

Universidad Autónoma de Baja California
Facultad de Contaduría y Administración
Facultad de Ciencias Químicas e Ingeniería
Maestría en Tecnologías de la Información y la Comunicación



Diseño Y Desarrollo De Un Sistema De Información Web Utilizando
Una Estación Meteorológica Weatherlink En El Centro De Ingeniería Y
Tecnología De UABC Valle Las Palmas

TESIS

Para Obtener el Grado de
Maestro en Tecnologías de la Información y la
Comunicación

Presenta:

Oscar Yunue Quiroz Gaxiola

Bajo la dirección de:

M.C. José M. Villegas Izaguirre

Co dirección:

M.C. Marco A. Pinto Ramos

Tijuana, Baja California, México

Mayo 2014

Dedicatoria

A mi familia.

Aprobación de la tesis por el comité de la MTIC

Agradecimientos

A la Universidad Autónoma de Baja California, CONACYT, maestros y amigos que estuvieron siempre mostrando su apoyo.

Resumen

El objetivo del presente trabajo de tesis, es el diseño y desarrollo de un sistema de información para gestionar los datos capturados por una estación meteorológica en el Centro De Ingeniería y Tecnología Valle Las Palmas, enfocado al registro de las variables en un sistema de base de datos. Además, considerar como parte de los requerimientos necesidades del área de investigación meteorológica y energías renovables, donde es común el manejo de altos volúmenes de datos, producto de los diferentes tipos de variables y periodos de registros cortos, justificando la importancia de utilizar un sistema de base de datos y de consultas eficiente, que permita un acceso preciso a las actividades meteorológicas registradas y proveer la información solicitada por los usuarios con mayor fluidez y practicidad.

El desarrollo de consultas sin un sistema de información adecuado, es una restricción que complica el desarrollo de investigaciones, debido al alto volumen de información que se manejan en este tipo de estudios, este trabajo presenta una solución mediante el diseño y la implementación de un sistema para gestión de información y el manejo flexible de registros de datos y consultas históricas.

Abstract

The objective of this thesis is the design and development of an information system to manage data captured by a weather station in the Engineering and Technology Center Valle Las Palmas, focused record variables in a database system. Also considered part of the requirements needs of the area meteorological research and renewable energy, which is common handling high volumes of data, due to the different types of variables and short periods of records, justifying the importance of using a system database queries and efficiently, enabling access to accurate weather activities recorded and provide the information requested by users with practicality in the consultation process.

The development of queries without an information system is a constraint that complicates the development of research, due to the high volume of information to be handled in such studies, this paper presents a solution by designing and implementing a system for information management and flexible handling of data logs and historical queries.

INDICE

INTRODUCCIÓN Y ANTECEDENTES	13
1.1 Introducción	14
1.2 Antecedentes	16
PLANTEAMIENTO DEL PROBLEMA	20
2.1 Problema de investigación	21
2.2 Justificación	22
2.3 Objetivos	23
2.3.1 General	23
2.3.2 Específicos	23
1.4 Alcance	24
MARCO TEÓRICO	25
3.1 Desarrollo de software	26
3.1.1 Definición de desarrollo de software.	26
3.1.2 Ingeniería de Software	26
3.1.3 Calidad de software	27
3.1.4 Programación Orientada a objetos	28
3.2 Meteorología	30
3.2.1 Conceptos Meteorológicos	30
3.2.2 Estaciones Meteorológicas :	30
3.2.3 Estaciones WeatherLink	31
3.2.4 Parámetros meteorológicos :	34
2.4 Observaciones Meteorológicas:	38
3.3.3 Herramientas de desarrollo de software	40
3.3.3.1 lenguaje SQL	40
3.3.3.2 Sistema gestor de bases de datos	40
3.3.3 Apache 2	41
3.3.3.4 Base de Datos	42

3.3.5 Ruby	43
3.3.6 Ruby on Rails 3	46
3.3.7 Modelo MVC	47
3.3.8 ORM	48
METODOLOGÍA	50
4.0 Introducción	51
4.1 - Investigación preliminar	51
4.2 - Metodología de desarrollo	52
4.3 Elección de tecnologías para el proyecto	54
DESARROLLO	56
5.1 Desarrollo	57
6 Pruebas	66
6.1 Caja blanca	67
6.3 Caja Negra:	72
6.3 Resultado:	73
7.1 Ajustes	74
RESULTADOS Y CONCLUSIONES	76
7.1 RESULTADOS	77
7.2 Conclusión	80
7.3 Trabajo Futuro	81
BIBLIOGRAFIA	82
7.1. Bibliografía	83
ANEXOS	85
Entrevista semi estructurado para el análisis de requerimientos.	86
Entrevista semi estructurada para validación.	89
Apéndice	91

Índice de imágenes

<i>Imagen 1 Centro de Ingeniería y Tecnología (CITEC) en UABC Valle las Palmas</i>	<i>14</i>
<i>Imagen 2 Interfaz de la aplicación de monitoreo WeatherLink</i>	<i>32</i>
<i>Imagen 3 Estación meteorológica WeatherLink en funcionamiento</i>	<i>33</i>
<i>Imagen 4 Flujo del método ORM (Mapeo Objeto Relacional)</i>	<i>48</i>
<i>Imagen 5 Re presentación grafica de las etapas del modelo de desarrollo de software en cascada</i>	<i>54</i>
<i>Imagen 6 Sitio oficial con las ultimas versiones del lenguaje de programación Ruby.....</i>	<i>59</i>
<i>Imagen 7 Asistente de instalación del lenguaje de programación Ruby</i>	<i>60</i>
<i>Imagen 8 Creación del proyecto en Rails</i>	<i>61</i>
<i>Imagen 9 Asistente de instalación para el gestor de versiones GIT.</i>	<i>62</i>
<i>Imagen 10 Instalación del componente passenger para la utilización de la aplicación en el servidor apache.....</i>	<i>62</i>
<i>Imagen 11 Repositorio remoto utilizado para el desarrollo del sistema web.</i>	<i>63</i>
<i>Imagen 12 La terminal muestra los comandos para la creación de los modelos y esquema de la base de datos.....</i>	<i>64</i>
<i>Imagen 13 Cambio de entorno de "development" a "production", para iniciar el servidor en modo producción</i>	<i>64</i>
<i>Imagen 14 Portal rubygems.org con repositorio de componentes utilizados en el presente proyecto.....</i>	<i>68</i>
<i>Imagen 15 Vista para despliegue en dispositivos móviles.</i>	<i>69</i>
<i>Imagen 16 Código fuente de la aplicación backend utilizada para pruebas.</i>	<i>70</i>
<i>Imagen 17 La aplicación muestra una lista ordenada de los datos extraídos del archivo de registro de la estación meteorológica.</i>	<i>70</i>
<i>Imagen 18 La aplicación muestra la opción "add" encargada de guardar los registros extraídos del archivo en la base de datos.</i>	<i>71</i>
<i>Imagen 19 Tabla de parámetros después de ajuste, mostrando por default solo los parámetros primarios.....</i>	<i>74</i>
<i>Imagen 20 Interfaz para agregar una nueva estación meteorológica.</i>	<i>75</i>
<i>Imagen 21 Herramienta web de reporte de datos meteorológicos con componente de reporte grafico.</i>	<i>78</i>
<i>Imagen 22 Lista de parámetros detallada aplicando herramientas de usabilidad.</i>	<i>79</i>
<i>Imagen 23 Sitio oficial con las ultimas versiones del gestor de versiones GIT.</i>	<i>93</i>

<i>Imagen 24 Asistente de instalación para el gestor de versiones GIT.....</i>	<i>93</i>
<i>Imagen 25 Asistente de instalación de las dependencias de Microsoft Visual C++ 2008.....</i>	<i>94</i>
<i>Imagen 26 Asistente de instalación para el sistema gestor de base de datos en Windows.</i>	<i>95</i>
<i>Imagen 27 Asistente de instalación para el servidor apache2.</i>	<i>96</i>
<i>Imagen 28 Agregado del modulo "passenger" u "virtual host" al archivo de configuración del servidor apache.....</i>	<i>97</i>

Índice de Tablas

<i>Tabla 1</i>	<i>Parámetros capturados por la estación WeatherLink.....</i>	<i>34</i>
<i>Tabla 2</i>	<i>Descripción de pasos en la investigación preliminar.....</i>	<i>51</i>
<i>Tabla 3</i>	<i>Lista de herramientas involucradas en el desarrollo del proyecto</i>	<i>55</i>
<i>Tabla 4</i>	<i>Especificaciones del servidor utilizado para la instalación del sistema web.....</i>	<i>57</i>
<i>Tabla 5</i>	<i>Componentes utilizados para mejoras de interfaz, importación de datos y registros ala base de datos.....</i>	<i>67</i>

Capítulo I

INTRODUCCIÓN Y ANTECEDENTES

1.1 Introducción

El presente trabajo de tesis se desarrollara en el Centro de Ingeniería y Tecnología (CITEC) en UABC Valle Las Palmas y esta enfocado en el área de energías renovables.



