad SCRUM

La calidad no es opcional. Es un aspecto que siempre decide el cliente no quien gestiona el proyecto.

El aspecto clave la gestión de la calidad es mantenerla a lo largo del proyecto. Es sumamente difícil medir el nivel de calidad de una manera objetiva, precisamente, porque es algo que depende de los usuarios, de los clientes y de su percepción subjetiva. Pero siempre se tiene una idea más o menos intuitiva sobre cuál es el nivel de calidad que nuestro cliente requiere.

Las claves son claras: descubrir el nivel mínimo de calidad que el cliente está dispuesto a aceptar y mantenerse alrededor del mismo durante todo el proyecto. La pregunta es ¿qué mecanismos establece Scrum para que esto ocurra?: El sprint review y los incrementos de funcionalidad potencialmente entregables.

El sprint review, actúa como regulador de la calidad. Por un lado nos permitir detectar, mediante el feedback que los asistentes aporten, si estamos lejos o no del nivel de calidad requerido. Además, que el equipo presente las nuevas funcionalidades, es un catalizador clave de la calidad: asegurando una calidad mínima y asegurando que el equipo percibe cual es el nivel de calidad mínimo aceptable.

Un error habitual es recortar el sprint review o saltárselo, de tal manera que el equipo nunca llega a descubrir cuál es el nivel de calidad mínimo aceptable por el cliente o no se asegura que no estamos lejos de este.

3.4.6. Metodología XP (Extreme Programming)

Esta metodología es una de las mas destacadas, la principal deferencia con otras metodologías es que pone mayor énfasis en la adaptabilidad. Se puede considerar como la adopción de las mejores metodologías de desarrollo, esta metodología busca eliminar actividades relacionadas con la elaboración de algunos documentos de especificaciones que no tienen relación directa con el resultado final del software. Se basa en los siguientes valores principales:

1. **Comunicación:** Permite que los involucrados intercambien ideas e información sobre el proyecto.
2. **Sencillez:** Se trata de realizar el proyecto con el menor número de clases y métodos que no contengan código duplicado.
3. **Retroalimentación:** Debe ser rápida en todos los niveles, principalmente se consigue ejecutando y probando el código, por lo que las entregas tempranas y frecuentes son muy importantes.
4. **Valor:** Los involucrados deben valorar el trabajo de todos y no deben molestarse si un miembro del grupo realiza cambios en el trabajo de alguien más.
5. **Respeto:** Es el valor mas importante que debe existir en el equipo de trabajo, el la base para lograr una entrega a corto tiempo gracias a la cooperación que exista entre todos.[29]

La programación extrema se basa en:

Desarrollo en iteraciones: Se agrega nuevas funcionalidades cada que se tiene una iteración.

Pruebas unitarias continuas: Mediante estas pruebas se comprueba que la aplicación mantenga sus funcionalidades.

Programación en parejas: Permite que los programadores puedan discutir sobre el código mientras se desarrolla el programa.

Interacción entre los desarrolladores y el usuario: Gracias a esto se podrá minimizar el esfuerzo de ambas partes ya que se tendrá una mejor comprensión sobre los requerimientos o cambios y el desarrollador podrá describir una solución en ese instante.

Refactorización de código: El objetivo es hacer el código mucho mas fácil y mantenible garantizando un funcionamiento optimo.

Propiedad de código compartido: Todos los integrantes del grupo de desarrollo pueden modificar el código hecho por otro.

Simplicidad de diseño: Gracias a los diseños simples se puede agregar funcionalidades de manera fácil y rápida en el futuro.

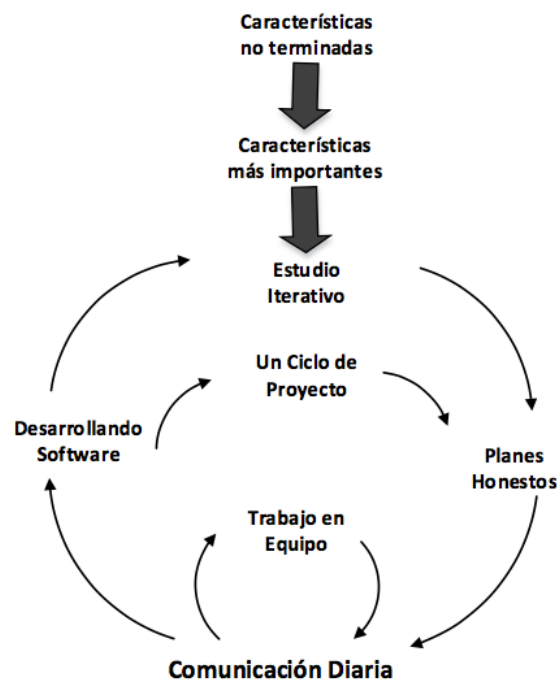


Figura 3.4: Flujo de características de la Programación Extrema
Elaborado por: extremeprogramming.org
Fuente: [29]

3.4.6.1. Proceso XP

El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.[30]

3.4.6.2. Fases de desarrollo de la Programación Extrema

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

“Fase I: Exploración:

Aquí el cliente plantea todas las historias de usuario que sea posible, lo que sea mas necesario resolver en la primera entrega. En paralelo a lo anterior el equipo de desarrollo empieza a empaparse en el uso de la herramienta y técnicas a usar, se prueba la tecnología que servirá de intermediario y se realizan prototipos para indicar al usuario.

Fase II Planificación de la Entrega: Aquí el cliente establece las prioridades de cada historia de usuario para que el programador realice una estimación de esfuerzo necesario de cada una. Se acuerda la primera entrega. Una de las restricciones de esta metodología es que las entregas no debería sobre pasar mas de tres meses.

Fase III Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

Fase IV: Producción: En esta fase se realizan pruebas y revisión del rendimiento para luego montar el sistema en el entorno de desarrollo. En paralelo se deberá determinar las nuevas características que se adicionara en una próxima entrega.

Fase V: Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura.

La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo”. [30]

3.4.7. Comparación

XP vs RUP

La metodología XP es adecuada para cualquier tipo de proyecto y para equipos de cualquier tamaño, muchos novatos que quieren iniciar un trabajo de investigación piensan que la XP es para equipos muy chicos, de dos a cuatro personas, y para proyectos de corta duración y metas modestas.

En el caso del RUP, sin embargo, la implementación de estos procesos para equipos pequeños o proyectos chicos se convierte en un gasto de tiempo y dinero. La mayoría de los equipos que utilizan RUP lo han hecho por varios años y tienen especialistas del proceso que facilitan a los miembros sin experiencia el conocimiento de las prácticas y metodologías. La metodología XP es mucho más fácil de implementar y de aprender, por lo que los equipos jóvenes pueden incorporarla de manera más natural.

En la Transición del RUP o entrega de XP las diferencias pueden ser mayores, para RUP la entrega final debe ser algo mucho más definido, mientras que en XP se realizan entregas continuas y discretas que permiten evaluar el sistema conforme se colocan las versiones finales. Sin embargo, por el esquema iterativo de ambas, las dos metodologías contemplan entregas parciales después de cada iteración para una evaluación y monitoreo continuo.

XP vs SCRUM

Scrum es un marco de trabajo iterativo e incremental que está enfocado a la gestión de procesos de desarrollo y administración de proyectos.

XP, por su parte, es una metodología de desarrollo de la ingeniería de software, formulada por Kent Beck, que considera que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.

Como se puede entender en el descrito anterior, procedemos a descartar la metodología SCRUM ya que es una metodología que esta enfocada mas bien a la administración de procesos como un marco de trabajo para la elaboración de estas.

A fin de descartar una metodología mas, se analizará las dos ultimas opciones

mediante los siguientes cuadros.

CARACTERÍSTICAS	
XP	RUP
Desarrollo iterativo e incremental	Desarrollo interno en etapas interactivas
Pruebas unitarias continuas, frecuentemente repetidas y automatizadas	Esta integrado en todo el ciclo de vida
Programación por parejas	Programación por equipos
Interacción con el usuario final	Interacción con el usuario estratégico
Refactorización del código	
Propiedad del código	
Simplicidad del código	
ROLES	
XP	RUP
Programador	Analistas
Encargado de pruebas	Desarrolladores
Cliente	Gestores
Encargado del seguimiento	Apoyo y especialistas
Entrenador ó coaching	Stakeholders
Consultor	Revisor
Gestor	Coordinación de revisiones
	Revisor técnico
SELECCIÓN DE METODOLOGÍA	
XP	RUP
Los requisitos cambian (clientes indecisos)	Comunicación entre equipos
Proyectos con alto grado de riesgos	Complejidad de desarrollo de acuerdo al tamaño del proyecto
Grupos pequeños de programadores entre 2 y 12	Configuración y control de cambios(Artefactos)
VENTAJAS	
XP	RUP
Comunicación	Mayor documentación
Simplicidad	Verificar la calidad de software
Realimentación	Configuración y control de cambios
Coraje (satisfacción de los programadores)	Es modelado, guiado por casos de uso
Disminuye traza de errores	Es centrado en arquitectura, guiado por riesgos
Alta calidad mínimo tiempo	
DESVENTAJAS	
XP	RUP
Dificultad para determinar el costo de proyecto	Los cambios son en una fase
Se usa principalmente en proyectos pequeños	Proyectos grandes

Tabla 3.1: Comparativa entre metodologías RUP y XP N° 2

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Fuente: [31]

Metodología	Características	Ventajas	Desventajas
RUP	Forma disciplinada de asignar las tareas y responsabilidades.	Los usuarios están involucrados continuamente.	Requiere conocimiento del proceso y UML.
	Desarrollo incremental en etapas.	Permite definir de manera clara: Quien debe hacer las cosas, Que debe hacerse, Cuando y Como.	En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.
	Control de cambios.	Se puede usar para pequeños, medianos y grandes proyectos.	El grado de complejidad puede no ser adecuado para ciertos proyectos.
	Programación por equipos.	El conocimiento adquirido en una iteración puede aplicarse de iteración a iteración.	Las iteraciones en cada ciclo puede tomar mas tiempo de lo previsto.
	Desarrollo incremental y en iteraciones.	Se adapta muy bien a los requisitos cambiantes.	Dificultad para llegar a saber el costo y el tiempo que tomará el proyecto.
XP	Programación por parejas.	Cuenta con una tasa de errores muy pequeña.	Es recomendable emplearla solo en proyectos a corto plazo.
	El Usuario es parte del equipo de desarrollo.	Permite superar con mayor facilidad circunstancias imprevistas en el desarrollo de software.	En caso de fallar, las comisiones son muy altas.
	Simplicidad del Código.	Garantizar que el software funcione como el usuario lo requiere.	Generalmente se aplica solo para proyectos pequeños, no muy eficiente en proyectos de gran tamaño.
	Entregas rápidas y frecuentes.	Puede ser aplicada a cualquier lenguaje de programación.	En caso de fallar, las comisiones son muy altas.
	Pruebas unitarias Continuas.	Facilita los cambios.	Puede no siempre ser más fácil que el desarrollo tradicional.

Tabla 3.2: Comparativa entre metodologías RUP y XP N° 1

Elaborado por: Edgar Hipólito Quishpe Collaguazo
Fuente: [25, 29]

Analizado estos cuadros, se descarta la metodología RUP debido a que responde mejor a proyectos de mayor escala que la propuesta y se selecciona la metodología XP como la mas acertada para servir de directriz de este trabajo investigativo y de desarrollo.

3.4.8. Metodología seleccionada

Es comprensible entender que no existe una metodología que sea en su totalidad un éxito al aplicarla en algún proyecto ya que es difícil adaptarla a todo el contexto del proyecto, no obstante se ha seleccionado la metodología de desarrollo Extreme Programming en vista de que nos proporciona muchas ventajas permitiendo ahorrar tiempo y recursos.

Los aspectos mas sobresalientes encontrados en esta metodología son:

- Permite la entrega de productos usables con mayor rapidez lo que permite la entrega de versiones continuas al usuario.
- Presenta un código más simple y fácil de entender que permite la reducción del número de errores en su desarrollo.
- Capacidad de respuesta ante cambios permitiendo la adaptación a nuevos requisitos de la empresa u organización.

CAPÍTULO 4

Desarrollo de la propuesta

4.1. Datos informativos

Título

Aplicación móvil para consulta de turnos de regadío del Canal de Riego Píllaro Ramal Norte Yatchil Central.

Institucionales

- Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Beneficiarios

- Usuarios del Canal de Riego de Yatchil Central
- Directiva del Canal de Riego Yatchil Central

Ubicación

Provincia: Tungurahua

Cantón: Santiago de Píllaro

Parroquia: San Andrés

Barrio: Yatchil Central

Equipo responsable

Tutor: Ing. Clay Fernando Aldas Flores

Investigador: Edgar Hipólito Quishpe Collaguazo

Tomando en cuenta la metodología Extreme Programming (XP), se detalla a continuación el proceso de desarrollo del proyecto.

4.2. Exploración

En esta primera etapa de la metodología XP, se toma como base de análisis el proceso manual que realiza actualmente la directiva y los usuarios del Canal de Riego de Yatchil Central.

El agua para el regadío nace en las vertientes del páramo de Pisayambo, este avanza a lo largo de 15Km hasta llegar a la comunidad de Santa Rita, ahí se divide el caudal del agua repartiendo una parte al sur y otra al norte. El caudal dirigido al norte avanza por casi 10Km hasta llegar a Yatchil Central, el agua va ingresando en los tanques de reserva que están ubicados en lugares estratégicos por todas partes que recorre el agua, luego sigue avanzando hasta su destino final en las quebradas de Cardo Santo. En cada comunidad existe una persona llamada aguatero que se encarga de abrir las compuertas del canal o del tanque de reserva para que el agua circule a los terrenos a través de la acequia secundaria.



Figura 4.1: Compuerta del canal
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Para la distribución individual del agua existe una directiva que realiza una lista de turnos de regadío de forma manual en hojas de cuaderno, a cada solar de terreno que esta inscrito se le acredita dos horas de uso de agua y en función de esto se va acreditando horas de riego a cada uno de los usuarios, una vez finalizado la creación de todos los turnos de riego en hojas de cuaderno que toma aproximadamente 5 días, se procede a transferir a una hoja de calculo en Excel.

Cuando se tiene los turnos ingresados en Excel, se comunica a los socios del canal de riego para que se acerquen a la casa barrial para que retiren el turno respectivo a un costo de 0,30 ctvs. por turno.

Visto esta situación, se ha propuesto brindar solución mediante una base de datos que albergue toda la información necesaria del usuario y que permita generar la calendarización de forma automática mediante un trigger, así mismo se realizará una aplicación móvil para que el usuario pueda revisar sus turnos de regadío mediante una consulta ejecutado desde su equipo terminal, por ultimo se desarrollara una aplicación web para que el administrador del canal de riego pueda generar la calendarización, modificar datos del usuario, modificar turnos, entre otros.

Una vez hecho el análisis, se procede a establecer los diferentes requerimientos técnicos para ser utilizadas en el desarrollo de la aplicación móvil, esto se llevara a cabo tanto a nivel de software como a nivel de hardware.

4.3. Planificación del proyecto

4.3.1. Requerimientos

Recursos humanos

Para la recolección de datos, se requerirá del personal administrativo de la Directiva del Canal de Riego y personal encargado de la distribución de los turnos.

La dirección de un docente tutor que oriente a la ejecución correcta de las actividades a realizarse y que encamine en el desarrollo mismo del trabajo.

El rol de desarrollador será el trabajo principal que deberá asumir el investigador para resolver el problema planteado.

Recursos Materiales

■ Software para desarrollo:

Las herramientas utilizadas para el desarrollo de la aplicación son gratuitas y licenciadas bajo costo, esto nos va a permitir obtener un resultado de alta calidad.

Software	Qué hace?
MySQL Server 5.6	Por medio del SGBD Workbench 6.3, nos proporciona todo el recurso para el manejo de la información, creación de funciones y/o procedimientos, triggers, etc.
Android Studio 2.1.2	El desarrollo de la aplicación Móvil se llevara a cabo en Android Studio, este ofrece un entorno de desarrollo multiplataforma de aplicaciones para plataforma Android, en el se puede encontrar los recursos necesarios para implementar las apps bajo el lenguaje de programación Java 8 Update 91.
Genymotion 2.5.2	Proporciona la posibilidad de simular un dispositivo móvil con plataforma Android pudiendo ver los cambios que se realiza en la aplicación durante el desarrollo en tiempo real y comprobar cómo se visualiza en diferentes dispositivos con distintas configuraciones y resoluciones de forma simultánea.
Php Generator for Mysql Professional 16.9.0.3	Es un software con interfaz GUI de paga para la creación scripts PHP de alta calidad, permite generar conexión a una base de datos para seleccionar tablas, vistas y consultas a fin de crear paginas de gestión de datos (añadir, editar y eliminar registros), proporciona la capacidad de personalizar la apariencia HTML y mucho mas.

Tabla 4.1: Software de desarrollo

Elaborado por: Edgar Hipólito Quishpe Collaguazo

■ Servicios:

Para poder acceder a la base de datos y página web desde Internet, es necesario contratar un servicio de alojamiento web y un dominio que se detalla a continuación.

Servicio	Qué ofrece?
HostGator	Es un proveedor de alojamiento web con muy buenas características como el ilimitado espacio de almacenamiento, ilimitado ancho de banda, servicio de compra de dominios, entre otros.

Tabla 4.2: Servicios

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.3.2. Historias de usuario

Una historia de usuario es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos (acompañadas de las discusiones con los usuarios y las pruebas de validación). Cada historia de usuario debe ser limitada, ésta debería poderse escribir sobre una nota adhesiva pequeña. Dentro de la metodología XP las historias de usuario deben ser escritas por los clientes.

4.3.2.1. Historia de usuario de aplicación móvil

Después de realizar el análisis del funcionamiento manual que lleva actualmente la directiva del Canal de Riego y socios usuarios de la misma, verbalmente nos

han descrito como quisieran que fuera la aplicación móvil y la aplicación web, las pantallas que debería incorporar y las acciones que tendría que ejecutar en cada una de ellas, de esta manera trasladando lo escrito en lenguaje común a una plantilla mejor elaborada, se ha obtenido las siguientes historias de usuario.

Historia de usuario	
Número: 1	Usuario: Todos
Nombre de historia: Mostrar pantalla inicial de ingreso a consultas.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 4	Iteración asignada: 1
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: El usuario visualizará una pantalla donde pueda seleccionar entre dos opciones el ingreso a consulta general o individual, para la consulta general no necesitara ingresar ningún parámetro, pero para el ingreso a consulta individual, deberá ingresar el número de cédula, si la cédula esta incorrecta, la aplicación mostrará un mensaje "Campo vacío o Cédula Incorrecta", si la cédula es correcta se realizará una consulta para verificar que el usuario esta registrado en la tabla de usuario de la base de datos del canal de riego, si se encuentra registrado, la aplicación le dirigirá a una nueva pantalla donde se mostrará los componentes de selección de turnos y su respectivo menú de acciones para "Mostrar turnos", "Generar PDF" y "Ayuda".	
Observaciones: La aplicación debe revisar la conexión antes de permitir el ingreso a la pantalla de consulta.	

Tabla 4.3: Historia de usuario 1 - Mostrar pantalla inicial de ingreso al sistema

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Historia de usuario	
Número: 2	Usuario: Todos
Nombre de Historia: Consultar turnos en la aplicación móvil.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 5	Iteración asignada: 2
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: El usuario podrá realizar la consulta del turno de riego de cualquier toma y módulo, si es una consulta general, se mostrará todos los turnos de la toma y modulo que corresponda al elemento actual seleccionado, si es una consulta individual, se mostrará todos los turnos de la toma y modulo que corresponda al elemento actual seleccionado y que acierten en la condición de ser igual al número de cédula ingresados en la pantalla anterior que describe la historia de usuario 1.	
Observaciones: En esta pantalla añadir un FloatingActionButton para permitir realizar acciones para "Mostrar turnos", "Generar PDF" y "Ayuda".	

Tabla 4.4: Historia de usuario 2 - Consultar turnos en la aplicación móvil

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Historia de usuario	
Número: 3	Usuario: Todos
Nombre de Historia: Generar archivo PDF.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: El usuario que realice tanto la consulta general como la individual de turnos de riego, luego de que la aplicación muestre el listado en pantalla, podrá guardar ese listado en un archivo PDF dentro del espacio de almacenamiento externo del dispositivo móvil.	
Observaciones: La ruta donde se guardará el archivo PDF es el siguiente: “/sdcard/” y el nombre del archivo se deberá llamar, en el caso de consulta general: igual que el nombre de la toma y modulo actual seleccionado y en el caso de consulta individual: igual que la toma y el módulo actual seleccionado mas el nombre del usuario. Por esta razón el usuario deberá contar con una tarjeta Micro SD en su dispositivo móvil para que la aplicación pueda escribir sobre ella.	

Tabla 4.5: Historia de usuario 3 - Generar archivo PDF

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.3.2.2. Historia de usuario de aplicación web

Historia de usuario	
Número: 4	Usuario: Todos
Nombre de Historia: Acceder a la aplicación web.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 5	Iteración asignada: 4
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Para el manejo de la base de datos, se deberá crear una aplicación web que muestre como pantalla inicial la respectiva vista de acceso al sistema. Al sistema podrán acceder por medio de una autentificaron como usuario invitado o administrador. Todo el que visite esta pagina, podrá ingresar como usuario invitado para realizar consultas de los turnos de riego. Por otra parte se permitirá el ingreso a un usuario administrador, que tendrá permisos de escritura en la base de datos, en consenso con la directiva del canal de riego de Yatchil Central, se le entregará la clave de usuario administrador únicamente al presidente.	
Observaciones: <ul style="list-style-type: none"> • El nombre y clave de usuario invitado, deberá estar impreso en la misma página de inicio de sesión. • Las pantallas de consultas mostraran toda la lista de turnos de regadío, estas podrán ser filtradas por el nombre del usuario. 	

Tabla 4.6: Historia de usuario 4 - Acceder a la aplicación web

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Historia de usuario	
Número: 5	Usuario: Todos
Nombre de Historia: Consultar turnos en la aplicación web.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 6	Iteración asignada: 5
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Los usuarios pueden visualizar la lista de los turnos de riego de todas las tomas y módulos, podrán filtrar los turnos de su interés ya sea por el nombre, la fecha, número de solares, número de horas, etc. Una vez filtrado la lista de su interés, el sistema le proporcionará las opciones de exportar los datos a un archivo PDF o Excel, de igual forma le proporcionará una interfaz amigable para imprimir la lista.	
Observaciones: Ninguno.	

Tabla 4.7: Historia de usuario 5 - Consultar turnos en la aplicación web

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Historia de usuario	
Número: 6	Usuario: Administrador
Nombre de Historia: Insertar y actualizar usuarios.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 7	Iteración asignada: 6
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: El usuario administrador tendrá los privilegios de realizar cambios o insertar nuevos registros en la tabla de usuarios, se deberá considerar los siguientes atributos: cédula de ciudadanía, nombre, apellido, dirección, teléfono, email, estado.	
Observaciones: No se debe permitir eliminar ningún registro de la tabla usuarios, el administrador únicamente podrá dar de baja el registro mediante un checkbox en el atributo estado.	

Tabla 4.8: Historia de usuario 6 - Insertar y actualizar usuarios

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Historia de usuario	
Número: 7	Usuario: Administrador
Nombre de Historia: Insertar y actualizar y eliminar turnos.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 7	Iteración asignada: 7
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: El usuario administrador tendrá los privilegios de realizar inserciones, actualizaciones y eliminaciones en la tabla de turnos, se deberá considerar los siguientes atributos: id, usuario, total terreno, referencia, h1_hi, h1_hf, h2_hi, h2_hf, h3_hi, h3_hf, h4_hi, h4_hf, h5_hi, h5_hf.	
Observaciones: En vista a lo extenso que resulta ser los campos de turnos, se deberá abreviar de la siguiente manera. horario uno_hora inicial = h1_hi horario uno_hora final = h1_hf	

Tabla 4.9: Historia de usuario 7 - Insertar, actualizar y eliminar turnos

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Historia de usuario	
Número: 8	Usuario: Administrador
Nombre de Historia: Generar calendarios.	
Prioridad en negocio: (Alta / Media / Baja)	Riesgo en desarrollo: (Alta / Media / Baja)
Puntos estimados: 9	Iteración asignada: 8
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: El usuario administrador podrá insertar y eliminar los datos de inicio de calendarización. Cuando el administrador realiza la inserción de un registro en el que deberá ingresar la fecha de inicio y la tabla a la que se va a generar los calendarios, se disparará un trigger que ejecute un procedimiento que genere automáticamente los calendarios.	
Observaciones: Se deberá buscar una solución mediante programación sql que solucione la automatización de los calendarios.	

Tabla 4.10: Historia de usuario 8 - Generar calendarios

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.3.3. Estimación de historias de usuario

Tomando en cuenta que un punto de estimación es equivalente a una semana de trabajo de lunes a viernes haciendo uso de 4 horas diarias, se ha estimado un total

de 3 puntos a cada historia de usuario entregado y listo para publicar su avance, multiplicando por las 8 historias se obtiene un total de 24 puntos estimados que necesita nuestro proyecto para ser desarrollado en su totalidad y que este listo para subir a publicación. Así se ilustra en la siguiente tabla:

N°	HISTORIA DE USUARIO	Puntos de estimación		
		TIEMPO ESTIMADO EN:		
		SEMANAS	DÍAS	HORAS
1	Mostrar pantalla inicial de ingreso a consultas.	3	15	60
2	Consultar turnos en la aplicación móvil.	3	15	60
3	Generar archivo PDF.	3	15	60
4	Acceder a la aplicación web.	3	15	60
5	Consultar turnos en la aplicación web.	3	15	60
6	Insertar y actualizar usuarios.	3	15	60
7	Insertar y actualizar y eliminar turnos.	3	15	60
8	Generar calendarios.	3	15	60
	TOTAL:	24	120	480

Tabla 4.11: Puntos de estimación

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4. Iteraciones

Una iteración es repetir un proceso con la intención de alcanzar una meta deseada, objetivo o resultado. Cada repetición del proceso también se le denomina una "iteración", y los resultados de una iteración se utilizan como punto de partida para la siguiente iteración.

4.4.1. Plan de entrega

Visto las prioridades de las historias de usuario, en consenso entre la directiva del canal de riego y el equipo de desarrollo se ha determinado 8 iteraciones que

hará cumplir el desarrollo total de las 8 historias de usuario dividido en varias tareas, asignando así una historia a una iteración, cada iteración no excederá las 3 semanas. Se deberá ajustar en lo posible al siguiente cuadro que indica el avance de cada iteración y el plan de entrega. Cada 21 días se iniciará a desarrollar las tareas de una nueva historia de usuario y al mismo tiempo se entregará software útil resultado de las tareas realizadas de una historia de usuario anterior. Al final de la última iteración, el sistema estará listo para ser entregado y llevarlo a producción.

AVANCE DEL PROYECTO EN CADA ITERACIÓN					
Nº Iteración	HISTORIA DE USUARIO PARA ENTREGA	FECHA DE ENTREGA	NUOVA HISTORIA DE USUARIO	FECHA DE COMIENZO	% COMPLETADO
0			Mostrar pantalla inicial de ingreso a consultas.	1 de septiembre de 2016	0%
1	Mostrar pantalla inicial de ingreso a consultas.	21 de septiembre de 2016	Consultar turnos en la aplicación móvil.	22 de septiembre de 2016	13%
2	Consultar turnos en la aplicación móvil.	14 de octubre de 2016	Generar archivo PDF.	15 de octubre de 2016	25%
3	Generar archivo PDF.	4 de noviembre de 2016	Acceder a la aplicación web.	5 de noviembre de 2016	38%
4	Acceder a la aplicación web.	24 de noviembre de 2016	Consultar turnos en la aplicación web.	25 de noviembre de 2016	50%
5	Consultar turnos en la aplicación web.	16 de diciembre de 2016	Insertar y actualizar usuarios.	17 de diciembre de 2016	63%
6	Insertar y actualizar usuarios.	6 de enero de 2017	Insertar y actualizar y eliminar turnos.	7 de enero de 2017	75%
7	Insertar y actualizar y eliminar turnos.	28 de enero de 2017	Generar calendarios.	29 de enero de 2017	88%
8	Generar calendarios.	22 de febrero de 2017			100%

Tabla 4.12: Avance de iteraciones y plan de entrega

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4.2. Tareas

Basado en los puntos de estimación que suman 24 semanas (Ver figura: 4.11, se ha podido establecer 13 tareas. Cada tarea deberá ser entregado en no más de 3 semanas, a la vez cada tarea entregado estará debidamente probado listo para ser puesto en producción.

4.4.2.1. Tareas de aplicación móvil

Tarea	
Número: 1	Número de historia: 1
Nombre de historia: Compra de un dominio y contratación de hosting.	
Tipo de tarea: Gestión	Puntos estimados: 1
Fecha de inicio: 01 - Septiembre - 2016	Fecha de fin: 09 - Septiembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Realizar todos los procesos transaccionales para hacer el pago de el dominio y el hosting. Se contratará los servicios de la empresa hostgator, de quienes se tiene confianza por contrataciones anteriores.	

Tabla 4.13: Tarea 1 - Compra de un dominio y contratación de hosting

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 2	Número de historia: 1
Nombre de historia: Diseño y construcción de la base de datos.	
Tipo de tarea: Programación sql	Puntos estimados: 1
Fecha de inicio: 10 - Septiembre - 2016	Fecha de fin: 18 - Septiembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Analizar, diseñar y desarrollar el mejor esquema de base de datos que responda a las operaciones que pretende realizar el cliente, realizar las pruebas necesarias y trasladar al servidor contratado.	

Tabla 4.14: Tarea 2 - Diseño y construcción de la base de datos

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 3	Número de historia: 1
Nombre de historia: Codificación y desarrollo de interfaz para ingreso a pantalla de consultas de turnos de riego de aplicación móvil.	
Tipo de tarea: Programación Android	Puntos estimados: 1
Fecha de inicio: 19 - Septiembre - 2016	Fecha de fin: 21 - Septiembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Desarrollar la interfaz, verificar si existe conexión a internet en el caso de consulta general e individual, en el caso de consulta individual, también realizar la validación de la cédula de ciudadanía y verificar si esta registrado en la tabla de usuarios, finalmente permitir ingreso.	

Tabla 4.15: Tarea 3 - Codificación y desarrollo de interfaz para ingreso a pantalla de consultas de turnos de riego de aplicación móvil

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 4	Número de historia: 2
Nombre de historia: Desarrollo de web service que permita tomar registros desde la base de datos.	
Tipo de tarea: Programación PHP	Puntos estimados: 2
Fecha de inicio: 22 - Septiembre - 2016	Fecha de fin: 06 - Octubre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: <ul style="list-style-type: none"> • Revisar los conceptos necesarios del lenguaje de programación php para la creación de web service que permita el envío de información desde la base datos en mysql hacia la aplicación Android. • Crear los web service necesarios. • Subir los web service al servidor. 	

Tabla 4.16: Tarea 4 - Desarrollo de web service que permita tomar registros desde la base de datos

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 5	Número de historia: 2
Nombre de historia: Desarrollo de la interfaz con los componentes necesarios para mostrar los turnos de riego.	
Tipo de tarea: Programación Android	Puntos estimados: 1
Fecha de inicio: 07 - Octubre - 2016	Fecha de fin: 14 - Octubre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: <ul style="list-style-type: none"> • Desarrollar una interfaz intuitiva que contenga controles para seleccionar los turnos de interés y mostrarlos en un componente de lista desplegable. • Crear una clase AsyncTask que permita realizar operaciones en segundo plano y publicar resultados en el subproceso de interfaz de usuario. • Usar el mecanismo de codificación JSON para el intercambio de información. • Desplegar información dependiendo si es consulta general, mostrar toda la lista y si es consulta individual, mostrar la lista que corresponda a la cédula de ciudadanía del usuario actual. 	

Tabla 4.17: Tarea 5 - Desarrollo de la interfaz con los componentes necesarios para mostrar los turnos de riego

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 6	Número de historia: 3
Nombre de historia: Usar librería iTextG para generar archivo PDF.	
Tipo de tarea: Programación Android	Puntos estimados: 3
Fecha de inicio: 15 - Octubre - 2016	Fecha de fin: 04- Noviembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: <ul style="list-style-type: none"> • Investigar el uso de los componentes de la librería iTextG que se usará en este trabajo. • Crear métodos escribirPDF y mostrarPDF recibiendo parámetros obtenidos por JSON. 	

Tabla 4.18: Tarea 6 - Usar librería iTextG para generar archivo PDF

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4.2.2. Tareas de aplicación web

Tarea	
Número: 7	Número de historia: 4
Nombre de historia: Comprar herramienta “PHP Generator for MySQL” e investigar funcionamiento.	
Tipo de tarea: Gestión	Puntos estimados: 1
Fecha de inicio: 05 - Noviembre - 2016	Fecha de fin: 11- Noviembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Realizar gestión de compra de la herramienta CASE, instalar en un sistema operativo compatible, revisar el funcionamiento y configurar los parámetros iniciales de acceso a la base de datos situada en el servidor.	