Imagen 1 Centro de Ingeniería y Tecnología (CITEC) en UABC Valle las Palmas

Los investigadores desean transmitir los datos capturados por la estación meteorológica a través de un sistema web que utiliza como mecanismo de captura una estación del clima *WeatherLink*, la cual esta en operación, y por medio de la cual se capturan actualmente los parámetros climatológicos.

Las estaciones meteorológicas son una herramienta fundamental en el estudio de los parámetros climatológicos, para que los investigadores generen un diagnóstico y vigilancia meteorológica, así como para elaborar los pronósticos, por otra parte se planea obtener mas terminales en el futuro y desarrollar estudios mas precisos, y publicar estos resultados en un sistema de informacion web el cual podra ser acesado por los involucrados del area de energia renovables.

Al considerar los escenarios del desarrollo de investigaciones, surgen restricciones que se deben considerar durante la etapa del análisis de requerimientos e incorporar terminologías y significados intrínsecos de cada uno de los parámetros para poder definir una presentación y clasificación eficiente de los datos de los datos meteorológicos por medio de un sistema de información para la utilización mas eficiente. El estudio meteorológico reúne una serie de necesidades en aspectos de accesibilidad y usabilidad que presentes permiten formar y aportar una panorama completo para las lecturas y diagnósticos que por su naturaleza son complicados de predecir, referente a esta complejidad es que los medios necesarios para aproximarse lo mas posible a la realidad de acuerdo a los datos que utilizan los investigadores deben ser claros y funcionales.

1.2 Antecedentes

Desde la antigüedad el hombre ha intentado comprender elementos meteorológicos para poder utilizar el conocimiento a su favor, la observación de los cambios del clima y fenómenos atmosféricos es una actividad importante y compleja si se toma en cuenta la cantidad de variables que se manejan, su estudio ha evolucionado con el tiempo al igual que las herramientas de medición, lo que dio origen a la meteorología como ciencia encargada del estudio de la atmósfera, de sus propiedades y de los fenómenos que en ella tienen lugar. El estudio de la atmósfera se basa en el conocimiento de una serie de magnitudes, o variables meteorológicas, como la temperatura, presión atmosférica, humedad entre otras.

La meteorología es una disciplina amplia observación en campo y se apoya en equipamiento, técnicas y herramientas para mejorar la calidad de los estudios, por tanto la observación es una técnica que tiene sus limitaciones y sin las herramientas para comprobar las inferencias, puede hacer que simples observaciones se desvíen hacia una afirmación errónea.

Los sistemas de información, por otra parte brindan una solución para gestionar la información, de las herramientas meteorológicas, porque a través de su uso se logran importantes mejoras, debido a que pueden almacenar y registrar una gran cantidad de datos para posteriormente procesar y generar información.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información (O'Brien & Marakas, 2006). En un sistema de información web meteorológica se utiliza como mecanismo de entrada, el registro de parámetros a través de una estación meteorológica, la cual se almacena en una base de datos, para posteriormente ser procesada y consultada por el usuario final.

Los datos capturados por la estación se utilizan tanto para la elaboración de predicciones meteorológicas a partir de modelos numéricos como para estudios climáticos. Los estudios e investigaciones meteorológicos son complejos debido a la diversidad de variables que interactúan, es importante entonces la incorporación de herramientas que faciliten el procedimiento de acceso a los datos con el propósito de generar información relevante que se aproxime cuanto mas posible a la realidad.

Las herramientas de medición en el área de la meteorología juegan un papel importante en los estudios e investigaciones y pueden ser un factor determinante en el éxito de un proyecto, como en estudios anteriores en los cuales se proponen encontrar soluciones en base a herramientas de software, en el caso de la el Instituto Politécnico Superior de la Habana Cuba, tenían el inconveniente de no contar con el equipo para la captura de variables secundarias, “el problema se soluciono, por medio de herramientas de minería de datos en el cual podían obtenerlas en base a variable primarias” (Gil, 2006) , en otro caso de “*Monitoreo de una Estación Meteorológica vía internet.*” se implemento una herramienta de monitoreo de parámetros a través del internet por medio de un sistema web, proyecto en el cual se pudo completar el objetivo para la “creación de mapas solares y eólicos para su posterior aprovechamiento de las fuentes renovables de energía en el estado de Zacatecas” (Ismael Guardado Dávila, 2007).

En ambos casos se pudo alcanzar el objetivo haciendo uso de estrategias tecnológicas, el primer caso mediante la utilización de técnicas de base de datos y en el segundo con la utilización de el internet como canal de comunicación. Con la investigación de los parámetros secundarios en el instituto superior politécnico de la Habana Cuba, la estimación de parámetros meteorológicos con la correcta utilización de consultas de datos, se aplicaron técnicas de minerías de datos para procesar los datos en información relevante se concluyo lo siguiente “se pueden obtener datos gracias a los modelos normalizados que presentan un coeficiente de correlación alto, un error cuadrático pequeño y un bajo porcentaje

de instancias fuera de rango” (Gil, 2006).

El presente proyecto se desarrolla en el Centro de Ingeniería y Tecnología (CITEC) unidad Valle Las Palmas, “inaugurado en el 2009 en respuesta al compromiso establecido con la política institucional de apelación, el cual cuenta con la convicción de ser una entidad académica en el área de profesionales en el área de ingeniería y tecnología, como lo marca el plan de desarrollo”. (CITEC, 2010)

Como se planteo anteriormente el énfasis en la investigación y el desarrollo de nuevos conocimientos, se propone en el 2011 implementar un sistema de monitoreo meteorológico como complemento al plan de estudio de ingeniería en energías renovables, la cual será utilizada como herramienta primaria para estudios e investigaciones meteorológicas, el proyecto comenzó con la implementación de una estación *WeatherLink* la cual se instalo en CITEC y fue puesta en funcionamiento.

El Centro de Ingeniería y Desarrollo Tecnológico (CITEC), utiliza la estación *WeatherLink* para capturar los parámetros meteorológicos sin embargo las investigaciones por parte del área de ciencias renovables se han visto afectadas por la falta de una herramienta de consulta y gestión de información que permita traducir los registros de la estación meteorológica en información relevante, ya que las investigaciones meteorológicas requieren de una gran cantidad de capturas y consultas a datos históricos, además de aplicar comparaciones a los mismos para identificar tendencias y patrones, por tal motivo es complicado desarrollar estudios sin las herramientas adecuadas.

La ausencia de una herramienta adecuada para el manejo y las consultas a las bases de datos e información histórica puede obstruir el desarrollo de una investigación, tal es el caso de del grupo de medio ambiente del Instituto Superior Politécnico José Antonio Echeverría, Que cuenta con una estación

meteorológica automática en la cual no se dio ningún tratamiento o explotación a la estación antes de implementarse el sistema de variables meteorológicas (Gil, 2006).

Las estaciones meteorológicas *WeatherLink* del proveedor Davis, se utilizan como instrumentos de medición y suelen ser eficientes en la captura de variables meteorológicas, su procedimiento de registro de parámetros comienza cuando la estación captura los valores correspondientes a temperatura, humedad, dirección del viento, entre otras. Posteriormente los envía vía *wireless* a una terminal que despliega continuamente las mediciones. Por último la terminal envía las mediciones al servidor donde se guardan en un archivo donde automáticamente se registran los valores en un archivo periódicamente, el tiempo de registro con el que se trabajara en el proyectos es de diez minutos.

La automatización como en todos los ámbitos es una característica muy útil para mantener actualizada la información de un sistema, en este caso los datos meteorológicos tales como: temperatura ambiental, humedad, velocidad y dirección del viento, presión atmosférica, precipitación de lluvia, radiación solar, entre otras. Por tanto un sistema con información actualizada permite incrementar la precisión en la medición y determinación de todos los elementos que en su conjunto representan las condiciones del estado atmosférico. (Carlos Gómez, 2007).

Capítulo II

PLANTEAMIENTO DEL PROBLEMA

2.1 Problema de investigación

A raíz de la apertura de proyectos de investigación meteorológicos en el área de energías renovables en CITEC ¹Valle Las Palmas, surge la necesidad de buscar una solución a la siguientes problemáticas: no existe una base de datos, ni una herramienta de gestión de información para una estación meteorológica.

Actualmente se cuenta con una estación meteorológica ubicada en las instalaciones de CITEC, se planea incorporar nuevas estaciones en el futuro y es importante que se busque una solución que tenga la capacidad de implementación de mas estaciones meteorológicas.

¹ Centro De Ingeniería y Tecnología, en la Universidad Autónoma de Baja California ubicado en Valle las Palmas.

2.2 Justificación

Por medio del sistema se obtendrán resultados que permitan consultar los parámetros meteorológicos en el Centro de Ingeniería y Tecnología de UABC Valle las Palmas. El sistema será utilizado como herramienta en el desarrollo de proyectos universitarios por alumnos e investigadores de energías renovables y ciencias afines, además contempla la integración de mas estaciones meteorológicas, para obtener información con mayor precisión.

2.3 Objetivos

2.3.1 General

Diseñar e implementar un sistema de información web en el Centro de Ingeniería y Tecnología (CITEC) UABC Valle Las Palmas, que permita almacenar los parámetros que registra una estación meteorológica *WeatherLink* en una base de datos, así como una herramienta de consulta de los mismos, con el propósito de facilitar a los maestros y estudiantes del área de energías renovables en las investigaciones meteorológicas.

2.3.2 Específicos

- Identificar los criterios y características que se toman en cuenta durante una investigación meteorológica.
- Aplicar técnicas que optimicen el rendimiento de la base de datos.
- Implementar un modelo flexible que facilite el proceso de consultas complejas.
- Aplicar medidas de usabilidad y eficiencia en los procedimientos de acceso a la información.
- Implementar un modelo que contemple la incorporación de mas estaciones meteorológicas en el futuro.
- Diseño y desarrollo de sitio web para consulta y publicación de datos.

1.4 Alcance

El proyecto consiste en el diseño e implementación de una base de datos para capturar los registros de una estación meteorológica *WeatheLink* y la construcción de un sistema de información web mediante el cual se desarrollaran consultas de los parámetros capturados, con opciones de visualización tanto de los registros actuales como de una interfaz para consultas históricas a la base de datos.

El sistema web contemplará la integración de mas estaciones meteorologías en el futuro, pero en el presente proyecto se trabajara solamente con una estación.

Capítulo III

MARCO TEÓRICO

3.1 Desarrollo de software

3.1.1 Definición de desarrollo de software.

La asociación IEEE [5] define ingeniería de software como :

“La aplicación de un enfoque sistemático, disciplinado, capaz de cuantificación para el desarrollo, operación y mantenimiento de software, es decir, la aplicación de la ingeniería de software” (Alain Abran, 2004).

Desde la descripción que brinda IEEE definición de ingeniería de software es evidente que la medición es fundamental para el proceso de desarrollo de software como una disciplina de ingeniería. En las últimas décadas, cientos de modo - métricas de software llamados han sido propuestos por investigadores y profesionales por igual, tanto en estudios teóricos y empíricos, para los productos de software de medición y los procesos de software (Abran, 2008).

Como la IEEE menciona en su definición, es evidente que la medición es importante para la ingeniería de software, por que expone como incógnita la madurez del proceso de medición o calificación del software, en las ultimas décadas como respuesta a una necesidad por parte de los desarrolladores de software (Abran, 2008).

3.1.2 Ingeniería de Software

La ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza. En esta definición, existen dos frases clave:

1. Disciplina de la ingeniería. Aplican teorías, métodos y herramientas

donde sean convenientes, pero las utilizan de forma selectiva y siempre en busca de soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales (Sommerville, 2005).

2. Todos los aspectos de producción de software. La ingeniería del software no solo comprende los procesos técnicos del desarrollo de software, sino también con actividades tales como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software (Pressman, 1997).

3.1.3 Calidad de software

Para las empresas de desarrollo de software lograr un alto nivel de calidad de un producto o servicio es un objetivo importante, tienen exigencias particulares que son resultado de altas expectativas del cliente , un mercado globalizado y competitivo, para satisfacer estas demandas, las organizaciones de software debe tener la capacidad de desarrollar y mantener software para satisfacer las necesidades del cliente. El concepto de “calidad de software” es claro, importante y popular, pero poco extendido en el área de desarrollo de software. A continuación algunas definiciones de calidad extraídas de estándares internacionales:

“La calidad es la suma de todos aquellos aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades, expresadas o implícitas” (ISO 9000 , 2005)

“La calidad de Software se define como la conformidad del software producido a las necesidades implícitas y explícitas“ (ISO 25010, 2011).

Según Distal Rosique, (Distal Rosique, 1999): “Calidad del Software consiste en desarrollar productos lógicos que satisfagan las necesidades del usuario, los requisitos implícitos (a menudo, no mencionados) y que tiendan a cero defectos.” Para conseguir un software de calidad es necesario realizar una serie de tareas a lo largo de todo el proceso de desarrollo de la aplicación. Es lo que se conoce como la garantía de calidad del software (SQA: “*Software Quality Assurance*”).

La calidad del software es un concepto complejo que no se puede definir de una manera sencilla ya que intervienen diversos elementos. En principio, la administración de la calidad comprende el diseñar procedimientos y estándares a utilizar durante el desarrollo de software y comprobar que todo el personal los siga.

3.1.4 Programación Orientada a objetos

La POO es una técnica para desarrollar soluciones computacionales mediante la implementación de componentes de software (objetos de software).

- **Objeto:** Componente o código de software que contiene en sí mismo tanto sus características como sus comportamientos; se accede a través de su interfaz o signatura.
- **Campo:** Es una característica de un objeto, que ayuda a definir su estructura y permite diferenciarlo de otros objetos. Se define con un identificador y un tipo, el cual indica los valores que puede almacenar. El conjunto de valores de los campos definen el estado del objeto.
- **Método:** Es la implementación de un algoritmo que representa una operación o función que un objeto realiza. El conjunto de los métodos de

un objeto determinan el comportamiento del objeto.

La POO es un paradigma de la programación de computadores; esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, para proporcionar un medio bien definido, con el objetivo de visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.

La POO se basa en el modelo objeto donde el elemento principal es el objeto, el cual es una unidad que contiene todas sus características y comportamientos en sí misma, lo cual lo hace como un todo independiente pero que se interrelaciona con objetos de su misma clase o de otras clase, como sucede en el mundo real (Fernández L. A., 2013).

Anterior al paradigma de objetos, está el paradigma algorítmico o de procesos, el cual se fundamenta en los procesos o funciones que se llevan a cabo en el mundo real dentro del dominio del problema analizado. Se refiere a lo que entra, como lo maneja el proceso, y lo que sale del proceso. La programación tradicional la sustentan los procesos, algoritmos, bloques de construcción modulares cuya abstracción va de lo general a lo particular, mientras que en la POO tiene como marco de referencia conceptual el objeto, el cual pertenece a una clase que agrupa a todos compañeros con las mismas características y un comportamiento similar.

Una ventaja de la POO frente al paradigma algorítmico es la facilidad que brinda a través de sus herramientas, de concebir, analizar, modelar, diseñar e implementar el mundo real de manera fiel a como se presenta en la realidad; el paso que hay desde la concepción y asimilación del problema hasta la implementación del mismo es un proceso que se hace de manera casi natural. Esto porque el mundo está lleno de objetos reales, los cuales se pueden representar como tales en una solución computarizada (Fernández, 2013).

3.2 Meteorología

3.2.1 Conceptos Meteorológicos

Es el estudio de los fenómenos atmosféricos y de los mecanismos que producen el tiempo, orientado a su predicción. Del griego, meteoros (alto), logos (tratado).

Los fenómenos atmosféricos o meteorológicos pueden ser:

Aéreos, como el viento, acuosos, como la lluvia, la nieve y el granizo, luminosos, como la aurora polar, presión, temperatura y humedad son los factores climáticos fundamentales en el estudio y predicción del tiempo. La temperatura, sometida a numerosas oscilaciones, se halla condicionada por la latitud y por la altura sobre el nivel del mar.

La presión atmosférica, variable también en el transcurso del día, es registrada en los mapas meteorológicos mediante el trazado de las isobaras o puntos de igual presión, que permiten identificar los centros de baja presión o borrascas, cuya evolución determina en gran parte el tiempo reinante.

La meteorología utiliza instrumentos esenciales, como el barómetro, el termómetro y el higrómetro, para determinar los valores absolutos, medios y extremos de los factores climáticos. Para el trazado de mapas y la elaboración de predicciones es fundamental la recolección coordinada de datos en amplias zonas, lo que se realiza con la ayuda de los satélites meteorológicos.

3.2.2 Estaciones Meteorológicas :

Las observaciones se realizan en lugares establecidos, donde es necesario contar con datos meteorológicos para una o varias finalidades, ya sea en tiempo real, en tiempo diferidos o ambos. Estos lugares deben reunir determinadas

condiciones técnicas normalizadas y se los denomina "estaciones meteorológicas".

3.2.3 Estaciones WeatherLink

Las estaciones meteorológicas *WeatherLink Imagen 3*, esta pensada para medición integrando herramientas de medición meteorológica integral. Viene programada para entregar, medir y registrar datos meteorológicos, incluyendo las variables de velocidad y dirección de viento, temperatura del aire, humedad relativa, presión barométrica, radiación solar y lluvia, evapotranspiración que se asocia con el manejo de riego de áreas verdes (Dunkel, 2006).