Tabla 4.19: Tarea 7 - Adquirir herramienta “PHP Generator for MySQL” e investigar funcionamiento

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 8	Número de historia: 4
Nombre de historia: Crear pantalla de acceso a la aplicación web.	
Tipo de tarea: Uso de herramienta CASE	Puntos estimados: 2
Fecha de inicio: 12 - Noviembre - 2016	Fecha de fin: 24 - Noviembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Haciendo uso de la herramienta CASE y CSS, realizar la pantalla de inicio al sistema y proporcionar permisos bajo el siguiente termino: <ul style="list-style-type: none"> • Usuario invitado: Puede consultar de forma general o individual los turnos de regadío, el nombre y la clave para ingresar se publicará en la misma página. • Usuario administrador: Puede generar nuevos turnos, ingresar y modificar usuarios, ingresar, modificar y eliminar turnos. 	

Tabla 4.20: Tarea 8 - Crear pantalla de acceso a la aplicación web

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 9	Número de historia: 5
Nombre de historia: Crear módulos de consulta de turnos de riego para la aplicación web.	
Tipo de tarea: Uso de herramienta CASE	Puntos estimados: 3
Fecha de inicio: 25 - Noviembre - 2016	Fecha de fin: 16 - Diciembre - 2016
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Haciendo uso de la herramienta CASE desarrollar los módulos de consulta de turnos de riego que sea visible para el usuario invitado.	

Tabla 4.21: Tarea 9 - Crear módulos de consulta de turnos de riego para la aplicación web

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 10	Número de historia: 6
Nombre de historia: Crear módulos de inserción y actualización de usuarios para la aplicación web.	
Tipo de tarea: Uso de herramienta CASE y javascript	Puntos estimados: 3
Fecha de inicio: 17 - Diciembre - 2016	Fecha de fin: 06 - Enero - 2017
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: <ul style="list-style-type: none"> Haciendo uso de la herramienta CASE, desarrollar los módulos de inserciones y actualizaciones de usuarios que sea visible para el usuario administrador. mediante código javascript, realizar la validación de la cédula de ciudadanía. 	

Tabla 4.22: Tarea 10 - Crear módulos de inserción y actualización de usuarios para la aplicación web

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 11	Número de historia: 7
Nombre de historia: Crear módulos de inserción y actualización y eliminación de turnos de riego para la aplicación web.	
Tipo de tarea: Uso de herramienta CASE	Puntos estimados: 3
Fecha de inicio: 07 - Enero - 2017	Fecha de fin: 28 - Enero - 2017
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Haciendo uso de la herramienta CASE, desarrollar los módulos de inserción, actualización y eliminación de turnos de riego que sea visible para el usuario administrador.	

Tabla 4.23: Tarea 11 - Crear módulos de inserción y actualización y eliminación de turnos de riego para la aplicación web

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 12	Número de historia: 8
Nombre de historia: Crear un trigger de base de datos que genere los turnos de riego de manera automática.	
Tipo de tarea: Programación sql	Puntos estimados: 2
Fecha de inicio: 29 - Enero - 2017	Fecha de fin: 18 - Febrero - 2017
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Realizar el análisis de cómo se estructura los calendarios actualmente e idear la forma de automatizar mediante un trigger que dispare en el evento de inserción en la tabla control.	

Tabla 4.24: Tarea 12 - Crear un trigger de base de datos que genere los turnos de riego de manera automática

Elaborado por: Edgar Hipólito Quishpe Collaguazo

Tarea	
Número: 13	Número de historia: 8
Nombre de historia: Crear módulos de inserción y eliminación de generación de calendarios.	
Tipo de tarea: Uso de herramienta CASE	Puntos estimados: 1
Fecha de inicio: 19 - Febrero - 2017	Fecha de fin: 22 - Febrero - 2017
Programador responsable: Edgar Hipólito Quishpe Collaguazo	
Descripción: Haciendo uso de la herramienta CASE, desarrollar los módulos de inserción y eliminación de calendarización que sea visible para el usuario administrador.	

Tabla 4.25: Tarea 13 - Crear módulos de inserción y eliminación de generación de calendarios

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4.3. Prototipo

Mediante los siguientes diseños se podrá observar las principales pantallas de interfaz de usuario que permitirán el manejo tanto de la aplicación móvil como el de la aplicación web.

4.4.3.1. Diseño de la pantalla de inicio de la aplicación móvil

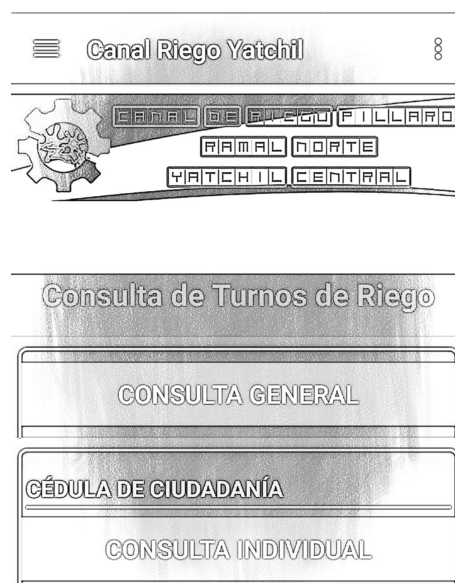


Figura 4.2: Diseño de la pantalla de inicio de la aplicación móvil
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4.3.2. Diseño de la pantalla de consulta de la aplicación móvil

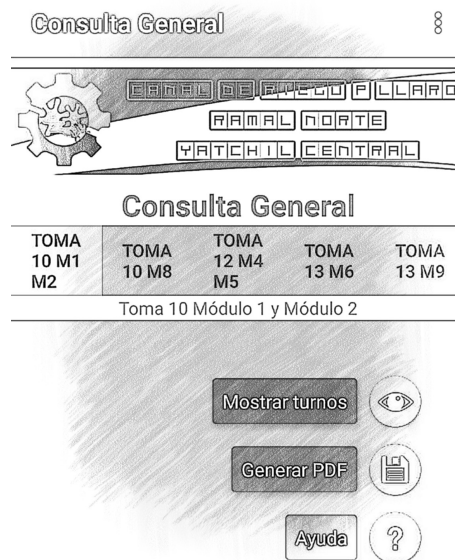


Figura 4.3: Diseño de la pantalla de consulta de la aplicación móvil
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4.3.3. Diseño de la pantalla de inicio de sesión de la aplicación web



Figura 4.4: Diseño de la pantalla de inicio de sesión de la aplicación web
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.4.3.4. Diseño de la pantalla de administración de la aplicación web



Figura 4.5: Diseño de la pantalla de administración de la aplicación web
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.5. Desarrollo

4.5.1. Iteración 1

4.5.1.1. Tarea 1 - Adquisición de un dominio y contratación de hosting

Para el efecto de esta tarea se ha realizado la contratación tanto del dominio como del hosting a la empresa Hostgator, esto nos evita configurar el DNS (Domain Name System, Sistema de Nombre de Dominio). Podemos apreciar entonces el detalle del dominio poloque.com

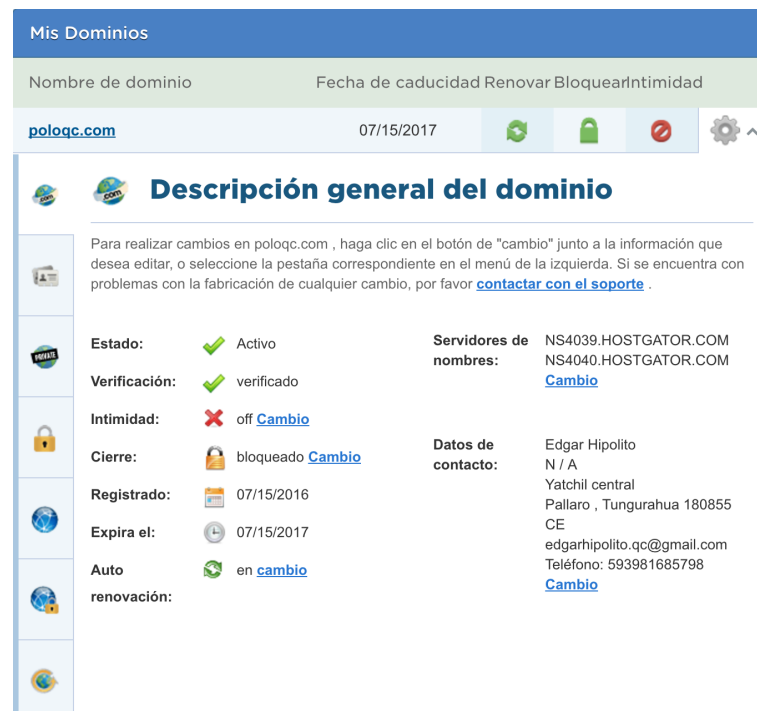


Figura 4.6: Detalle del dominio
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Una vez que se dispone del dominio y el hosting, creamos un sub dominio que se llama canalriegoYatchil que contendrá todos los recursos necesarios para poner en funcionamiento el sistema, ejemplo: los web service. La dirección URL (Uniform Resource Locator, Localizador Uniforme de Recursos) quedará así: <http://www.canalriegoYatchil.poloqc.com/ws>

4.5.1.2. Tarea 2 - Diseño y construcción de la base de datos

Motor de Base de Datos MySQL 5.6.21 con el SGBD MySQL WorkBench 6.3

A continuación se muestra el diseño de la Base de Datos que se ha establecido en consenso con la directiva del canal de riego. Las tablas de usuarios alberga los datos personales de los usuarios socios del canal, las cinco tablas que antecede la palabra turnos, almacenan los turnos de riego de cada usuario en orden estipulado por la directiva, luego esta tabla también guardará las fechas y horarios de turnos que generará el trigger, finalmente la tabla llamada control almacenará las fechas de inicio y tablas de turnos a las cuales se ha generado fechas y horas de turnos de riego, esto por su puesto ocurrirá en el momento de hacer una inserción sobre esta tabla. (Ver figura: 4.7)

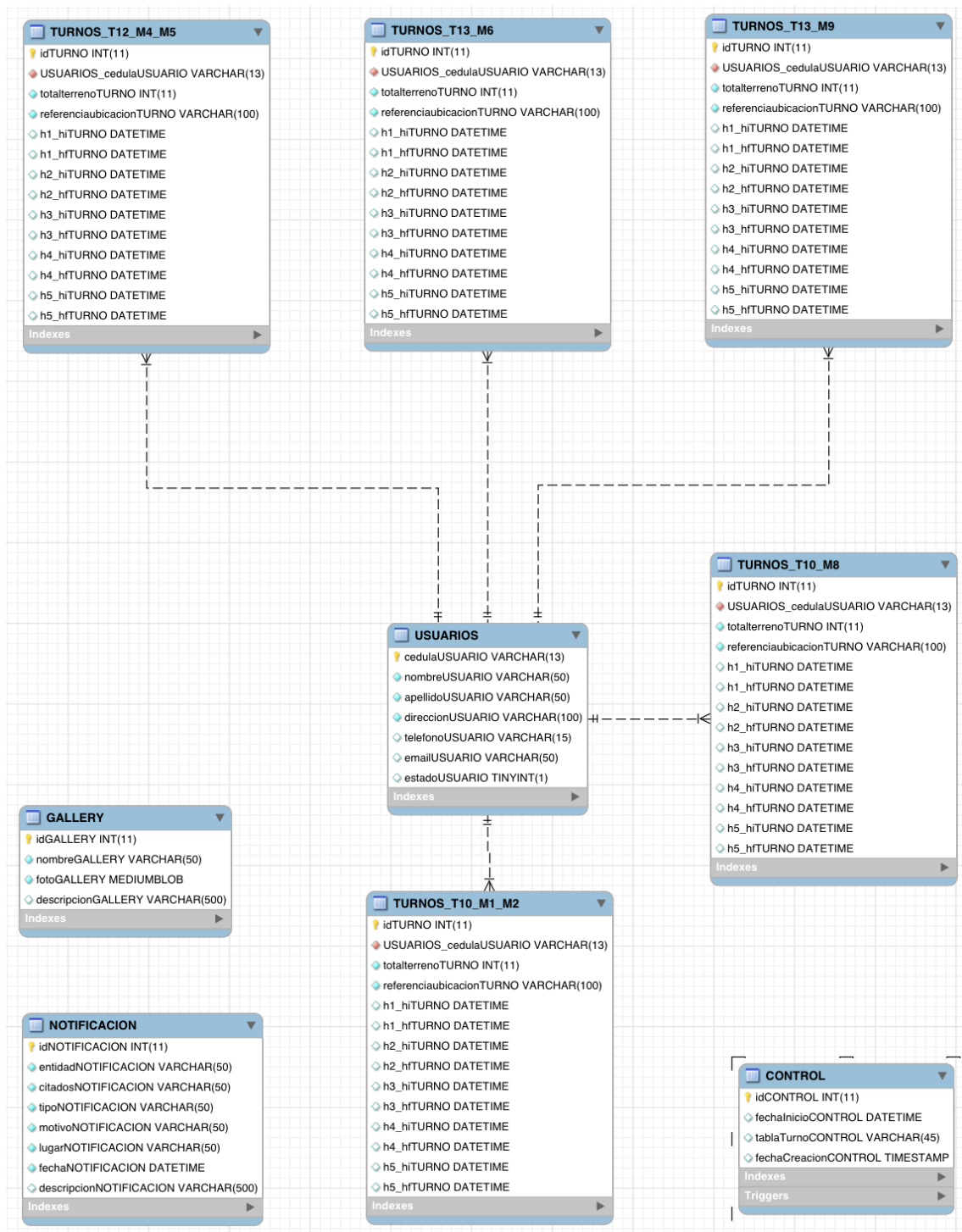


Figura 4.7: Diseño de la base de datos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.5.1.3. Tarea 3 - Codificación y desarrollo de interfaz para ingreso a pantalla de consultas de turnos de riego de aplicación móvil

Interfaz

Para el cumplimiento de esta tarea se empieza el uso de la herramienta de desarrollo de aplicaciones móviles para Android llamado Android Studio, se ha agregando varios componentes que nos da como resultado la siguiente pantalla. (Ver figura: 4.8)

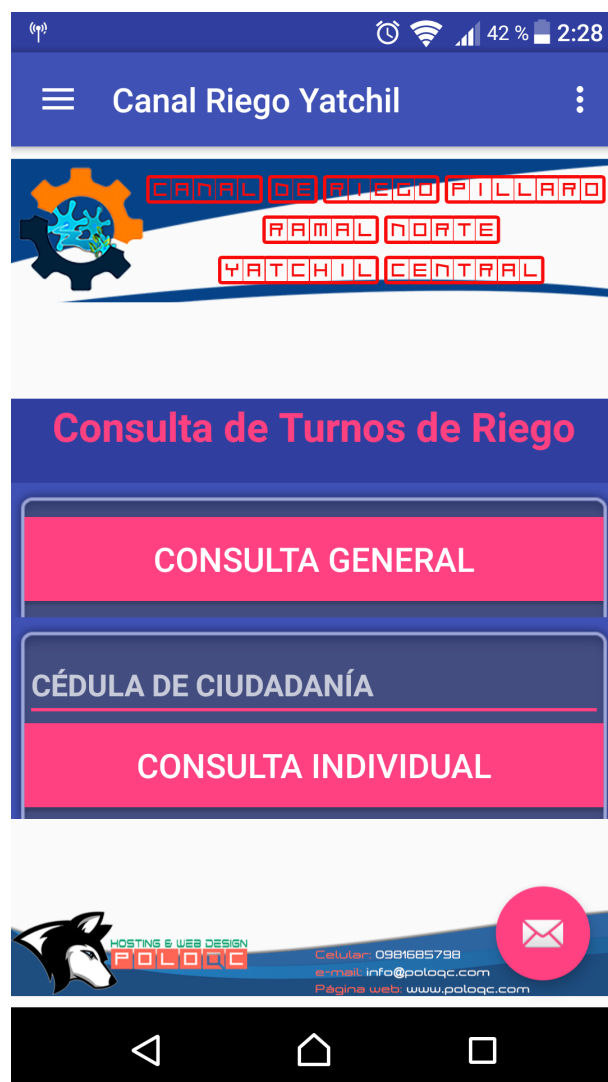


Figura 4.8: Pantalla de inicio de la aplicación móvil
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Manejo de la clase AsyncTask

Nos permite el manejo de eventos que no están coordinados en la interfaz de usuario, aísla los procesos que interrumpen el hilo de proceso que muestra la interfaz de

usuario en un subproceso que ejecuta en segundo plano y luego publica el resultado en la interfaz. Para utilizarlo debemos aplicar el método de devolución de llamada `doInBackground()`, éste método se ejecuta en un grupo de subprocesos de fondo. Para actualizar la interfaz de usuario, se debe implementar el método `onPostExecute()` que proporciona el resultado de `doInBackground()` y éste es ejecutado en el hilo de interfaz de usuario. [32]

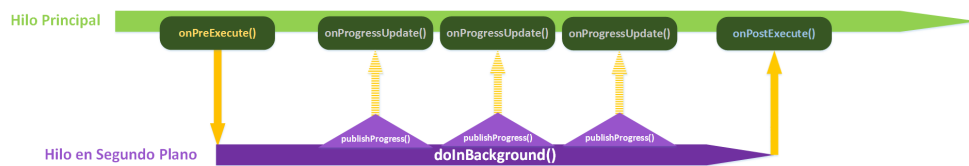


Figura 4.9: Ciclo de vida clase AsyncTask
Elaborado por: jarroba.com
Fuente: [32]

Para pruebas posteriores previo a la creación de código, otorgamos permiso de acceso a Internet en `AndroidManifest.xml`

```
1 // AndroidManifest.xml
2 <uses-permission android:name="android.permission.Internet" />
```

Código usado para verificar existencia de usuario en la base de datos haciendo uso de la clase AsyncTask

Se realiza este proceso para determinar si un usuario que esta haciendo uso de la aplicación web y quiere realizar una consulta, esta previamente registrado en la tabla de usuarios. Si no esta registrado, obviará el resto de procesos que se debe realizar para mostrar los turnos de un usuario específico.