El proceso de medición consiste en la captura y registro de cuarenta y cuatro parámetros tabla 2.1, y empieza con la captura de estos por parte de la estación vía inalámbrica a una terminal receptora que recibe los datos y los registra en el servidor o PC al que esta conectado, por ultimo, utilizando como medios de salida los registros del día se almacenan en un archivo de texto y pueden ser visualizados e la terminal local del sistema *WeatherLink Imagen 2*.

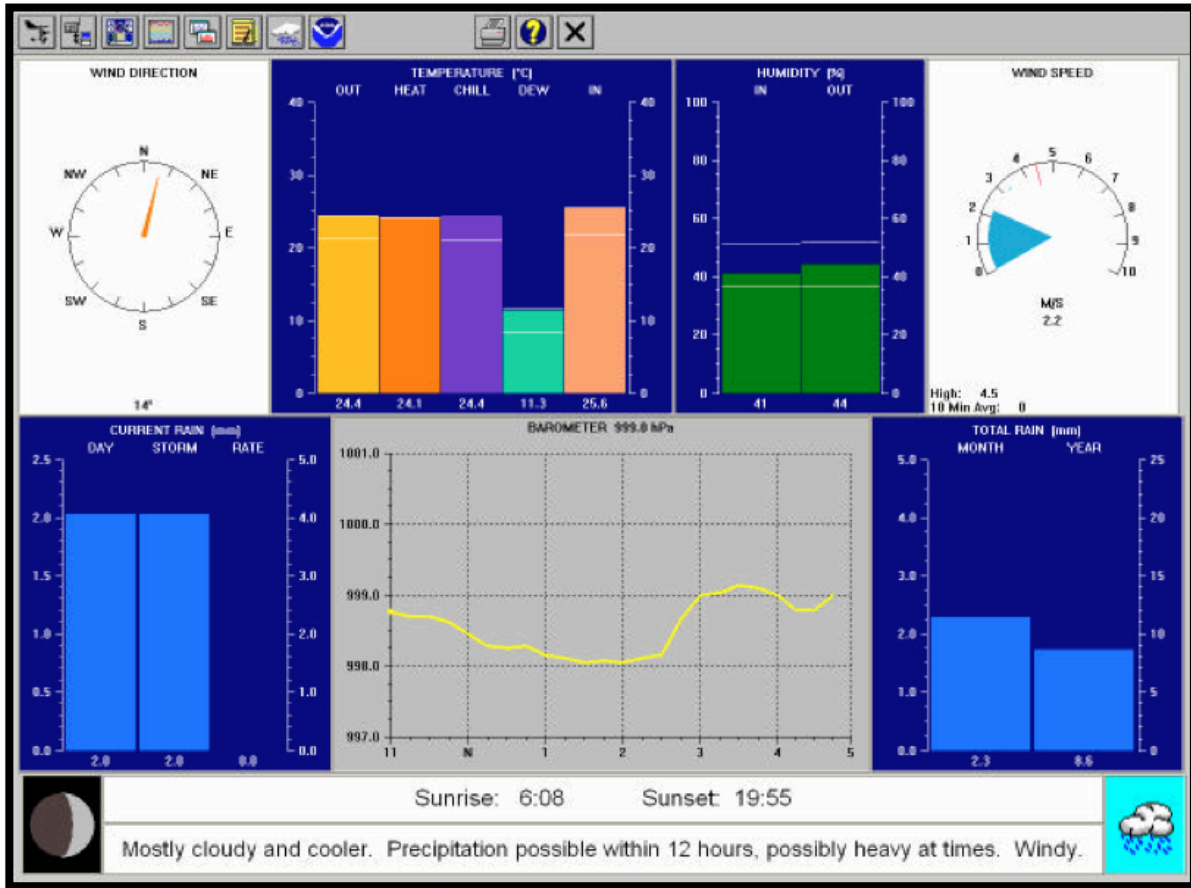


Imagen 2 Interfaz de la aplicación de monitoreo WeatherLink

El software del *WeatherLink* es compatible con computadoras que tengan Windows 2000 o posteriores. La cantidad de espacio necesario para los archivos de información dependerá del intervalo de los mismos archivos. Los archivos de la Base de datos que contengan información almacenada en un intervalo de 30 minutos requerirán, aproximadamente, 36K de espacio del disco por mes de información o datos. Por ejemplo, datos almacenados con un intervalo de 1 minuto necesitará 1 MB/ por mes, mientras que datos almacenados a intervalos de 2 horas requerirán, aproximadamente, 9 K / por mes.

Un *router* de banda ancha o “*switch*” con un puerto libre, una conexión a Internet “Always on” (siempre encendido). - Un DSL de alta velocidad, o cable para conexión con Internet y que está en conexión constante con Internet.



**Imagen 3 Estación meteorológica WeatherLink en funcionamiento
tomada del manual de usuario**

3.2.4 Parámetros meteorológicos :

Para estudiar la atmósfera nos valemos del elemento meteorológico y que se define como aquella variable atmosférica o fenómeno (temperatura del aire, presión, viento, humedad, tormentas, nieblas, ciclones o anticiclones, etc.) que caracteriza el estado del tiempo en un lugar específico y en un periodo de tiempo determinado, con los cuales se pueden desarrollar cálculos y predicciones, en la *Tabla 1* se muestran los parámetros capturados por la estación *WeatherLink* en la que se basa el presente trabajo (Dunkel, 2006).

Tabla 1 Parámetros capturados por la estación WeatherLink

Variable Ingles.	Variable español.	Unidades.	Indicación.
Date	Fecha	<i>Dia/Mes/Año</i>	Indica la Fecha de Captura.
Time	Hora	<i>Horas:Minutos</i>	Indica la Hora de Captura.
Temp. Out	Temperatura Exterior	°C	Indica la temperatura ambiente exterior.
Hi Temp	Temperatura Máxima	°C	Indica la temperatura máxima registrada.
Low Temp	Temperatura Mínima	°C	Indica la temp. mínima registrada.
Out Hum.	Humedad Exterior	%	Indica la cantidad

			de humedad en el exterior.
Dew Pt.	Punto de Rocío		Indica el Punto de Rocío.
Wind Speed	Velocidad de Viento	<i>m/s</i>	Indica la velocidad de viento.
Wind Dir.	Dirección de Viento	<i>N,S,E,O, etc ...</i>	Indica la Dirección del viento
Wind Run	Carrera del viento	<i>Milla de Viento ó Kilometro de Viento.</i>	Indica la cantidad de viento.
Hi Speed	Velocidad Máxima	<i>m/s</i>	Velocidad máxima registrada.
Hi Dir.	Dirección Máxima	<i>N,S,E,O, etc..</i>	Dirección máxima registrada.
Wind Chill	Enfriamiento del Viento	<i>°C</i>	Indica el enfriamiento.
Heat Index	Índice de Calor		
THW Index	Índice THW		Indica el Índice de percepción.
THSW Index	Índice THSW		Indica el Índice de percepción.
Bar	Presión Barométrica	<i>Bar</i>	Presión atmosférica expresada en Bares.
Rain	Lluvia	<i>mm.</i>	Cantidad de Lluvia.
Rain Rate	Intensidad de	<i>mm.</i>	

	Lluvia		
Solar Rad.	Radiación Solar	w/m^2	Indica la cantidad de radiación solar.
Solar Energy	Energía Solar	w/m^2	Indica la cantidad de Energía Solar.
Hi-Solar Rad.	Radiación Solar Máxima	w/m^2	Indica la cantidad de Radiación solar Máxima registrada.
UV Index	Índice UV		Indica el índice de rayos UV.
UV Dose	Dosis de UV		Indica la porción de rayos UV.
Hi UV	UV Máxima		Indica la Cantidad de rayos UV máximos registrados.
Heat D-D	Calor D-D	$^{\circ}C$	Indica el calor de los Grados-Día.
Cool D-D	Frio D-D	$^{\circ}C$	Indica el Frio de los Grados-Día.
In Temp.	Temperatura Interior	$^{\circ}C$	Indica la temperatura en los interiores.
In Hum.	Humedad Interior	w/m^2	Indica la cantidad de humedad que hay en los interiores

In Dew	Roció Interior		Indica el punto de rocío en los interiores.
In Heat	Calor Interior		Indica la cantidad de calor en los interiores.
In EMC	EMC Interior		Indica la cantidad del EMC en los interiores.
In-Air Density	Densidad de Aire Interior	Kg/m^3	Indica la densidad del aire en los interiores.
Temp 2nd	Segunda Temperatura.	°C	Segunda Temperatura.
Hum 2nd	Segunda Humedad.	°C	Segunda Humedad.
ET	Evapotranspiración	<i>mm.</i>	Cantidad de Evapotranspiración.
Leaf Wet 1	Hoja Húmeda 1		Indica la humedad de hoja registrada por el sensor 1.
Leaf Wet 2	Hoja Húmeda 2		Indica la humedad de hoja registrada por el sensor 2.
Leaf Temp. 1	Temperatura de Hoja 1	°C	Indica la temperatura de hoja registrada por el

			sensor 1.
Leaf Temp. 2	Temperatura de Hoja 2	°C	Indica la temperatura de hoja registrada por el sensor 2.
Wind Samp	Vientos Muestra	<i>Cantidad</i>	Cantidad de Muestras.
Wind Tx	Canal de recepción.	Canal.	Canal de Recepción.
ISS Recept	Recepción ISS	%	Porcentaje de Recepción.
Arc. Int.	Intervalo de Archivos	#	Archivos.