Método `doInBackground`

Hilo secundario que se ejecuta en segundo plano para obtener los datos mediante un objeto JSON (JavaScript Object Notation) que acceda a los atributos (campos) del objeto.

```
1 public class ObtenerWebService extends AsyncTask<String, Void, ArrayList<
    String>>
2 {
3
4     @Override
5     protected ArrayList<String> doInBackground(String... strings) {
6         String cadena = strings[0];
```

```

7 // Url de donde queremos obtener información
8         URL url = null;
9         Boolean retorno = null;
10 // consulta por id
11         if(strings[1]=="2")
12             {
13                 try
14                     {
15                         url = new URL(cadena);
16 // Abrir la conexión
17                         HttpURLConnection connection = (HttpURLConnection)
18                             url.openConnection();
19                         connection.setRequestProperty("User-Agent", "
20                             Mozilla/5.0" +
21                             " (Linux; Android 1.5; es-ES) Ejemplo_HTTP"
22                             );
23                         int respuesta = connection.getResponseCode();
24                         StringBuilder result = new StringBuilder();
25 // Verificar si tenemos acceso a Internet
26                         if (respuesta == HttpURLConnection.HTTP_OK) {
27
28                             Log.d("Respuesta", "Conexion Ok Polo dos");
29 // Preparo la cadena de entrada
30                             InputStream in = new BufferedInputStream(
31                                 connection.getInputStream());
32 // Introduzco la cadena en un BufferedReader
33                             BufferedReader reader = new BufferedReader(new
34                                 InputStreamReader(in));
35
36                             // El siguiente proceso lo hago porque el
37                             // JSONObject necesita un String y tengo
38                             // que transformar el BufferedReader a String.
39                             // Esto lo hago a traves de un
40                             // StringBuilder.
41
42                             String line;
43                             while ((line = reader.readLine()) != null) {
44 // Paso toda la entrada al StringBuilder
45                                 result.append(line);
46                             }
47
48                             //Creamos un objeto JSONObject para poder
49                             // acceder a los atributos (campos) del objeto
50                             .
51 // Creo un JSONObject a partir del StringBuilder pasado a cadena

```

```

43             JSONObject respuestaJSON = new JSONObject(result
                    .toString());
44 // Accedemos al vector de resultados
45 // Estado es el nombre del campo en el JSON
46             String resultJSON = respuestaJSON.getString("
                    estado");
47 // Verifico por consola si tengo resultado de JSON
48             Log.d("Respuesta", "resulJSON_Ok_Polo_dos");
49             ArrayList<String> allUsuario = new ArrayList
                    <>();
50             if (resultJSON.length() > 0){
51 // Estado es el nombre del campo en el JSON
52                 JSONArray turnosJSON = respuestaJSON.
                    getJSONArray("turnos");
53 // Revisar indice de objetos
54                 allUsuario.add(turnosJSON.getJSONObject(0).
                    getString("cedulaUSUARIO"));
55                 allUsuario.add(turnosJSON.getJSONObject(0).
                    getString("nombre"));
56
57                 return allUsuario;
58             }
59             else
60             {
61 // No realizo nada
62             }
63         }
64     } catch (MalformedURLException e) {
65         e.printStackTrace();
66     } catch (IOException e) {
67         e.printStackTrace();
68     } catch (JSONException e) {
69         e.printStackTrace();
70     }
71 }
72 return null;
73 }

```

Método onPostExecute

En este paso procedemos a llamar al método onPostExecute que de por acabado el hilo ejecutado en segundo plano.

```

1  @Override
2      protected void onPostExecute(ArrayList<String> s) {
3      super.onPostExecute(s);

```

```

4          try {
5              cedula = s.get(0);
6              nombre = s.get(1);
7              verificarUsuarioExistente();
8          } catch (Exception e)
9          {
10             Toast.makeText(getApplicationContext(), "Usuario no
                registrado", Toast.LENGTH_SHORT).show();
11         }
12     }

```

Método que verifica si el usuario ingresado existe en la base de datos

Se verifica que la variable global cédula obtenida por el objeto JSON sea el mismo que ingreso el usuario en la caja de texto.

```

1 private void verificarUsuarioExistente()
2 {
3     if (cedula.equals(editTextCedula.getText().toString()))
4     {
5         // Parametros actividad actual, actividad que llamamos
6         Intent intent = new Intent(this, ConsultaIndividualActivity
            .class);
7
8         // pasar informacion a la activity
9         intent.putExtra("cedula", cedula);
10        intent.putExtra("nombre", nombre);
11
12        // Mostrar actividad
13        startActivity(intent);
14    }
15    else
16    {
17
18        Toast.makeText(this, "Usuario no registrado", Toast.
            LENGTH_SHORT).show();
19
20    }
21 }

```

Ahora necesitamos los web service que se crea con funcionalidades que permitan obtener datos actualizados en tiempo real. El hecho de que sea dinámico incorpora el uso de un lenguaje web para la gestión HTTP que en este caso será Php.

Código PHP para proveer las constantes de conexión a la base de datos Mysql

```

1 <?php
2 // Nombre del host
3 define("HOSTNAME", "localhost");
4 // Nombre de la base de datos
5 define("DATABASE", "poloqc_canalriegoYatchil");
6 // Nombre del usuario
7 define("USERNAME", "poloqc_cry");
8 // Nombre de la contraseña
9 define("PASSWORD", "*****");
10 ?>

```

Código PHP de la Clase que envuelve una instancia de la clase PDO (PHP Data Objects, Objeto de Datos PHP) para el manejo de la base de datos

```

1 <?php
2 require_once 'mysql_login.php';
3 class Database
4 {
5     /**
6      * Única instancia de la clase
7      */
8     private static $db = null;
9
10    /**
11     * Instancia de PDO
12     */
13    private static $pdo;
14
15    final private function __construct()
16    {
17        try {
18            // Crear nueva conexión PDO
19            self::getDb();
20        } catch (PDOException $e) {
21            // Manejo de excepciones
22        }
23    }
24
25    /**
26     * Retorna en la única instancia de la clase
27     * @return Database|null
28     */
29    public static function getInstance()
30    {

```



```

31         if (self::$db === null) {
32             self::$db = new self();
33         }
34         return self::$db;
35     }
36
37     /**
38      * Crear una nueva conexión PDO basada
39      * en los datos de conexión
40      * @return PDO Objeto PDO
41      */
42     public function getDb()
43     {
44         if (self::$pdo === null) {
45             self::$pdo = new PDO(
46                 'mysql:dbname=' . DATABASE .
47                 ';host=' . HOSTNAME .
48                 ';port:63343;', // Eliminar este elemento si se usa una
49                             instalación por defecto
49                 USERNAME,
50                 PASSWORD,
51                 array(PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8")
52             );
53
54             // Habilitar excepciones
55             self::$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::
56                 ERRMODE_EXCEPTION);
57
58             return self::$pdo;
59         }
60
61         /**
62          * Evita la clonación del objeto
63          */
64         final protected function __clone()
65         {
66         }
67
68         function __destruct()
69         {
70             self::$pdo = null;
71         }
72     }
73     ?>

```

Código PHP de la Clase que representa la estructura de usuarios en la base de datos

```
1 <?php
2 require 'Database.php';
3 class USUARIOS
4 {
5     function __construct()
6     {
7     }
8
9     /**
10      * Retorna en la fila especificada de la tabla 'TURNOS_T10_M1_M2'
11      *
12      * @param $USUARIOS_cedulaUSUARIO Identificador del registro
13      * @return array Datos del registro
14      */
15     public static function getAll()
16     {
17         $consulta = "SELECT _cedulaUSUARIO , _CONCAT(apellidoUSUARIO , _'_' ,
18             nombreUSUARIO) _AS _nombre , _direccionUSUARIO , _telefonoUSUARIO
19             , _emailUSUARIO , _estadoUSUARIO _FROM _USUARIOS _ORDER _BY _2 ";
20         try {
21             // Preparar sentencia
22             $comando = Database::getInstance()->getDb()->prepare(
23                 $consulta);
24             // Ejecutar sentencia preparada
25             $comando->execute();
26
27             return $comando->fetchAll(PDO::FETCH_ASSOC);
28
29         } catch (PDOException $e) {
30             return false;
31         }
32     }
33
34     /**
35      * Obtiene los campos de un Alumno con un identificador
36      * determinado
37      *
38      * @param $idAlumno Identificador del alumno
39      * @return mixed
40      */
41     public static function getId($USUARIOS_cedulaUSUARIO)
42     {
43     }
```

```

40      // Consulta de la tabla Alumnos
41      $consulta = "SELECT _cedulaUSUARIO, _CONCAT(apellidoUSUARIO, _' _',
              nombreUSUARIO) _AS _nombre, _direccionUSUARIO, _telefonoUSUARIO
              , _emailUSUARIO, _estadoUSUARIO _FROM _USUARIOS _WHERE _
              cedulaUSUARIO _= _? ";

42
43      try {
44          // Preparar sentencia
45          $comando = Database::getInstance()->getDb()->prepare(
              $consulta);
46          // Ejecutar sentencia preparada
47          $comando->execute(array($USUARIOS_cedulaUSUARIO));
48          // Capturar primera fila del resultado
49          $row = $comando->fetchAll(PDO::FETCH_ASSOC);
50          return $row;
51
52      } catch (PDOException $e) {
53          // Aquí puedes clasificar el error dependiendo de la
              excepción
54          // para presentarlo en la respuesta Json
55          return false;
56      }
57  }
58
59
60 }
61
62 ?>

```

Código PHP de la Clase que obtiene el detalle de un alumno especificado por su identificador "idalumno"

```

1  <?php
2
3  require 'USUARIOS.php';
4
5  if ($_SERVER['REQUEST_METHOD'] == 'GET') {
6
7      if (isset($_GET['USUARIOS_cedulaUSUARIO'])) {
8
9          // Obtener parámetro idalumno
10         $parametro = $_GET['USUARIOS_cedulaUSUARIO'];
11
12         // Tratar retorno
13         $turnos = USUARIOS::getById($parametro);

```

```

14
15
16     if ($turnos) {
17
18         $datos["estado"] = 1;
19         $datos["turnos"] = $turnos;
20         // Enviar objeto json del alumno
21         print json_encode($datos);
22     } else {
23         // Enviar respuesta de error general
24         print json_encode(
25             array(
26                 'estado' => '2',
27                 'mensaje' => 'No se obtuvo el registro'
28             )
29         );
30     }
31
32 } else {
33     // Enviar respuesta de error
34     print json_encode(
35         array(
36             'estado' => '3',
37             'mensaje' => 'Se necesita un identificador'
38         )
39     );
40 }
41 }

```

4.5.2. Iteración 2

4.5.2.1. Tarea 4 - Desarrollo de web service que permita tomar registros desde la base de datos

Todos los web service que necesitamos para mostrar los registros de la base de datos, se realiza de manera similar a los obtenidos para la tabla usuario en los puntos anteriores. Por esta razón se omite mencionar extractos de estos códigos.

4.5.2.2. Tarea 5 - Desarrollo de la interfaz con los componentes necesarios para mostrar los turnos de riego

Se aplica el uso del componente TabHost que permite navegar a travez de las diferentes tomas y módulos que el usuario puede consultar. Otro componente

importante es el FloatingActionMenu que ejecuta las acciones principales de la aplicación.

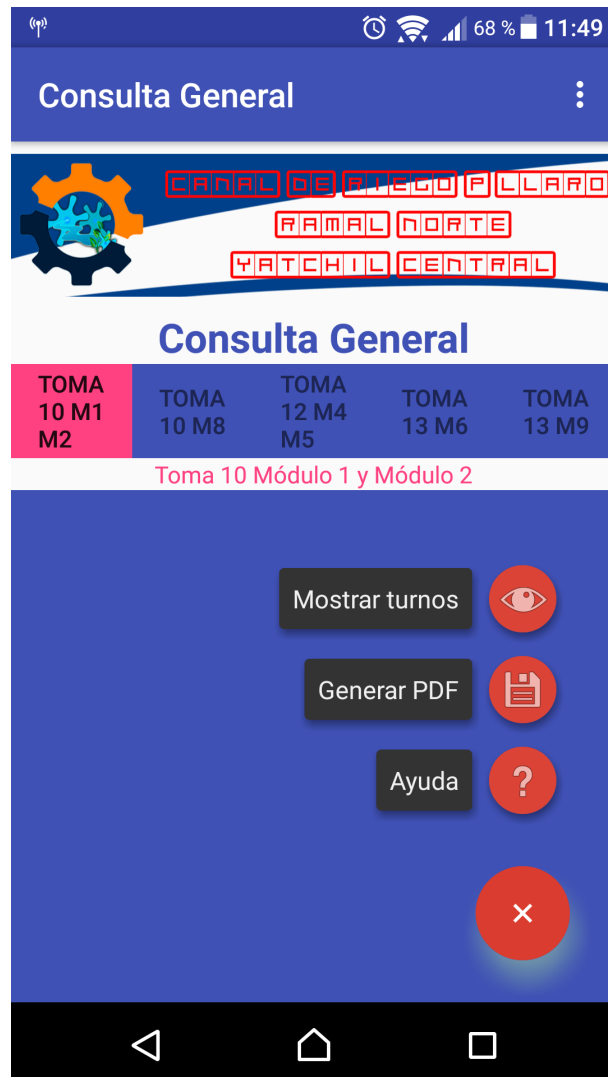


Figura 4.10: Pantalla de consulta de la aplicación móvil
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.5.3. Iteración 3

4.5.3.1. Tarea 6 - Usar librería iTextG para generar archivo PDF

Se agrega la dependencia en build.gradle y se otorga permisos de escritura de los dispositivos de almacenamiento en el AndroidManifest.xml

```
1 // Buid.gradle
2 compile files('libs/5.5.5/itextg-5.5.5.jar')
3
4 // AndroidManifest.xml
```

```

5 <uses-permission Android:name="Android.permission.READ_EXTERNAL_STORAGE
   " />
6 <uses-permission Android:name="Android.permission.
   WRITE_EXTERNAL_STORAGE" />

```

Ahora creamos la clase FileOperations que permita recibir parámetros necesarios para crear el archivo PDF y posteriormente mostrar al usuario.

```

1 public class FileOperations {
2     public FileOperations() {
3     }
4     public Boolean write(String fname, String fcontent) {
5         try {
6             String fpath = "/sdcard/" + fname + ".pdf";
7             File file = new File(fpath);
8             // Si el archivo no existe lo creamos
9             if (!file.exists()) {
10                 file.createNewFile();
11             }
12             // step 1
13             Document document = new Document(PageSize.A4);
14             // step 2
15             PdfWriter.getInstance(document,
16                 new FileOutputStream(file.getAbsoluteFile()));
17             // step 3
18             document.open();
19             // step 4
20
21             // Solo puedo alinear al parrafo
22             Paragraph par1 = new Paragraph("Canal de Riego Píllaro
23                 Ramal Norte Yatchil Central\n\n",
24                 FontFactory.getFont("arial", // fuente
25                     34, // tamaño
26                     Font.BOLD, // estilo
27                     BaseColor.ORANGE)); // color
28             par1.setAlignment(Element.ALIGN_CENTER);
29             document.add(par1);
30             // Solo puedo alineat al parrafo
31             Paragraph par2 = new Paragraph(fname + "\n\n",
32                 FontFactory.getFont("arial", // fuente
33                     34, // tamaño
34                     Font.BOLD, // estilo
35                     BaseColor.BLUE));
36             par2.setAlignment(Element.ALIGN_CENTER);
37             document.add(par2);

```

```

37         try
38         {
39             String url1 = "http://www.poloqc.com/wp-content/uploads
               /2016/10/cry.png";
40             Image foto = Image.getInstance(new URL(url1));
41             foto.scaleToFit(100, 100);
42             foto.setAlignment(Chunk.ALIGN_MIDDLE);
43             document.add(foto);
44         }
45         catch ( Exception e )
46         {
47             e.printStackTrace();
48         }
49         document.add(new Paragraph(fcontent , FontFactory.getFont("
               arial",    // fuente
50                               20)));
51
52         // step 5
53         document.close();
54         return true;
55     } catch (IOException e) {
56         e.printStackTrace();
57         return false;
58     } catch (DocumentException e) {
59         // TODO Auto-generated catch block
60         e.printStackTrace();
61         return false;
62     }
63 }
64 public void MostrarPDF(String fname, Context context)
65 {
66     String pdfFile = "/sdcard/" + fname + ".pdf";
67     File filepdf = new File(pdfFile);
68     Intent intent = new Intent(Intent.ACTION_VIEW);
69     intent.setDataAndType(Uri.fromFile(filepdf), "application/pdf");
70     intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
71     intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
72     try {
73         context.startActivity(intent);
74
75     } catch ( ActivityNotFoundException e)
76     {
77         Toast.makeText(null, "No tiene una aplicación para abrir
               este tipo de archivo", Toast.LENGTH_LONG).show();
78

```

```

79         }
80     }
81 }

```

Se hace uso de la clase FileOperation para enviar los parámetros que escriban en el nuevo archivo PDF que se creará.

```

1  // Creamos objeto de clase que crea pdf
2      FileOperations fop = new FileOperations();
3
4      // switch para comparar segun pestaña actual
5      if(idTbH == 0) {
6
7          String filecontent = ParaPDFT10M1M2;
8          // TODO Auto-generated method stub
9
10         String filename = textViewt10m1m2.getText().
            toString();
11         fop.write(filename, filecontent);
12         if (fop.write(filename, filecontent)) {
13             Toast.makeText(getApplicationContext(),
14                 filename + ".pdf creado", Toast.
15                     LENGTH_SHORT)
16                 .show();
17         } else {
18             Toast.makeText(getApplicationContext(), "
19                 Error interno, intento de nuevo",
20                 Toast.LENGTH_SHORT).show();
21         }
22         fop.MostrarPDF(filename, this);
23         break;
24     }

```

En el método onPostExecute de las clases de consulta, creamos la cadena que contenga los datos de interés del usuario que se enviara a la clase FileOperation para que cree el archivo PDF. Los datos que se observa tomar a continuación, son recibidos del método doInBackground de las clases de consulta,

```

1  ParaPDFT10M1M2 = ParaPDFT10M1M2 + ("Número de Turno:\n" + s.get(i).
    getId() + "\n") +
2      ("Nombre:\n" + s.get(i).getNombre() + "\n") +
3      ("Total terreno:\n" + s.get(i).
    getTotalterreno() + "\n") +

```



```

4      ("Total_horas:\n" + s.get(i).
      getTotalhoras() + "\n") +
5      ("Referencia:\n" + s.get(i).
      getReferencia() + "\n\n") +
6      ("Primer_turno_inicial:\n" + s.get(i).
      getH1_hiTURNO() + "\n") +
7      ("Primer_turno_final:\n" + s.get(i).
      getH1_hfTURNO() + "\n\n") +
8      ("Segundo_turno_inicial:\n" + s.get(i).
      getH2_hiTURNO() + "\n") +
9      ("Segundo_turno_final:\n" + s.get(i).
      getH2_hfTURNO() + "\n\n") +
10     ("Tercer_turno_inicial:\n" + s.get(i).
      getH3_hiTURNO() + "\n") +
11     ("Tercer_turno_final:\n" + s.get(i).
      getH3_hfTURNO() + "\n\n") +
12     ("Cuarto_turno_inicial:\n" + s.get(i).
      getH4_hiTURNO() + "\n") +
13     ("Cuarto_turno_final:\n" + s.get(i).
      getH4_hfTURNO() + "\n\n") +
14     ("Quinto_turno_inicial:\n" + s.get(i).
      getH5_hiTURNO() + "\n") +
15     ("Quinto_turno_final:\n" + s.get(i).
      getH5_hfTURNO() + "\n\n") +
16     ("=====\n\n");
17
18     AATurnosRecividoT10M1M2.notifyDataSetChanged();

```

4.5.4. Iteración 4

4.5.4.1. Tarea 7 - Adquirir herramienta “PHP Generator for MySQL” e investigar funcionamiento

Siguiendo la metodología XP que agiliza procesos a fin de obtener resultados en menor tiempo, se ha procedido a comprar la herramienta PHP Generator for MySQL, esta herramienta permite crear rápidamente un sitio web desde su base de datos haciendo uso de conocimientos básicos sobre PHP y JavaScript. La aplicación web creada permite a los usuarios autorizados ver, editar, añadir y eliminar registros de base de datos. Los pasos a seguir para obtener un producto terminado de alta calidad son los siguientes:

- Conectarse a MySQL
- Configurar fuentes de datos

- Configurar comportamiento de los campos (Columnas, eventos, gráficos y propiedades de la página (Validación de datos))
- Configurar apariencia de la página (Esquemas de color, encabezado y pié de página, estilos definidos por el usuario, javascript de usuario y uso de plantillas)
- Configurar seguridad[33]

4.5.4.2. Tarea 8 - Crear pantalla de acceso a la aplicación web

Para este punto se ha utilizado las CSS (Cascading Style Sheets, Hojas de Estilo en Cascada) para crear un diseño moderno que sea intuitivo y despliegue colores ergonomicos.

También se hace uso de las plantillas TPL (Template Procedural Language) que nos va a permitir crear la plantilla de diseño del ingreso a la aplicación web.

```

1 <!DOCTYPE html>
2   <head>
3     <meta charset="UTF-8" />
4     <meta http-equiv="X-UA-Compatible" content="IE=edge ,chrome=1">
5     <title>Canal de Riego Yatchil</title>
6     <meta name="viewport" content="width=device-width , initial -
      scale=1.0">
7     <meta name="description" content="Página de administración del
      Canal de Riego Yatchil" />
8     <meta name="keywords" content="canal , riego , Yatchil , polo ,
      poloqc" />
9     <meta name="author" content="poloqc.com" />
10    <link rel="shortcut icon" href="../favicon.ico">
11    <link rel="stylesheet" type="text/css" href="css/demo.css" />
12    <link rel="stylesheet" type="text/css" href="css/style1.css" />
13    <script type="text/javascript" src="js/modernizr.custom.86080.
      js"></script>
14  </head>
15  <body id="page">
16    <div class="container">
17      <!-- Codrops top bar -->
18      <div class="codrops-top">
19        <a href="http://www.poloqc.com" target="_blank">
20          <strong>&laquo; Powered By: </strong>poloqc.com
21        </a>
22        <span class="right">
23          <a href="https://fisei.uta.edu.ec/" target="_blank">
            >Trabajo de titulación <strong>UTA - FISEI</
            strong></a>

```

```

24         <!--
25         <a href="http://www.uta.edu.ec/v3.2/uta/" target="
           _blank">CC BY-SA 2.0</a>
26         <a href="http://tympanus.net/codrops/2012/01/02/
           fullscreen-background-image-slideshow-with-css3
           /">
27             <strong>Back to the Codrops Article</strong>
28         </a>
29         —>
30     </span>
31     <div class="clr "></div>
32 </div><!--/ Codrops top bar —>
33 <header>
34 <table>
35 <thead>
36
37 <tr>
38
39 <th>
40     <h1>CUIDA <span>EL AGUA...</span></h1>
41     </th>
42     <th>
43     <h2>&nbsp; &nbsp; &nbsp; &nbsp; no dejes que se acabe</
         h2>
44     </th>
45 </tr>
46 </thead>
47 </table>
48 <!--
49     <p class="codrops-demos">
50         <a href="#" class="current-demo">Demo 1</a>
51         <a href="index2.html">Demo 2</a>
52         <a href="index3.html">Demo 3</a>
53         <a href="index4.html">Demo 4</a>
54     </p>
55     —>
56 </header>
57 </div>
58
59 <!-- Empieza la parte de personalizacion de plantilla de login —>
60 {capture assign="ContentBlock"}
61     {* <CustomTemplate> *}
62 <!--
63     <h1>Bienvenidos al Sistema de Consulta de Turnos de Regadio del
         Canal de Riego Yatchil.</h1>

```

```

64
65     —>
66         <!-- <div align="center"> <IMG SRC="imagenes\banner_cry.
           png"; BORDER="0"; HEIGHT="80"; WIDTH="350"; align="
           center">
67         </div>
68     —>
69
70         {* </CustomTemplate> *}
71
72         {$Renderer->Render($LoginControl)}
73
74         {* <CustomTemplate> *}
75     </div>
76
77     <h2> <font color="#2c3e50">Información para ingreso al sistema</
       font> </h2>
78
79     <table class="table table-bordered">
80         <thead>
81             <FONT color="#000000">
82             <tr>
83
84                 <th>
85 <font color="#000000">
86                 Nombre
87 </font>
88                 </th>
89                 <th>
90 <font color="#000000">
91                 Clave
92 </font>
93                 </th>
94                 <th>
95 <font color="#000000">
96                 Descripción
97 </font>
98                 </th>
99             </tr>
100         </FONT>
101         </thead>
102         <tbody>
103         <tr>
104             <td>
105 <font color="#18bc9c">

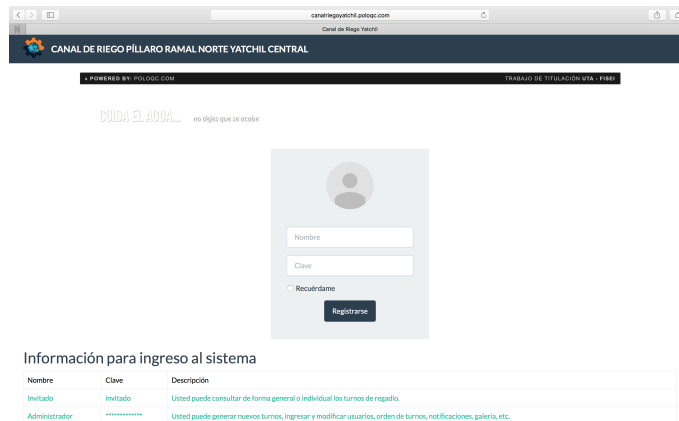
```

```

106             Invitado
107 </font>
108         </td>
109     <td>
110 <font color="#18bc9c">
111         invitado
112
113 </font>
114     </td>
115 <td>
116 <font color="#18bc9c">
117     Usted puede consultar de forma general o individual los
        turnos de regadío.
118
119 </font>
120 </td>
121 </tr>
122 <tr>
123     <td>
124 <font color="#18bc9c">
125         Administrador
126 </font>
127     </td>
128     <td>
129 <font color="#18bc9c">
130         *****
131 </font>
132     </td>
133 <td>
134 <font color="#18bc9c">
135     Usted puede generar nuevos turnos , ingresar y modificar
        usuarios , orden de turnos , notificaciones , galería , etc
        .
136 </font>
137 </td>
138 </tr>
139 <!-- -->
140 </tbody>
141 </table>
142 {* </CustomTemplate> *}
143 {/capture}
144 {include file="common/layout.tpl"}
145
146 </body>
147 </html>

```

página de inicio de sesión de la aplicación web



Nombre	Clave	Descripción
Invitado	Invitado	Usted puede consultar de forma general o individual los turnos de riego.
Administrador	*****	Usted puede generar nuevos turnos, ingresar y modificar usuarios, orden de turnos, notificaciones, galería, etc.

Figura 4.11: Inicio de sesión de la aplicación web
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.5.5. Iteración 5

4.5.5.1. Tarea 9 - Crear módulos de consulta de turnos de riego para la aplicación web

Para el cumplimiento de las tareas referentes a la administración web, se hará uso del software PHP Generator for MySQL, dado que los procesos son los mismos para el mantenimiento de todas las tablas, se describirá solo algunas de estas.

Conectarse a MySQL

Ingresamos todos los datos para que el programa pueda conectar con nuestra base de datos que se encuentra alojado en nuestro servidor anteriormente especificado.

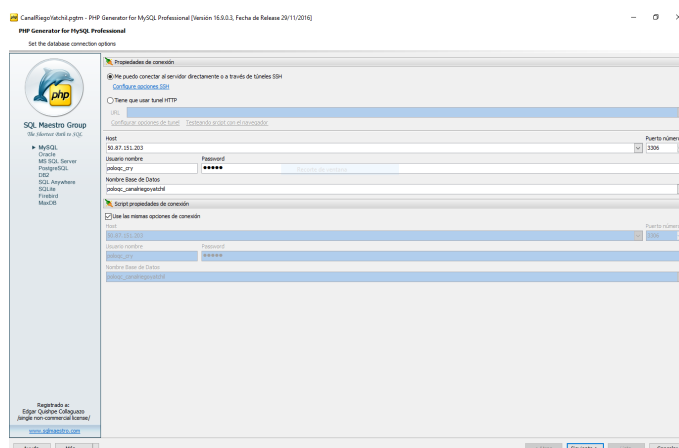
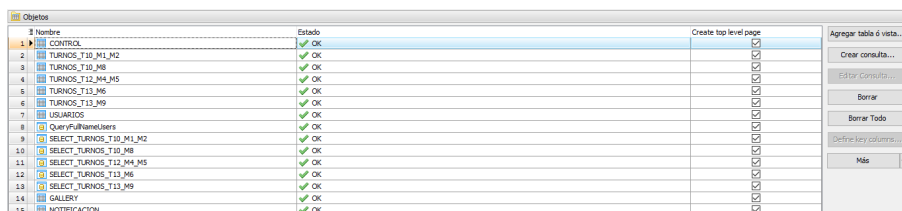


Figura 4.12: Conexión a la base de datos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Configurar fuentes de datos

Seleccionamos todas las tablas que vamos a configurar los campos posteriormente. Estas tablas se configurarán individualmente en los pasos siguientes.



Objetos	Nombre	Estado	Create top level page	Agregar tabla ó vista...
1	CONTROL	OK		
2	TURNOS_T10_M1_M2	OK		
3	TURNOS_T10_M3	OK		
4	TURNOS_T12_M4_M5	OK		
5	TURNOS_T13_M6	OK		
6	TURNOS_T13_M9	OK		
7	USUARIOS	OK		
8	QueryFullnameUsers	OK		
9	SELECT_TURNOS_T10_M1_M2	OK		
10	SELECT_TURNOS_T10_M3	OK		
11	SELECT_TURNOS_T12_M4_M5	OK		
12	SELECT_TURNOS_T13_M6	OK		
13	SELECT_TURNOS_T13_M9	OK		
14	GALLERY	OK		
15	NOTIFICACION	OK		

Figura 4.13: Configurar fuente de datos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Configurar comportamiento de los campos (Columnas, eventos, gráficos y propiedades de la página (Validación de datos))

Seleccionamos una de las tablas que deseamos personalizar y presionamos en editar, este paso es el mas importante de todos ya que se configura campo por campo el comportamiento que tendrá para la visualización, inserción, actualización y eliminación.