2.4 Observaciones Meteorológicas:

La observación meteorológica consiste en la medición y determinación de todos los elementos que en su conjunto representan las condiciones del estado de la atmósfera en un momento dado y en un determinado lugar mediante la utilización de un instrumental definido, estas observaciones realizadas con métodos y en forma sistemática, uniforme, continua y a horas establecidas, permiten conocer las características y variaciones de los elementos atmosféricos, los cuales constituyen los datos básicos que utilizan los servicios meteorológicos, tanto en tiempo real como diferido (Dunkel, 2006).

Las observaciones deben hacerse, invariablemente, a las horas pre-

establecidas y su ejecución tiene que efectuarse en el menor tiempo posible. Es importante que el observador preste preferente atención a estas dos indicaciones, dado que el incumplimiento propiciaría que no sean sincrónicas con observaciones tomadas en otros lugares, por la continua variación de los elementos que están bajo observación a la obtención de datos, que por ser tomados a distintas horas o por haberse demorado demasiado en efectuarlos (José María Escudero, 2008). La veracidad y exactitud de las observaciones es imprescindible, puesto que de no darse esas condiciones se lesionan los intereses, no solo de la meteorología, sino de todas las actividades humanas que se sirven de ella. En este sentido, la responsabilidad del observador es mayor de lo que generalmente el mismo supone.

3.3.3 Herramientas de desarrollo de software

3.3.3.1 lenguaje SQL

El lenguaje SQL se diseñó fundamentalmente para las aplicaciones de ayuda a la toma de decisiones, que son intensivas en consultas, mientras que el objetivo principal de las bases de datos en la década de 1980 fueron las aplicaciones de procesamiento de transacciones, que son intensivas en actualizaciones. La ayuda a la toma de decisiones y las consultas reemergieron como una importante área de aplicación para las bases de datos. Las herramientas para analizar grandes cantidades de datos experimentaron un gran crecimiento de uso.

Muchos vendedores de bases de datos introdujeron productos de bases de datos paralelas en este periodo, así como también comenzaron ofrecer bases de datos relacionales orientadas a objeto. El principal acontecimiento fue el crecimiento explosivo de *World Wide Web*. Las bases de datos se implantaron mucho más extensivamente que nunca antes. Los sistemas de bases de datos tienen ahora soporte para tasas de transacciones muy altas, así como muy alta fiabilidad y disponibilidad 24 horas al día y 7 días a la semana. Aunque el lenguaje SQL se considere un lenguaje de consultas, contiene muchas otras capacidades además de la consulta en base de datos, incluye características para definir la estructura de los datos y especificación de restricciones de seguridad (Abraham Silberschatz, 2002).

3.3.3.2 Sistema gestor de bases de datos

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben

proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos. dado que la información es tan importante en la mayoría de las organizaciones, los científicos informáticos han desarrollado un amplio conjunto de conceptos y técnicas para la gestión de los datos (Abraham Silberschatz, 2002).

3.3.3 Apache 2

El servidor HTTP Apache es un servidor web HTTP de código abierto multiplataforma para Microsoft Windows, Macintosh, Linux, entre otros, mediante la utilización del protocolo HTTP/1.1. Apache Server ha evolucionado desde que comenzó su desarrollo en 1995, con la incorporación de funcionalidades que son fruto del trabajo y aportaciones de Apache Software Fundación una comunidad creada para dar soporte a los proyectos de software bajo la denominación Apache (Kabir, 2002).

Apache tiene una buena aceptación en la web, desde 1996 es el servidor http mas utilizado debido a su seguridad, confiabilidad y la calidad del código que va dentro del producto y es utilizado principalmente para alojar páginas web estáticas y dinámicas en la Word Wide Web (Rich Bowen, 2002). Muchas aplicaciones web están diseñadas e implementadas con Apache, o que utilizarán características propias de este servidor web. Una de las principales características que presenta apache es que funciona en plataformas virtuales muy utilizadas; al principio, Apache se utilizaba para ser el primer servidor Web basado en Unix, que , además, funciona en Windows y en muchos otros sistemas operativos (Kabir, 2002).

3.3.3.4 Base de Datos

Una base de datos es una colección ordenada de datos, que normalmente se almacenan en uno o más archivos asociados. Los datos se estructuran en forma de tablas, en las referencias cruzadas entre las tablas son posibles. La existencia de tales relaciones entre las tablas conduce a la base de datos se llama una base de datos relacional.

Una base de datos puede ser una lista de direcciones que residen en un programa de hoja de cálculo, o puede ser los archivos de administración de una empresa de telecomunicaciones en el que varios millones de llamadas están registrados diariamente, sus cargas con precisión calculada, facturas mensuales calculado, y envió cartas de advertencia a los que están en mora. Una base de datos simple puede ser una operación independiente, mientras que otros pueden ser utilizados simultáneamente por miles de usuarios, con los datos distribuidos entre varias computadoras y decenas de discos duros. El tamaño de una base de datos puede variar desde unos cuantos kilobytes a varios terabytes.

En el uso ordinario, la palabra "base de datos" se utiliza para referirse a los datos reales, los archivos de base de datos resultante, el sistema de base de datos, o de un cliente de base de. Así surge un gran potencial para la confusión en cuanto a dos personas comienzan a conversar sobre el tema de bases de datos (Kofler, 2005) .

3.3.5 Ruby

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, inventado por el japonés Yukihiro Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente dos años más tarde.

Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

Ruby es orientado a objetos: todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivos, (como enteros, booleanos, y nulos). Toda función es un método. Las variables siempre son referencias a objetos, no los objetos mismos. Ruby soporta herencia con enlace dinámico, *mixins* y métodos singleton². A pesar de que Ruby no soporta herencia múltiple, las clases pueden importar módulos como mixins. La sintaxis procedural está soportada, pero todos los métodos definidos fuera del ámbito de un objeto son realmente métodos de la clase Object. Como esta clase es padre de todas las demás, los cambios son visibles para todas las clases y objetos. (Matsumoto, 2000).

² Herencia de clase pertenecientes y definidos por un sola instancia

La popularidad de este lenguaje radica en que reúne las características de los mejores lenguajes (Java, C, Python, entre otros) dentro de un relativamente sencillo lenguaje de programación. Dichas características son:

Características:

- Posibilidad de realizar directamente llamadas al sistema operativo.
- Potentes operaciones sobre cadenas de caracteres y expresiones regulares.
- Las variables son de tipo dinámico.
- La gestión de la memoria es automática.
- Modelo de procesamiento de excepciones.
- Carga dinámica.
- Clases, herencia y métodos innatos.
- Orientado a objetos.
- Cuatro niveles de ámbito de variable: global, clase, instancia y local.
- Manejo de excepciones.
- posibilidad de redefinir los operadores (sobrecarga de operadores).
- recolección de basura automática.
- altamente portable.
- Hilos de ejecución simultáneos en todas las plataformas.
- Carga dinámica de DLL/bibliotecas compartidas.
- introspección, reflexión y meta programación.
- amplia librería estándar.
- soporta inyección de dependencias.
- soporta alteración de objetos en tiempo de ejecución.
- continuaciones y generadores.

Ruby es un lenguaje de programación dinámico, con una gramática compleja, pero expresiva y una librería base de clase con una API rica y poderosa.

Ruby se inspira en *Smalltalk*, y *Perl*, pero utiliza una gramática que es relativamente fácil aprender. Una de las características más notables es su arquitectura orientada fuertemente a objetos, ya que desde su arquitectura en *Ruby* todo es un objeto, pero también es adecuado para los estilos de programación de procedimiento y funcionales e incluye potentes capacidades de (Yukihiro Matsumoto, 2008).

3.3.6 Ruby on Rails 3

Ruby on Rails, también conocido como RoR es un *framework* de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby y la integración del paradigma de la arquitectura Modelo Vista Controlador (*MVC*). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones con menos líneas de código que con otras alternativas y un mínimo de configuración. El lenguaje de programación Ruby permite la meta programación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de *RubyGems*, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones *Ruby*.