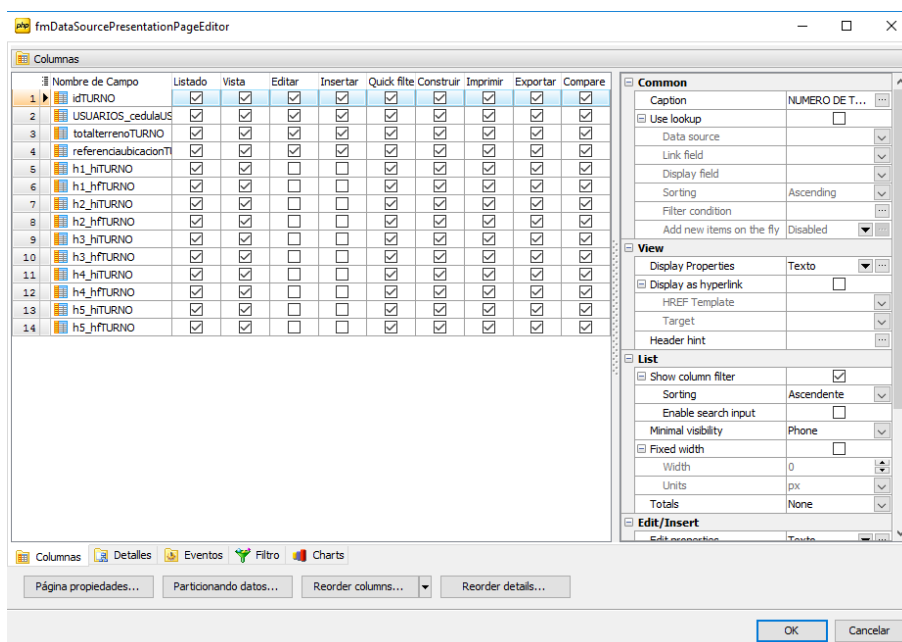


Figura 4.14: Configurar comportamiento de los campos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

También se realiza las validaciones de los campos mediante código de JavaScript o PHP. Ejemplo de validación de cédula con JavaScript.

```

Function OnInsertFormValidate(fieldValues , errorInfo){
    var i;
    var cedula;
    var acumulado;
    cedula=fieldValues [ 'cedulaUSUARIO ' ];
    var instancia;
    acumulado=0;
    for ( i=1;i<=9;i++){
        if ( i%2!=0)
        {
            instancia=cedula.substring(i-1,i)*2;
            if (instancia>9) instancia -=9;
        }
        else instancia=cedula.substring(i-1,i);
        acumulado+=parseInt(instancia);
    }
    while (acumulado>0)
        acumulado-=10;
        if (cedula.substring(9,10)!=(acumulado*-1)){
            alert("Cedula invalida!");
            return false;
        }
        else
            return true;
}

```

Configurar apariencia de la página (Esquemas de color, encabezado y pié de página, estilos definidos por el usuario, JavaScript de usuario y uso de plantillas)

Existe plantillas predeterminadas para dar el estilo a la página pero también podemos personalizar los CSS y los JavaScript. (Ver figura: 4.15)

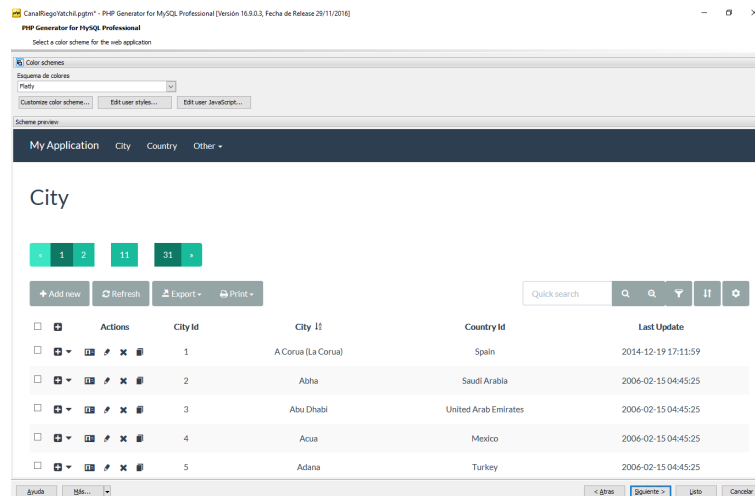


Figura 4.15: Configurar apariencia de la página
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Configurar seguridad

La seguridad que concederemos a nuestra aplicación web es el procedimiento de autorización directamente en el código PHP de la aplicación generada, todas las cuentas de usuario se almacenan en archivos generados .php. Como cualquier operación relacionada con la gestión de usuarios requiere la re-generación y re-carga de la aplicación. Se configurara los dos usuarios invitado y administrador con sus respectivas limitaciones en el caso del usuario invitado. (Ver figura: 4.16)

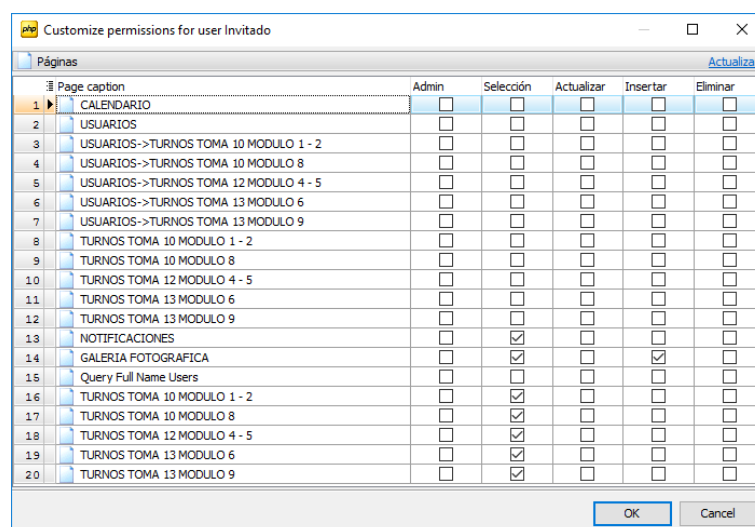


Figura 4.16: Configurar seguridad
Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.5.6. Iteración 6

4.5.6.1. Tarea 10 - Crear módulos de inserción y actualización de usuarios para la aplicación web

Se repite el procedimiento del punto 4.5.1.9 centrando la atención en la tabla de usuarios de la base de datos poloqc_canalriegoYatchil.

4.5.7. Iteración 7

4.5.7.1. Tarea 11 - Crear módulos de inserción y actualización y eliminación de turnos de riego para la aplicación web

Se repite el procedimiento del punto 4.5.1.9 centrando la atención en las cinco tablas de tomas y módulos de la base de datos poloqc_canalriegoYatchil.

4.5.8. Iteración 8

4.5.8.1. Tarea 12 - Crear un trigger de base de datos que genere los turnos de riego de manera automática

Esta tarea de gran importancia, es el motor del sistema ya que se desarrolla el proceso que mediante un trigger de base de datos, automatiza la inserción de fechas y horas de turnos de riego siendo el usuario administrador quien activará el disparador mediante una inserción en la tabla control.

Formado por una sentencia CASE de 5 posibles opciones de turnos que pueda seleccionar el administrador para generar los calendarios y dentro de cada uno, un ciclo repetitivo que se ejecuta hasta que complete el número de registros, el algoritmo va llenando uno a uno los campos de hora inicial y hora final, todo esto se repite para cinco columnas que corresponde a cinco turnos para cada parcela

Mediante la representación gráfica del algoritmo (Ver figura: 4.17), se detalla el trigger de automatización de calendarios.



Figura 4.17: Algoritmo de calendarización
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Aquí se muestra el extracto del código del primer bloque del algoritmo de calendarización generado mediante un trigger de base de datos.

```

1 DELIMITER //
2 CREATE TRIGGER DISPARA_INSERT_CONTROL
3 AFTER INSERT ON poloqc_poloqc_canalriegoYatchil.CONTROL
4 FOR EACH ROW
5 BEGIN
6
7     -- Variable para Tomar fecha ingresada en control
8     declare FECHA_CONTROL datetime;
9     -- Variable para Tomar nombre de tabla ingresada en control
10    declare TABLA_CONTROL varchar(50);
11    -- Variable para mostrar mensaje de error
12    declare msg varchar(255);
13    -- Variable para guardar total de elementos
14    declare TOTAL_ELEMENTOS int;
15    -- Variable para calcular horas correspondientes a cada turno
16    declare TOTAL_HORAS int;
17    -- Variable para iniciar ciclo totalFACTURA
18    DECLARE i int;
19    -- Variable para seleccionar horas inicio turno
20    DECLARE HORA_INICIO datetime;
21    -- Variable para seleccionar horas fin turno
22    DECLARE HORA_FIN datetime;
23
24    -- Seleccionar fecha ingresada para inicio de horario
25    select fechaInicioCONTROL
26    INTO FECHA_CONTROL
27    FROM poloqc_canalriegoYatchil.CONTROL
28    WHERE idCONTROL = new.idCONTROL;
29
30    -- Seleccionar tabla ingresada
31    select tablaTurnoCONTROL
32    INTO TABLA_CONTROL
33    FROM poloqc_canalriegoYatchil.CONTROL
34    WHERE idCONTROL = new.idCONTROL;
35
36    -- Actualiza primer campo de turnos
37    CASE TABLA_CONTROL
38
39    -- Turno TURNOS_T10_M1_M2
40    WHEN "TURNOS_T10_M1_M2"
41    THEN
42

```

```

43          — Llenado de primera columna
44
45      — Inicio de turno
46          update TURNOS_T10_M1_M2
47          SET h1_hiTURNO = FECHA_CONTROL
48          WHERE idTURNO = 1;
49
50      — Seleccionamos total de elementos para limitar el ciclo de
      todas las columnas
51      SELECT count(idTURNO)
52      INTO TOTAL_ELEMENTOS
53      FROM TURNOS_T10_M1_M2;
54
55      — Primer ciclo
56      set i=1;
57      WHILE i <= TOTAL_ELEMENTOS
58      Do
59          BEGIN
60
61              — Seleccionar Hora inicial
62              SELECT h1_hiTURNO INTO HORA_INICIO FROM
                  TURNOS_T10_M1_M2 WHERE idTURNO = i;
63              — Seleccionamos total area de terreno para
                  poder calcular horas totales
64              SELECT (totalterrenoTURNO * 2) INTO TOTAL_HORAS
                  FROM TURNOS_T10_M1_M2 WHERE idTURNO = i;
65              — Actualizamos campo de hora final
66              update TURNOS_T10_M1_M2
67              SET h1_hfTURNO = DATE_ADD(HORA_INICIO, INTERVAL
                  TOTAL_HORAS hour)
68              WHERE idTURNO = i;
69              — Seleccionamos Hora final
70              SELECT h1_hfTURNO INTO HORA_FIN FROM
                  TURNOS_T10_M1_M2 WHERE idTURNO = i;
71              — Actualizamos campo de hora inicial de otra fila
72              update TURNOS_T10_M1_M2
73              SET h1_hiTURNO = HORA_FIN
74              WHERE idTURNO = i + 1;
75
76          set i = i + 1;
77      end;
78  end WHILE;
79
80      — Llenado de segunda columna
81

```

```

82      — Inicio de turno
83      — Selección de fecha final de la ultima fila de la
      primera columna
84      select h1_hfTURNO
85      INTO FECHA_CONTROL
86      FROM poloqc_canalriegoYatchil.TURNOS_T10_M1_M2
87      WHERE idTURNO = TOTAL_ELEMENTOS;
88
89      — Hora inicial de segunda columna
90      update TURNOS_T10_M1_M2
91          SET h2_hiTURNO = FECHA_CONTROL
92          WHERE idTURNO = 1;
93
94      — Segundo ciclo
95          set i=1;
96      WHILE i <= TOTAL_ELEMENTOS
97      Do
98          BEGIN
99
100              — Seleccionar Hora inicial
101              SELECT h2_hiTURNO INTO HORA_INICIO FROM
              TURNOS_T10_M1_M2 WHERE idTURNO = i;
102              — Seleccionamos total area de terreno para
              poder calcular horas totales
103              SELECT (totalterrenoTURNO * 2) INTO TOTAL_HORAS
              FROM TURNOS_T10_M1_M2 WHERE idTURNO = i;
104              — Actualizamos campo de hora final
105              update TURNOS_T10_M1_M2
106                  SET h2_hfTURNO = DATE_ADD(HORA_INICIO, INTERVAL
                  TOTAL_HORAS hour)
107                  WHERE idTURNO = i;
108              — Seleccionamos Hora final
109              SELECT h2_hfTURNO INTO HORA_FIN FROM
              TURNOS_T10_M1_M2 WHERE idTURNO = i;
110              — Actualizamos campo de hora inicial de otra fila
111              update TURNOS_T10_M1_M2
112                  SET h2_hiTURNO = HORA_FIN
113                  WHERE idTURNO = i + 1;
114
115          set i = i + 1;
116      end;
117      end WHILE;
118
119      — Llenado de tercera columna
120

```

```

121      — Inicio de turno
122      — Selección de fecha final de la ultima fila de la
          primera columna
123      select h2_hfTURNO
124      INTO FECHA_CONTROL
125      FROM poloqc_canalriegoYatchil.TURNOS_T10_M1_M2
126      WHERE idTURNO = TOTAL_ELEMENTOS;
127
128      — Hora inicial de segunda columna
129      update TURNOS_T10_M1_M2
130          SET h3_hiTURNO = FECHA_CONTROL
131          WHERE idTURNO = 1;
132
133      — Tercer ciclo
134          set i=1;
135      WHILE i <= TOTAL_ELEMENTOS
136      Do
137          BEGIN
138
139              — Seleccionar Hora inicial
140              SELECT h3_hiTURNO INTO HORA_INICIO FROM
                  TURNOS_T10_M1_M2 WHERE idTURNO = i;
141              — Seleccionamos total area de terreno para
                  poder calcular horas totales
142              SELECT (totalterrenoTURNO * 2) INTO TOTAL_HORAS
                  FROM TURNOS_T10_M1_M2 WHERE idTURNO = i;
143              — Actualizamos campo de hora final
144              update TURNOS_T10_M1_M2
145                  SET h3_hfTURNO = DATE_ADD(HORA_INICIO, INTERVAL
                      TOTAL_HORAS hour)
146                  WHERE idTURNO = i;
147              — Seleccionamos Hora final
148              SELECT h3_hfTURNO INTO HORA_FIN FROM
                  TURNOS_T10_M1_M2 WHERE idTURNO = i;
149              — Actualizamos campo de hora inicial de otra fila
150              update TURNOS_T10_M1_M2
151                  SET h3_hiTURNO = HORA_FIN
152                  WHERE idTURNO = i + 1;
153
154              set i = i + 1;
155          end;
156      end WHILE;
157
158      — Llenado de cuarto columna
159

```

```

160      -- Inicio de turno
161      -- Selección de fecha final de la ultima fila de la
           primera columna
162      select h3_hfTURNO
163      INTO FECHA_CONTROL
164      FROM poloqc_canalriegoYatchil.TURNOS_T10_M1_M2
165      WHERE idTURNO = TOTAL_ELEMENTOS;
166
167      -- Hora inicial de segunda columna
168      update TURNOS_T10_M1_M2
169          SET h4_hiTURNO = FECHA_CONTROL
170          WHERE idTURNO = 1;
171
172      -- Cuarto ciclo
173          set i=1;
174      WHILE i <= TOTAL_ELEMENTOS
175      Do
176          BEGIN
177
178              -- Seleccionar Hora inicial
179              SELECT h4_hiTURNO INTO HORA_INICIO FROM
                   TURNOS_T10_M1_M2 WHERE idTURNO = i;
180              -- Seleccionamos total area de terreno para
                   poder calcular horas totales
181              SELECT (totalterrenoTURNO * 2) INTO TOTAL_HORAS
                   FROM TURNOS_T10_M1_M2 WHERE idTURNO = i;
182      -- Actualizamos campo de hora final
183      update TURNOS_T10_M1_M2
184          SET h4_hfTURNO = DATE_ADD(HORA_INICIO, INTERVAL
                   TOTAL_HORAS hour)
185          WHERE idTURNO = i;
186      -- Seleccionamos Hora final
187      SELECT h4_hfTURNO INTO HORA_FIN FROM
                   TURNOS_T10_M1_M2 WHERE idTURNO = i;
188      -- Actualizamos campo de hora inicial de otra fila
189      update TURNOS_T10_M1_M2
190          SET h4_hiTURNO = HORA_FIN
191          WHERE idTURNO = i + 1;
192
193      set i = i + 1;
194      end;
195      end WHILE;
196
197      -- Llenado de Quinta columna
198

```



```

199      — Inicio de turno
200      — Selección de fecha final de la ultima fila de la
          primera columna
201      select h4_hfTURNO
202      INTO FECHA_CONTROL
203      FROM poloqc_canalriegoYatchil.TURNOS_T10_M1_M2
204      WHERE idTURNO = TOTAL_ELEMENTOS;
205
206      — Hora inicial de segunda columna
207      update TURNOS_T10_M1_M2
208          SET h5_hiTURNO = FECHA_CONTROL
209          WHERE idTURNO = 1;
210
211      — Quinto ciclo
212          set i=1;
213      WHILE i <= TOTAL_ELEMENTOS
214      Do
215          BEGIN
216
217              — Seleccionar Hora inicial
218              SELECT h5_hiTURNO INTO HORA_INICIO FROM
                  TURNOS_T10_M1_M2 WHERE idTURNO = i;
219              — Seleccionamos total area de terreno para
                  poder calcular horas totales
220              SELECT (totalterrenoTURNO * 2) INTO TOTAL_HORAS
                  FROM TURNOS_T10_M1_M2 WHERE idTURNO = i;
221              — Actualizamos campo de hora final
222              update TURNOS_T10_M1_M2
223                  SET h5_hfTURNO = DATE_ADD(HORA_INICIO, INTERVAL
                      TOTAL_HORAS hour)
224                  WHERE idTURNO = i;
225              — Seleccionamos Hora final
226              SELECT h5_hfTURNO INTO HORA_FIN FROM
                  TURNOS_T10_M1_M2 WHERE idTURNO = i;
227              — Actualizamos campo de hora inicial de otra fila
228              update TURNOS_T10_M1_M2
229                  SET h5_hiTURNO = HORA_FIN
230                  WHERE idTURNO = i + 1;
231
232          set i = i + 1;
233      end;
234      end WHILE;

```

El extracto de código observado es únicamente para una de cinco tomas, código similar se repite para:

TURNOS_T10_M8
TURNOS_T12_M4_M5
TURNOS_T13_M6
TURNOS_T13_M9

4.5.8.2. Tarea 13 - Crear módulos de inserción y eliminación de generación de calendarios

Se repite el procedimiento del punto 4.5.1.9 centrando la atención en la tabla de control de la base de datos `poloqc_canalriegoYatchil`.

Pantalla para generar calendarios

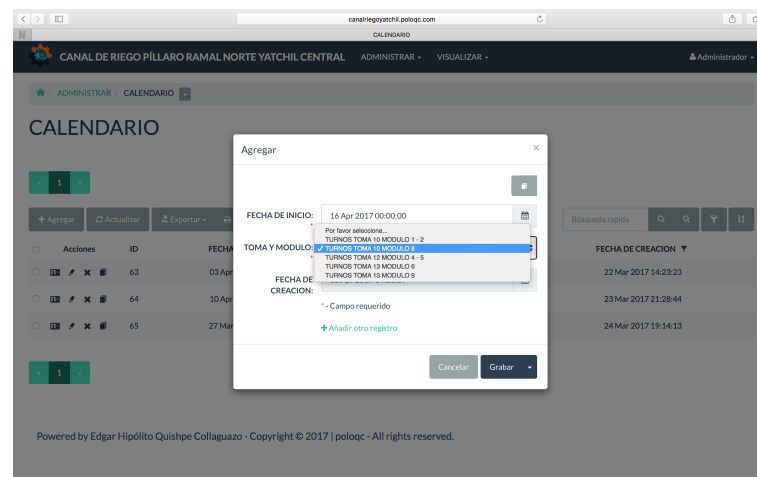


Figura 4.18: Pantalla para generar calendarios
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Para generar el calendario de riego el administrador debe ingresar la fecha de inicio y la tabla que desea llenar los horarios y fechas de turnos de riego en los campos correspondientes y al presionar grabar se guarda este registro y dispara el trigger llenando los campos de horarios en la tabla de turnos con nuevos horarios a partir de la fecha ingresada.

4.6. Pruebas

Pruebas unitarias

Las pruebas unitarias son una de las piedras angulares de XP. Cada historia de usuario deben pasar la prueba unitaria antes de ser liberado o publicado. La prueba de una historia de usuario se realiza al final de cada iteración que a la vez es el

inicio de otro, se procede a ilustrar la prueba unitaria de la iteración número 2 que corresponde a la historia de usuario del mismo número.

Para esta prueba, se ha cargado algunos registros en la base de datos y el usuario a ingresado su cédula de ciudadanía para acceder a la pantalla de consulta, luego se ha situado sobre la pestaña de la toma y módulo de interés, finalmente haciendo uso de los botones desplegables se ha ejecutado la acción de consulta que despliega el siguiente resultado.



Figura 4.19: Prueba iteración N° 2
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Pruebas de aceptación

Las pruebas de aceptación son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja

negra” (“Black box system tests”). Los clientes son responsables de verificar que los resultados de éstas pruebas sean correctos. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información. [34]

Se procede a realizar la prueba de la historia de usuario N° 2.

Prueba de aceptación	
Número: 2	Historia de usuario: 2
Nombre: Consultar turnos en la aplicación móvil.	
Descripción: Mostrar todos los turnos de la toma y modulo que corresponda a la pestaña actual seleccionado, si es una consulta individual, se mostrará todos los turnos de la toma y modulo que corresponda al la pestaña actual seleccionado y que acierten en la condición de ser igual al número de cédula ingresados en la pantalla anterior que describe la historia de usuario 1.	
Condiciones de ejecución: <ul style="list-style-type: none"> • Contar con acceso a Internet. • En el caso de consulta individual, el usuario debe estar registrado para poder acceder a la pantalla de consulta. 	
Entrada: <ol style="list-style-type: none"> I. El usuario ejecuta la aplicación “Canal Riego Yatchil”. II. Si selecciona “CONSULTA GENERAL”, no ingresa ningún parámetro y se concede acceso a la pantalla de consulta. III. Si selecciona “CONSULTA INDIVIDUAL”, debe ingresar el número de cédula de ciudadanía. Si la cédula está correcta se busca si esta registrado en la base de datos, si se encuentra registrado se concederá acceso a la pantalla de consulta, caso contrario aparecerá el siguiente mensaje “Usuario no registrado” y si hay problemas en el parámetro de entrada aparecerá el siguiente mensaje “Campo vacío o Cédula incorrecta”. IV. Ya sea la consulta general o individual, una vez ingresado le lleva a una pantalla similar en ambos casos. En esta pantalla existe cinco pestañas que corresponde a cada toma y módulos de riego, el usuario debe seleccionar el de su interés. V. El usuario ahora debe hacer uso del botón flotante o del menú tradicional de acciones para ejecutar la consulta presionando “Mostrar turnos”. 	
Resultado esperado: Datos mostrados en forma de lista.	
Evaluación de la prueba: Satisfactoria.	

Tabla 4.26: Prueba de aceptación - Consultar turnos en la aplicación móvil

Elaborado por: Edgar Hipólito Quishpe Collaguazo

4.7. Producción

Para acceder al sitio web del canal de riego se deberá ingresar a la siguiente dirección:

<http://www.canalriegoYatchil.poloqc.com/>

La página en construcción nos proporciona un enlace que nos lleva a la pantalla de inicio de sesión. El resto de proceso se detalla en el anexo correspondiente al manual de usuario de la aplicación web.

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. CONCLUSIONES

Se ha logrado levantar toda la información referente al proceso de calendarización de los turnos de riego, esto ha obligado a crear una base de datos con toda la información relevante del usuario, información detallada de los terrenos en orden secuencias que corresponde hacer uso del turno de riego a cada dueño de las propiedades.

Despues del análisis del proceso manual de la calendarización, se ha solucionado el problema de calendarización mediante un proceso automático realizado con código SQL. Esta solución generado mediante un trigger se ejecuta de manera automática cuando el administrador lo desea, si antes tomaba casi una semana realizar los turnos de riego, ahora unicamente se necesita unos pocos minutos para generarlos.

El mantenimiento de la base de datos se realizó mediante una herramienta generador de código PHP llamado "PHP Generator for MySQL", como resultado final del tiempo invertido en el aprendizaje de esta herramienta, el manejo de las funciones básicas para configurar el comportamiento de los campos, botones y otros componentes de la aplicación web, se obtuvo una página de alta calidad que proporciona seguridad y desempeño al cliente.

Android Studio mediante clases para ejecución de trabajos en segundo plano, librerías adicionales para generación de archivos PDF y componentes gráficos, ha permitido llegar a la información contenida en nuestra base de datos y hacerlo publico a través de una interfaz amigable e intuitivo para el usuario que al final puede exportar la información en un archivo PDF a su tarjeta Micro SD.

5.2. RECOMENDACIONES

Se recomienda el uso de la metodología Extreme Programming para el desarrollo de aplicaciones móviles de magnitudes pequeñas y medianas de poco presupuesto y personal.

Para la etapa de producción se recomienda contar con un servicio de hosting garantizado como lo es Hostgator, su amplio equipo de servicio al cliente están capacitados por niveles de complejidad así que si se produce algún problema en la publicación de nuestra página web, podremos buscar solución a través de soporte de primer nivel y si el problema persiste, nos direccionarán a los siguientes niveles con profesionales que atienden a problemas mas complejos.

Se recomienda incluir el módulo de notificaciones en la aplicación móvil para que los usuarios puedan estar al día con la información que proporciona la directiva del canal de riego como mingas, sesiones, comisiones, días de recaudaciones, entre otros.

Para mejorar el manejo de la base de datos mediante la aplicación web, se recomienda realizar un procesos que mediante la función drag and drop cambie el orden de los turnos. Así también una función para agregar un nuevo turno en una posición específica.

Para el desarrollo de este trabajo ha sido imprescindible almacenar información referente a los terrenos de cada uno de los usuarios, por lo que se recomienda hacer uso de estos datos para proporcionar información catastral que determine la extensión geográfica total o parcial de alguna demarcación específica. Estos datos se podrían convertir en un apoyo para determinar el cobro de las imposiciones del municipio o para entidades particulares que requieran este tipo de información.

Bibliografía

- [1] U. y Consultora IDG Connect, “Las tecnologías móviles aumentan la productividad y los ingresos de las empresas.” Disponible en: <http://aplicantes.com/tecnologias-moviles-empresas-productividad-ingresos-2014/>, 2014.
- [2] I. N. de Estadística y Censos, “Estadística de uso de telefonía.” Disponible en: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/Resultados_principales_140515.Tic.pdf, 2013.
- [3] E. N. N. E. Ecuador, “Produbanco - grupo promerica cuenta con el servicio de banca móvil.” Disponible en: <http://www.ekosnegocios.com/negocios/m/verArticulo.aspx?idart=6081&c=1>.
- [4] U. y DIRECCIÓN DE TECNOLOGÍA DE INFORMACIÓN Y COMUNICACIÓN, “Utamovil.” Disponible en: http://www.uta.edu.ec/v2.0/index.php?option=com_content&view=article&id=599&Itemid=172, 2016.
- [5] O. E. A. Mayorga, “Aplicación móvil bajo la plataforma android para la gestión de calificaciones en la unidad educativa “augusto nicolás martínez”.” Disponible en: <http://repositorio.uta.edu.ec/jspui/handle/123456789/10999>, 2015.
- [6] V. C. M. Roberto, “Sistema de gestión para historias clínicas bajo la plataforma android orientado a los médicos del condominio del hospital millennium.” Disponible en: <http://repositorio.uta.edu.ec/jspui/handle/123456789/19372>, 2015.
- [7] D. L. Gamboa Safla, “Aplicación móvil para el control de notas de los estudiantes de la universidad técnica de ambato utilizando la plataforma android.” Disponible en: <http://repositorio.uta.edu.ec/jspui/handle/123456789/8593>, 2015.

- [8] G. Gerbrandy, “Manual-de-riego-parcelario.” Disponible en: <http://www.agricultura.gob.ec/wp-content/uploads/downloads/2015/11/Manual-de-riego-parcelario.pdf>.
- [9] O. de las Naciones Unidas para la Alimentación y la Agricultura, “Uso de agua en riego uso de agua en riego uso de agua en riego.” Disponible en: http://www.fao.org/nr/water/aquastat/water_use_agr/indexesp.stm, 2016.
- [10] U. de la República Facultad de Agronomía, “Riego en cultivos y pasturas.” Disponible en: <http://www.grupodesarrolloriego.uy/pdf/Libro-2-Seminario-Internacional-de-Riego-en-Cultivos-y-Pasturas.pdf>, 2012.
- [11] J. D. M. Pozo, “Introducción a los dispositivos móviles.” Disponible en: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_2\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_2).pdf), 2012.
- [12] PcMag, “Definición smartphone.” Disponible en: <http://www.pcmag.com/encyclopedia/term/51537/smartphone>, 2016.
- [13] CCM, “Sistema operativo.” Disponible en: <http://es.ccm.net/contents/651-sistema-operativo>, 2014.
- [14] C. R. Sacristán and D. R. Fernández, *Programación en Android*. Ministerio de Educación, 2012.
- [15] L. columna 80 Ángel J. Vico, “Arquitectura de android.” Disponible en: <https://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>, 2011.
- [16] L. columna 80 Ángel J. Vico, “Estructura de un proyecto android.” Disponible en: <https://columna80.wordpress.com/2011/04/13/estructura-de-un-proyecto-android/>, 2011.
- [17] U. C. I. de Madrid, “Software de comunicaciones, programación en dispositivos móviles portables.” <https://sites.google.com/site/swcuc3m/home/android/generalidades/dalvikvm-1>, 2012.
- [18] DefiniciónABC, “Definición de base de datos.” Disponible en: <http://www.definicionabc.com/tecnologia/base-de-datos.php>, 2016.

- [19] M. del Web, “¿qué son las bases de datos?.” Disponible en: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>, 2012.
- [20] M. D. N. N. Tedeschi, “Web services, un ejemplo práctico.” Disponible en: <https://msdn.microsoft.com/es-es/library/bb972248.aspx>, 2012.
- [21] J. G. de proyectos y desarrollo de software, “Desarrollo de software. ciclo de vida rup (rational unified process).” Disponible en: <https://jummp.wordpress.com/2011/04/06/desarrollo-de-software-ciclo-de-vida-rup-rational-unified-process/>, 2011.
- [22] G. B. y. J. R. Ivar Jacobson, *El Proceso Unificado de Desarrollo de Software*. second edition ed., 2000.
- [23] G. B. Ivar Jacobson and J. Rumbaugh, *Unified Software Development Process*. Noviembre 2009.
- [24] P. Kruchten, *Rational Unified Process - An Introduction*,. 3^ª edición ed., Diciembre 2003.
- [25] SOFTENG, “Metodología scrum.” Disponible en: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum.html>, 2012.
- [26] K. Schwaber, “Advanced development methods. scrum development process retrieved,” Abril 2012.
- [27] S. Luis Gómez Sanchez, “Software engineers, metodología scrum para desarrollo de software – aplicaciones complejas,” 2010.
- [28] proyectosagiles.org, “Qué es scrum.” Disponible en: <https://proyectosagiles.org/que-es-scrum/>, 2010.
- [29] extremeprogramming, “Programación extrema.” Disponible en: <http://www.extremeprogramming.org/>, 2013.
- [30] K. B. y Andres Cynthia, “Extreme programming explained,” vol. Segunda, Noviembre 2004.
- [31] wigahluk, “Entre la xp y el rup?.” Disponible en: <https://wigahluk.wordpress.com/2007/06/26/entre-la-xp-y-el-rup/>, 2010.

- [32] R. Invarato, “Asynctask en android.” Disponible en: <https://jarroba.com/async-task-en-android/>, Septiembre 2013.
- [33] SQLMaestro, “Php generador de ayuda en línea de mysql.” Disponible en: <https://www.sqlmaestro.com/products/mysql/phpgenerator/help/>, 2017.
- [34] I. J. Joskowicz, “Reglas y prácticas en extreme programming.” Disponible en: <https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>, 2010.

Anexos y Apéndices

Canal de riego

Anexo A

Origen y destino

Laguna de Pisayambo, toma de captación del agua de riego.



Figura A.1: Laguna de Pisayambo
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Santa Rita, compuerta principal de control de caudal, aquí se distribuye al agua tanto por el ramal norte como por el sur.



Figura A.2: Compuerta principal
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Yatchil Central, canal de riego.



Figura A.3: Canal de riego
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Yatchil Central, tanque de reserva y acequia secundaria..



Figura A.4: Tanque de reserva y acequia secundaria
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Manual de usuario

Anexo B

Manual de aplicación móvil

Para el uso de la aplicación móvil, se ha incluido dentro de la app, dos pantallas de ayuda los cuales se detallará en este manual.

B.1. Instalación de la aplicación

Cabe mencionar de ante mano que la aplicación móvil esta diseñado para ejecutar sobre plataforma Android versión 4.4 en adelante, es justificable ya que los dispositivos móviles actuales incorporan versiones modernas de Android que ofrecen mejores funcionalidades.

Pasos para la instalación:

- Ingresar a la página oficial del Canal de Riego Yatchil en la siguiente dirección <http://www.canalriegoyatchil.poloqc.com/>, en esta ventana debemos presionar sobre el enlace “Descargar aplicación”. Este enlace de forma transparente accederá al repositorio del servidor que aloja la aplicación y empezará la descarga de inmediata.



Figura B.1: Descargar archivo

Elaborado por: Edgar Hipólito Quishpe Collaguazo

- Para poder instalar la aplicación debemos activar la opción fuentes desconocidas en ajustes de seguridad.

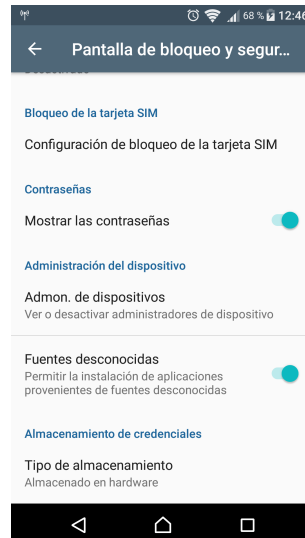


Figura B.2: Activar fuentes desconocidos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

- Buscamos la aplicación descargada en la carpeta download, abrimos el archivo Canal Riego Yatchil.apk para iniciar la instalación y presionamos Instalar.