Ruby es una parte clave del éxito de *Rails*, porque lo utiliza para crear lo que se llama un lenguaje específico de dominio (*DSL*). En este caso, el dominio es el del desarrollo web, cuando se trabaja en *Rails* es casi como si estuviera escrito en un idioma que fue diseñado específicamente para la construcción de aplicaciones web-un lenguaje con su propio conjunto de reglas y gramática. (Cloves Carneiro, 2010)

Rails es una plataforma de aplicaciones web para el lenguaje de programación Ruby. Rails está bien pensado y práctico: le ayudará a crear sitios web de gran alcance con rapidez, con el código que está limpio y fácil de mantener. Open source y multiplataforma

3.3.7 Modelo MVC

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

En el paradigma *MVC* (Modelo-Vista-Controlador) la entrada del usuario, los modelos y la información visual al usuario están explícitamente separadas y manejado por tres tipos de objetos, cada uno especializado para su tarea.

La vista administra la salida gráfica y/o textual a la porción de la pantalla de mapa de bits que se asigna a su aplicación. El controlador interpreta las entradas del ratón y el teclado por parte del usuario, al mando del modelo y/o el fin de cambiar según sea apropiado. Por último, el modelo gestiona el comportamiento y los datos del dominio de aplicación, responde a las solicitudes de información acerca de su estado (por lo general de la vista), y responde a las instrucciones para cambiar de estado (por lo general desde el controlador). La separación formal de estas tres tareas es un concepto importante que es especialmente adecuado para Smalltalk-80 donde se encuentra el comportamiento básico plasmado en objetos abstractos: Vista, Controlador, Modelo y Objeto. El comportamiento MVC es entonces heredado, añadido a, y modificarse según sea necesario para proporcionar un sistema flexible y potente. (Burbeck, 1992).

3.3.8 ORM

El ORM (Mapeo Objeto relacional) es la técnica de programación que permite transformar los datos entre el sistema de tipos utilizados en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia (Kevin Marshall, 2007). Lo que da como resultado una base de datos orientada a objetos virtual sobre la base de datos relacional. Esta transformación permite el uso de las ventajas de la programación orientada a objetos especificada en los modelos y construir la base de datos relacional utilizando un solo esquema como se muestra en la *Imagen 4*, a diferencia del modelo tradicional en donde se crea el esquema de la aplicación (modelos y clases) y el de la base de datos.

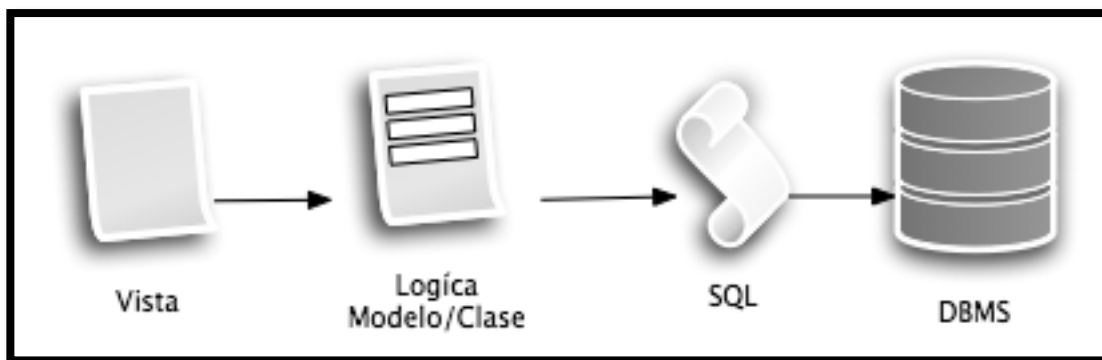


Imagen 4 Flujo del método ORM (Mapeo Objeto Relacional)

El Mapeo Objeto relacional, se basa en el Modelo Relacional, un patrón de datos que utiliza la lógica de predicado y la teoría de conjuntos para la gestión de una base de datos. Siguiendo este modelo se puede construir una base de datos relacional, un conjunto de una o más tablas estructuradas en registros (filas) y campos (columnas), que se vinculan entre sí por un campo en común (Busto, 2012). Sin embargo, en el Modelo Orientado a Objetos en una única entidad denominada objeto, se combinan las estructuras de datos con sus comportamientos. En este modelo se destacan conceptos básicos tales como objetos, clases y herencia.

Algunas de las ventajas de utilizar el mapeo objeto relacional son las siguientes :

- Abstracción de la base de datos. Al utilizar un sistema ORM, permite separar la lógica del sistema de base de datos, y así si en un futuro debemos de cambiar de motor de bases de datos, tendremos la seguridad de que este cambio no nos afectará a nuestro sistema, siendo el cambio más sencillo.
- Reutilización. Nos permite utilizar los métodos de un objeto de datos desde distintas zonas de la aplicación, incluso desde aplicaciones distintas.
- Seguridad. Los ORM suelen implementar sistemas para evitar tipos de ataques como pueden ser los SQL inyecciones.
- Mantenimiento del código. Nos facilita el mantenimiento del código debido a la correcta ordenación de la capa de datos, haciendo que el mantenimiento del código sea mucho mas sencillo.
- Lenguaje propio para realizar las consultas. Estos sistemas de mapeo incluyen su propio lenguaje para crear las consultas, lo que hace que los usuarios dejen de utilizar la sentencias SQL para que pasen a utilizar el lenguaje propio de cada herramienta.

Capítulo IV

METODOLOGÍA

4.0 Introducción

Este capítulo tiene como objetivo exhibir y explicar la metodología que se utilizará para el desarrollo del presente trabajo de tesis, con la descripción de cada una de las actividades necesarias para el desarrollo del análisis preliminar y metodología de desarrollo, otro punto importante es la descripción de las herramientas de desarrollo de software que se utilizarán y cual será su propósito para el desarrollo de la aplicación.

4.1 - Investigación preliminar

La etapa de captura y representación tiene como objetivo conocer el estado actual del proceso, donde se usan distintas fuentes de información, la cual consiste en los pasos que se muestran en la *Tabla 1*, en el caso particular de esta investigación, se consideraron los documentos generados durante los proyectos de cursos anteriores, entrevistas a participantes en dichos proyectos y revisión de las notas del curso:

Tabla 2 Descripción de pasos en la investigación preliminar

#	Pasos	Descripción
1	Revisión de documentación.	Se estudia documentación de software, hardware y procesos utilizados en el presente proyecto con el propósito de obtener un panorama de las funciones y condiciones.
2	Entrevistas semi estructuradas.	Buscar respuesta a cuestiones sobre el funcionamiento, requerimientos y objetivos de los procedimientos de consultas de datos con los encargados del área de energías renovables en CITEC.
3	Valoración de aspectos Socio-Técnicos	Revisión de los aspectos socio-técnicos involucrados del proceso de acceso a los datos meteorológicos.

Asimismo, los datos obtenidos en la investigación de campo son primarios, es decir, de primera mano a través del método de la observación en donde los investigadores analizaron y estudiaron la situación en su ambiente real, según explica (*Carlos Sabino, 2002*).

4.2 - Metodología de desarrollo

La metodología de desarrollo de software que se contemplara el desarrollo del presente proyecto es la metodología en cascada, en base a cada una de sus etapas como se muestran la *Imagen 5* (I. Somerville, 2004).

1. **Análisis del Sistema:** Debido a que el software es siempre parte de un sistema mayor el trabajo comienza con establecer los requisitos de todos los elementos del sistema y luego integrar un subconjunto de estos requisitos al software, se utilizará la información recabada durante la investigación preliminar.

2. **Análisis de los requisitos del software:** el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

3. **Diseño:** el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.(P. López, 2006)

4. **Codificación:** el diseño debe traducirse en una forma legible para la maquina.

El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

5. **Pruebas:** una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, a través de pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

- Pruebas de caja Blanca : Sera utilizada para revisar detalles procedimentales del software, con un enfoque fuertemente ligado al código fuente
- Pruebas de caja Negra : Sera utilizada para corroborar que las entradas y salidas son correctos así como el funcionamiento del sistema completo.

6. **Ajustes:** el software sufrirá cambios después de la implementación los cuales ocurrirán debido a que a la incorporación de nuevas estaciones meteorológicas y las necesidades de los usuarios finales, los sistemas de acceso a datos registrados será incorporada al sistema aural con el fin de obtener información optimizada, en términos de precisión.

Al usar el modelo de cascada, es necesario apreciar un conjunto de tareas en forma de fase para después continuar con la fase próxima. El modelo en cascada trabaja perfectamente para los proyectos en los cuales los requisitos del proyecto se encuentran definidos claramente y no son obligados a futuras modificaciones adicionales que fueron previa mente planteados durante la adaptación del prototipo inicial.