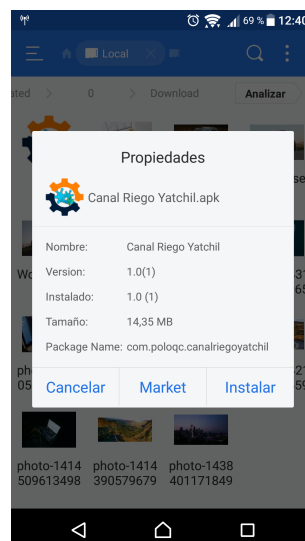


Figura B.3: Instalar
Elaborado por: Edgar Hipólito Quishpe Collaguazo

B.2. Manejo de la aplicación

Una vez instalado nuestra aplicación, nos quedará el siguiente ícono en la pantalla de aplicaciones de nuestro celular.



Figura B.4: Abrir aplicación
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Encontrado nuestra aplicación la abrimos y observaremos la pantalla principal.

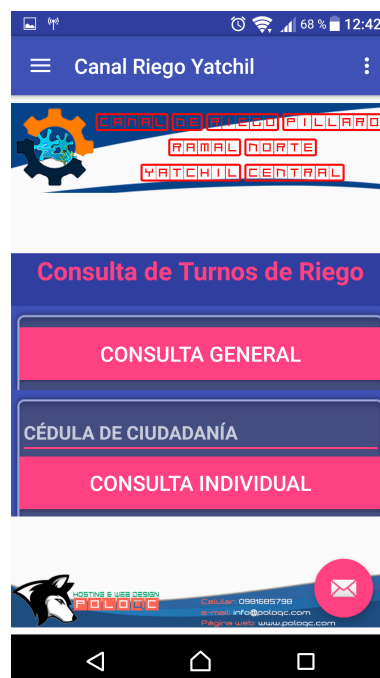


Figura B.5: La aplicación
Elaborado por: Edgar Hipólito Quishpe Collaguazo

El botón de consulta general, realiza el despliegue de turnos de riego de todos los usuarios existentes en la base de datos.

El botón de consulta individual, realiza el despliegue de turnos de riego individual en función del número de cédula ingresado. Si el número de cédula ingresado esta

incorrecto o no existe, la aplicación no permitirá seguir con el proceso de consulta.

Nota: Para hacer consultas en la aplicación es necesario contar con acceso a Internet.

Dependiendo de la opción que se seleccione y una vez validado el dato en el caso de consulta individual, se mostrara la siguiente pantalla que es donde se realizará la consulta.

Para realizar la consulta General o Individual los pasos a seguir son los mismos y se describe a continuación:

- Buscar la toma y el módulo de su interés presionando en las pestañas con los nombres de tomas y módulos.



Figura B.6: Selección de tomas y módulos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

- En las opciones de menú, seleccionamos “Mostrar turnos”.



Figura B.7: Menú
Elaborado por: Edgar Hipólito Quishpe Collaguazo

- Ahora podemos ver la lista de los turnos.

Número de Turno:	1
Nombre:	CHILUIZA NELSON
Total terreno:	6
Total horas:	12
Referencia:	ESCUELA
Primer turno inicial:	2017-01-30 00:00:00

Figura B.8: Lista de turnos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

- Si necesitamos descargar el listado de turnos en un archivo manejable, en las opciones de menú, seleccionamos “Generar PDF”, el archivo .pdf se abre automáticamente.
- Con el archivo .pdf podemos realizar varias operaciones como copiar a un dispositivo externo, compartir, imprimir, entre otras opciones.

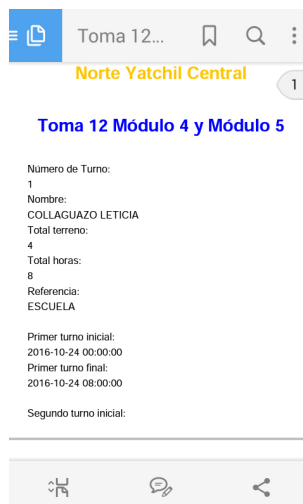


Figura B.9: Archivo PDF
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Nota: Para “Generar PDF” se necesita contar con una memoria microSD en nuestro dispositivo móvil.

Anexo C

Manual de aplicación web

Para acceder al sitio web debemos digitar la siguiente dirección <http://www.canalriegoyatchil.poloqc.com/>, en esta ventana debemos presionar sobre el enlace “Iniciar sesión”

En la página de inicio de sesión encontraremos la información necesaria para el ingreso como usuario invitado y por políticas de privacidad los datos de ingreso del usuario administrador no se proporciona en este documento.

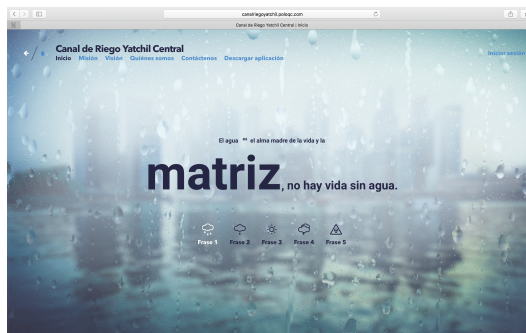


Figura C.1: Página principal

Elaborado por: Edgar Hipólito Quishpe Collaguazo

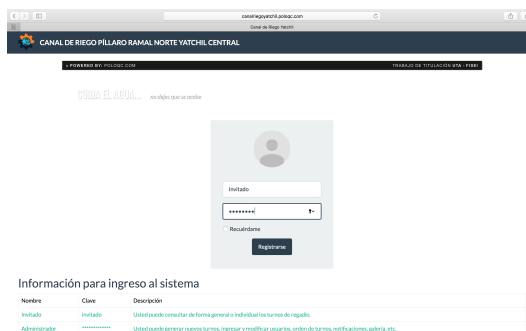


Figura C.2: Página de inicio de sesión

Elaborado por: Edgar Hipólito Quishpe Collaguazo

C.1. Manejo de la aplicación (usuario invitado)

Una vez dentro de la aplicación móvil, el usuario podrá seleccionar cualquiera de las opciones y visualizar el contenido de los turnos de riego.

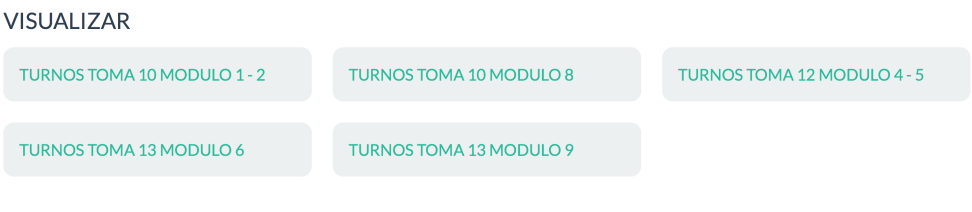


Figura C.3: Turnos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Dentro de la pantalla que muestra los turnos, se podrá visualizar todos los turnos que corresponde a la toma y módulos actuales, se podrá realizar un turno específico ingresando los nombres y/o apellidos. Para poder trasladar la información en un archivo manejable, se tiene la opción de exportar o se puede imprimir directamente.

Acciones	NUMERO DE TURNO	USUARIO	TOTAL TERRENO	TOTAL HORAS	REFERENCIA	H1 HI	H1 HF	H2 HI	H2 HF	H3 HI	H3 HF	H4 HI	H4 HF	H5 HI	H5 HF
[Icon]	4	CHICAIZA RAMIREZ SEGUNDO CARLOS	2	4	A LADO DE JUAN TITUANA Y ANGEL ORTEGA	10 Apr 2017 20:00:00	11 Apr 2017 00:00:00	24 Apr 2017 10:00:00	24 Apr 2017 14:00:00	08 May 2017 00:00:00	08 May 2017 04:00:00	21 May 2017 14:00:00	21 May 2017 18:00:00	04 Jun 2017 04:00:00	04 Jun 2017 08:00:00
[Icon]	35	SIZA RAMIREZ SEGUNDO JUAN SILVERIO	1	2	A LADO DE AGUSTINA GRANJA Y JORGE TITUANA	17 Apr 2017 18:00:00	17 Apr 2017 20:00:00	01 May 2017 08:00:00	01 May 2017 10:00:00	14 May 2017 22:00:00	15 May 2017 00:00:00	28 May 2017 12:00:00	28 May 2017 14:00:00	11 Jun 2017 02:00:00	11 Jun 2017 04:00:00
[Icon]	49	CHICAIZA RAMIREZ SEGUNDO ASENCIO	1	2	A LADO DE JORGE GUAITA Y JORGE RAMIREZ	21 Apr 2017 08:00:00	21 Apr 2017 10:00:00	04 May 2017 22:00:00	05 May 2017 00:00:00	18 May 2017 12:00:00	18 May 2017 14:00:00	01 Jun 2017 02:00:00	01 Jun 2017 04:00:00	14 Jun 2017 16:00:00	14 Jun 2017 18:00:00
[Icon]	50	RAMIREZ ALTASHI JORGE ERNESTO	1	2	A LADO DE ASENCIO CHICAIZA Y ROSALINO CHICAIZA	21 Apr 2017 10:00:00	21 Apr 2017 12:00:00	05 May 2017 00:00:00	05 May 2017 02:00:00	18 May 2017 14:00:00	18 May 2017 16:00:00	01 Jun 2017 04:00:00	01 Jun 2017 06:00:00	14 Jun 2017 18:00:00	14 Jun 2017 20:00:00
[Icon]	51	CHICAIZA CH ROSALINO	1	2	A LADO DE JORGE RAMIREZ Y JOSE CHICAIZA	21 Apr 2017 12:00:00	21 Apr 2017 14:00:00	05 May 2017 02:00:00	05 May 2017 04:00:00	18 May 2017 16:00:00	18 May 2017 18:00:00	01 Jun 2017 06:00:00	01 Jun 2017 08:00:00	14 Jun 2017 20:00:00	14 Jun 2017 22:00:00

Figura C.4: Visualizar turnos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

C.2. Manejo de la aplicación (usuario administrador)

Para el manejo general de la base de datos, el administrador contará con la siguiente pantalla.

Acciones	CEDULA	NOMBRES	APELLIDOS	DIRECCION	TELEFONO	EMAIL	ACTIVO
	0000000011	EDELINA	TOAPANTA	YATCHIL	Nulo	Nulo	✓
	0000000012	CELIA	TOAPANTA	YATCHIL	Nulo	Nulo	✓
	0000000013	JORGE AMABLE	TITUANA TOAPANTA	YATCHIL	Nulo	Nulo	✓
	0500854096	SEGUNDO	TITUANA MOPOSITA	YATCHIL	Nulo	Nulo	✓
	0501865299	GLORIA DOLORES	ACOSTA HIDALGO	YATCHIL	Nulo	Nulo	✓
	0501899579	MARIA EULALIA	SHINGON PUMASHUNTA	YATCHIL	Nulo	Nulo	✓
	0502682982	LUIS ALFONSO	TIPANTUNA TIGSE	YATCHIL	Nulo	Nulo	✓
	0502754823	MARIA LOURDES	TOAPANTA CAÑAR	YATCHIL	Nulo	Nulo	✓

Figura C.5: Visualizar turnos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

En esta pantalla se puede observar los botones comunes de agregar, actualizar y eliminar, dependiendo de la política establecida en cada formulario.

En cuanto a lo mas importante que es la generación de calendarios, el administrador debe establecer con anterioridad el orden de turnos que corresponde a cada usuario en cada uno de las cinco tomas y módulos dejando los campos de horarios en nulo como se ve en el siguiente gráfico.

Acciones	NUMERO DE TURNO	USUARIO	TOTAL TERRENO	REFERENCIA	H1	H1	H2	H2	H3	H3	H4	H4	H5	H5
	1	CALAPIÑA ORTEGA YOLANDA	3	A LADO DE EDISON CALAPIÑA	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo
	2	CALAPIÑA ORTEGA EDISON RAUL	3	A LADO DE YOLANDA CALAPIÑA Y MYRIAN CHILQUIQUINGA	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo
	3	CHILQUIQUINGA ORTEGA LUZ MARIA	2	A LADO DE EDISON CALAPIÑA Y ALICIA CHILQUIQUINGA	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo
	4	CHILQUIQUINGA ORTEGA BLANCA ALICIA	2	A LADO DE MYRIAN CHILQUIQUINGA Y ROSA CHILQUIQUINGA	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo	Nulo

Figura C.6: Visualizar turnos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Luego de esto se deberá ir al formulario de calendario. En este formulario debemos presionar en agregar e ingresar los parámetros correspondientes que son: Fecha de inicio , Toma y módulo y fecha de creación que se ingresa automáticamente.

Fecha de inicio: Fecha a partir del cual se inicia a distribuir las fechas y horas de los turnos de riego.

Toma y módulo: Nombre de una de las cinco tablas que representa a cada toma de riego al que vamos a generar el calendario.

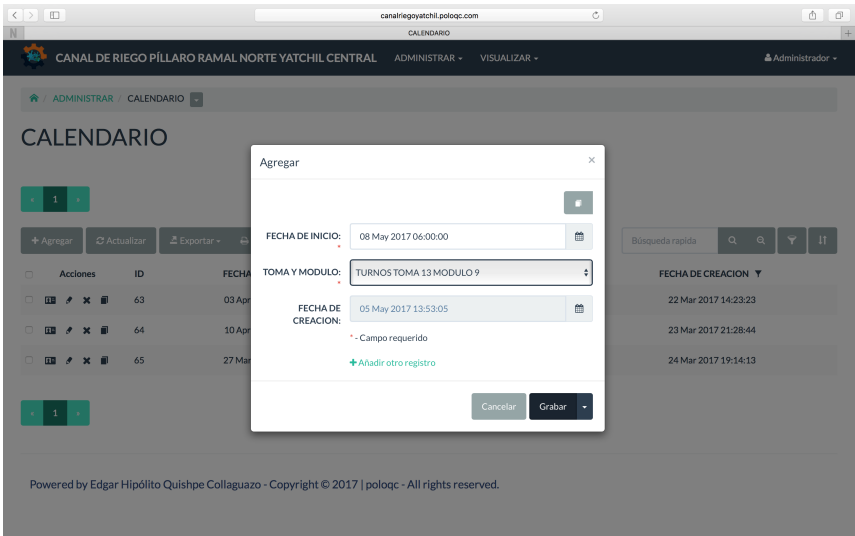


Figura C.7: Visualizar turnos
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Ya agregado la calendarización, regresamos al formulario que especificamos en el paso anterior (TURNOS TOMA 13 MODULO 9) y comprobamos los resultados.

	NUMERO DE TURNO	USUARIO	TOTAL TERRENO	REFERENCIA	H1 HI	H1 HF	H2 HI	H2 HF	H3 HI	H3 HF	H4 HI	H4 HF	H5 HI
	1	CALAPIÑA ORTEGA YOLANDA	3	A LADO DE EDISON CALAPIÑA	08 May 2017 06:00:00	08 May 2017 12:00:00	25 May 2017 20:00:00	26 May 2017 02:00:00	12 Jun 2017 10:00:00	12 Jun 2017 16:00:00	30 Jun 2017 00:00:00	30 Jun 2017 06:00:00	17 Jul 2017 14:00:00
	2	CALAPIÑA ORTEGA EDISON RAUL	3	A LADO DE YOLANDA CALAPIÑA Y MYRIAN CHILQUINGA	08 May 2017 12:00:00	08 May 2017 18:00:00	26 May 2017 02:00:00	26 May 2017 08:00:00	12 Jun 2017 16:00:00	12 Jun 2017 22:00:00	30 Jun 2017 06:00:00	30 Jun 2017 12:00:00	17 Jul 2017 20:00:00
	3	CHILQUINGA ORTEGA LUZ MARIA	2	A LADO DE EDISON CALAPIÑA Y ALICIA CHILQUINGA	08 May 2017 18:00:00	08 May 2017 22:00:00	26 May 2017 08:00:00	26 May 2017 12:00:00	12 Jun 2017 22:00:00	13 Jun 2017 02:00:00	30 Jun 2017 12:00:00	30 Jun 2017 16:00:00	18 Jul 2017 02:00:00
	4	CHILQUINGA ORTEGA BLANCA ALICIA	2	A LADO DE MYRIAN CHILQUINGA Y ROSA CHILQUINGA	08 May 2017 22:00:00	09 May 2017 02:00:00	26 May 2017 12:00:00	26 May 2017 16:00:00	13 Jun 2017 02:00:00	13 Jun 2017 06:00:00	30 Jun 2017 16:00:00	30 Jun 2017 20:00:00	18 Jul 2017 06:00:00

Figura C.8: Resultado
Elaborado por: Edgar Hipólito Quishpe Collaguazo

Como se observa en la imagen, se ha generado las fechas y horas que corresponde hacer uso del agua a cada usuario.