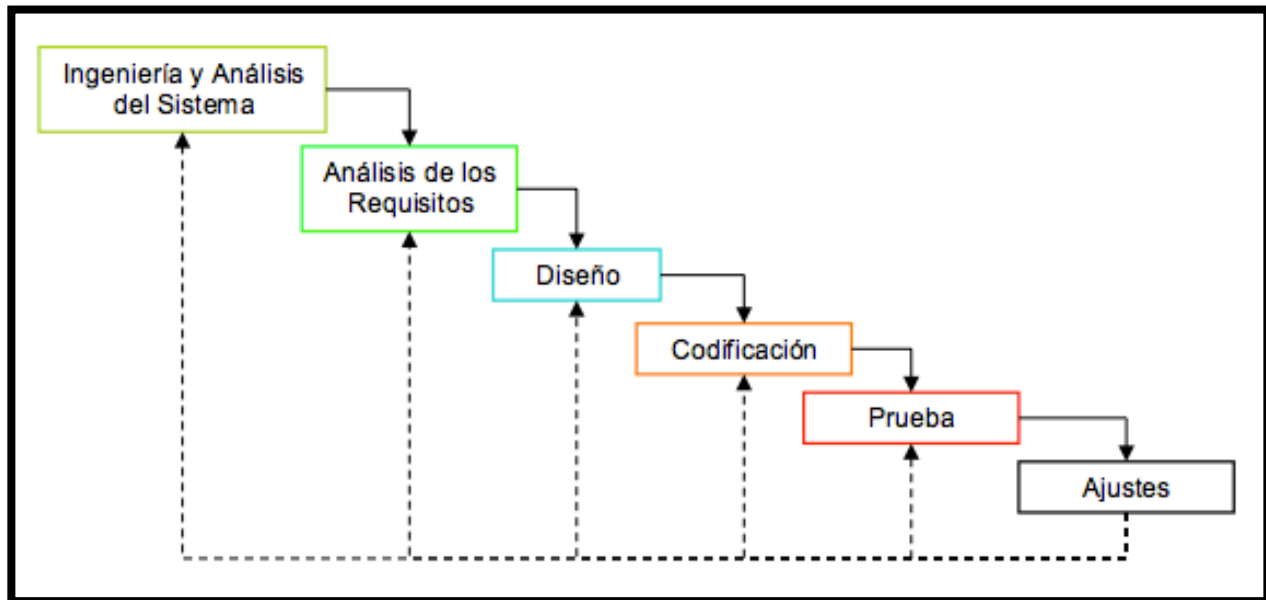


Imagen 5 Re presentación grafica de las etapas del modelo de desarrollo de software en cascada

4.3 Elección de tecnologías para el proyecto

Con las consideraciones expuestas anteriormente durante la investigación preliminar y metodología de desarrollo se considera que las tecnologías adecuadas para realizar el presente proyecto, las cuales se pueden observar en la *Tabla 4.2*, y el rol dichas tecnologías se describe a continuación :

Como servidor web se opto por Apache Server 2, que es de libre distribución y es también es actualmente utilizado en el CITEC, como lenguaje de programación para programar la aplicación en el servidor se elige *Ruby* con la plataforma *Rails 3*, que es un lenguaje potente y seguro y fuertemente orientado a objetos, Para la interfaz se integrara la tecnología *Responsive Design*³ con el

³ El diseño adaptable (en inglés, Responsive Design) es una técnica de diseño que mediante el uso de estructuras y hojas de estilo consigue adaptar el sitio web al entorno del usuario.

propósito de aumentar la usabilidad y disponibilidad con una interfaz que se adaptara a varios dispositivos de computo diferentes, sea una computadora personal o un teléfono inteligente, y de esta manera poder acceder ala información de manera sencilla, como servidor de base de datos se utilizara la tecnología *MySQL* el cual tiene una amplia compatibilidad con el servidor *Apache* y el lenguaje de programación *Ruby*.

Se utilizara una arquitectura modelo, vista, controlador (MVC) para desarrollar la aplicación, donde el sistema estará separado en módulos para facilitar el desarrollo paralelo de la funcionalidad del software, la codificación del sistema será basada en los modelos obtenidos basados en la investigación preliminar, y el diseño de la interfaz de parte del cliente se tomara como referencia los resultados de las análisis de requerimientos de usabilidad el cual consiste el modelado de las interfaces de usuario de sistema en desarrollo para posteriormente implementarlo en base a los resultados que se obtuvieron.

Tabla 3 Lista de herramientas involucradas en el desarrollo del proyecto

Herramientas de desarrollo.	
Apache 2	ORM
HTML5	JavaScript
Arquitectura MVC	GIT
CGI	MySQL
SSL	Active Records
CSS3	Active Relations
Ruby	
Rails 3	

Capítulo V

DESARROLLO

5.1 Desarrollo

La etapa de desarrollo, siguiendo el flujo del modelo en cascada (*Imagen 5*) comenzó con el análisis del sistema, la revisión de la documentación de la estación *WeatherLink*, así como su interacción con el servidor encargado de capturar los valores de las variables enviados por dicha estación, posteriormente se examinó el *software* de gestión de información incluido en el paquete de la estación (*Imagen 2*), donde se evaluó la opción de utilizar dicho sistema para envío de parámetros, se observó que carece de opciones de personalización y escalabilidad al ser una aplicación local y además pre compilada, por lo cual se descarta la posibilidad de re utilizarlo en la implementación.

La estación utiliza archivos de texto como medio exportación, en contraste a las observaciones obtenidas durante la entrevista con los coordinadores del área de energías renovables y los objetivos específicos del presente proyecto se definió que el sistema tenía que ser capaz brindar acceso vía web y soportar la inclusión de más estaciones meteorológicas en el futuro para lo cual es necesario agregar un sistema de base de datos.

Posteriormente, durante el análisis de requerimientos se asignó por parte de CITEC un servidor Dell con las características especificadas en la *Tabla 4*, donde se revisó que contara con los requerimientos necesarios para poder soportar sin problemas cada una de las herramientas de desarrollo definidas en la *Tabla 3*.

Tabla 4 Especificaciones del servidor utilizado para la instalación del sistema web

Modelo Optiplex GX520	
Sistema Operativo	Windows 7
Procesador	3.40GHz
Memoria RAM	4 GB
Capacidad del Disco duro	1 TB

Durante la etapa de diseño se revisan las características de componentes potenciales para ser integrados como módulos al sistema, con el objetivo de reutilizar y agilizar el proceso, para lo cual es necesario seleccionar herramientas de utilidad y que funcionen correctamente en la etapa de codificación, las principales se muestran en la *figura 14* y se describen en la *tabla 5*.

El proceso de codificación del sistema web comienza con la instalación del lenguaje de programación, gestor de versiones, manejador de base de datos y cada una de las herramientas especificadas en la *tabla 3*, posteriormente la configuración, codificación y publicación del sistema de información web.

El primer paso es entrar al portal oficial del lenguaje de programación Ruby como se observa en la *Imagen 6*, y bajar la versión correspondiente, Ruby 2.0.0 en el caso particular del presente proyecto.

La instalación de Ruby versión 2.0 se lleva a cabo en la ruta c:\Ruby200, el asistente provee las opciones recomendadas, instala y configura las rutas y dependencias de la aplicación presionando instalar y posteriormente finalizar.



Imagen 6 Sitio oficial con las últimas versiones del lenguaje de programación Ruby.

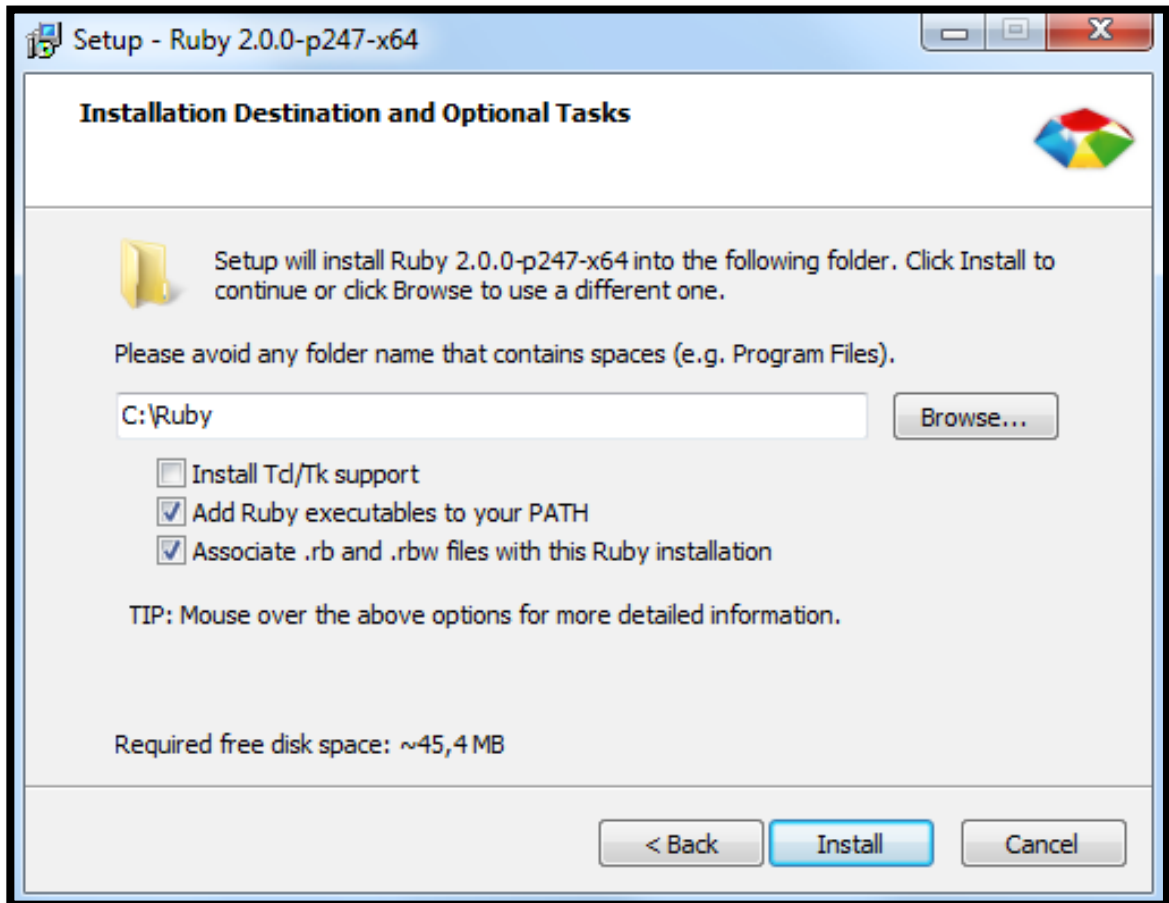


Imagen 7 Asistente de instalación del lenguaje de programación Ruby

1. Una vez instalado el lenguaje es posible instalar *bundler*⁴, digitando los siguientes comandos en la línea de comandos o terminal.

gem install bundler

2. Una vez instalado *bundler* es posible iniciar con la instalación de componentes y el primero es la plataforma de desarrollo *rails*, con el siguiente comando.

gem install rails

⁴ Gestor de rutas y versionamiento de componentes y extensiones de la aplicación.

Los componentes instalados en los pasos anteriores, admiten la creación del proyecto con el comando “rails new” y nombre del nuevo proyecto, así como con cada una de sus dependencias como se observa en la *Imagen 8*.

```
su@WIN-LBJPU9T8CLD ~/workspace
$ rails new estacion-citec
create
create  README
create  Rakefile
create  config.ru
create  .gitignore
create  Gemfile
create  app
create  app/controllers/application_controller.rb
create  app/helpers/application_helper.rb
create  app/mailers
create  app/models
create  app/views/layouts/application.html.erb
create  config
create  config/routes.rb
create  config/application.rb
create  config/environment.rb
create  config/environments
create  config/environments/development.rb
create  config/environments/production.rb
create  config/environments/test.rb
create  config/initializers
create  config/initializers/backtrace_silencers.rb
create  config/initializers/inflections.rb
create  config/initializers/mime_types.rb
create  config/initializers/secret_token.rb
create  config/initializers/session_store.rb
create  config/locales
create  config/locales/en.yml
create  config/boot.rb
create  config/database.yml
```

Imagen 8 Creación del proyecto en Rails

La inicialización del gestor de versiones *GIT* como se observa en la *Imagen 9*, además de proveer un control de los cambios y modificaciones de la aplicación, permite hacer *rollback*⁵ y clonar repositorios como instancias exactas y funcionales de la aplicación.

⁵ Operación que permite regresar la aplicación a una versión anterior, comúnmente utilizada como solución a problemas y errores durante la codificación.

```
su@WIN-LBJPU9T8CLD ~/Desktop/estacion-citec
$ git init
Initialized empty Git repository in c:/Users/su/Desktop/estacion-citec/.git/

su@WIN-LBJPU9T8CLD ~/Desktop/estacion-citec (master)
$ ls -la .git
total 8
drwxr-xr-x  9 su      Administ  4096 Dec 16 17:19 .
drwxr-xr-x 19 su      Administ  4096 Dec 16 17:19 ..
-rw-r--r--  1 su      Administ   23 Dec 16 17:19 HEAD
-rw-r--r--  1 su      Administ  157 Dec 16 17:19 config
-rw-r--r--  1 su      Administ   73 Dec 16 17:19 description
drwxr-xr-x 13 su      Administ  4096 Dec 16 17:19 hooks
drwxr-xr-x  3 su      Administ    0 Dec 16 17:19 info
drwxr-xr-x  1 su      Administ    0 Dec 16 17:19 objects
drwxr-xr-x  4 su      Administ    0 Dec 16 17:19 refs

su@WIN-LBJPU9T8CLD ~/Desktop/estacion-citec (master)
$
```

Imagen 9 Asistente de instalación para el gestor de versiones GIT.

La creación de una aplicación remota del repositorio por seguridad y disponibilidad se puede almacenar en un repositorio remoto, en el caso particular del presente proyecto se utiliza un repositorio local para el desarrollo y uno remoto para guardar cambios Imagen 11, y el componente *passenger* Imagen 10, permite que una versión del proyecto pueda ser puesta directamente en producción bajo un servidor http-apache 2.

```
$ gem install passenger
Fetching: daemon_controller-1.1.7.gem (100%)
Fetching: passenger-4.0.27.gem (100%)
Temporarily enhancing PATH to include DevKit...
Building native extensions. This could take a while...
Successfully installed daemon_controller-1.1.7
Successfully installed passenger-4.0.27
2 gems installed
Installing ri documentation for daemon_controller-1.1.7...
Installing ri documentation for passenger-4.0.27...
Installing RDoc documentation for daemon_controller-1.1.7...
Installing RDoc documentation for passenger-4.0.27...
```

Imagen 10 Instalación del componente *passenger* para la utilización de la aplicación en el servidor apache

repositorio remoto : <https://github.com/oscaryunue/Estacion-citeq>

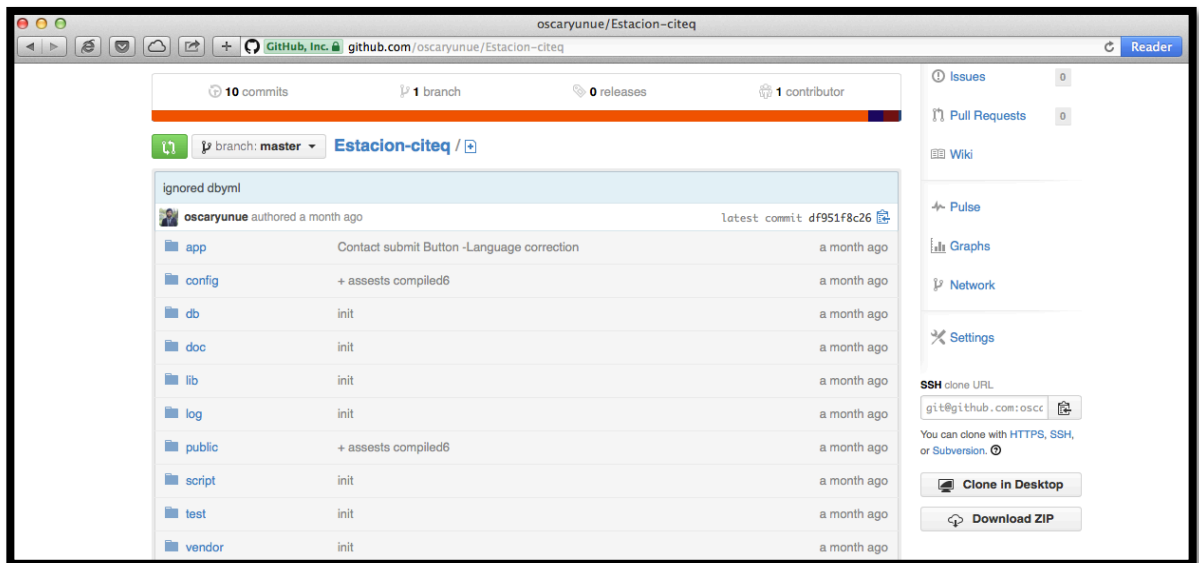


Imagen 11 Repositorio remoto utilizado para el desarrollo del sistema web.

Generar los modelos para estación y registro diario, seguido de la creación de la base de datos, el componente *ActiveRecord* se encarga de crear las tablas en la base de datos a partir de los modelos.

```
yunue:estacion-citec su$ rails g model station id nombre url rutaArchivo
  invoke  active_record
  create  db/migrate/20131211182023_create_stations.rb
  create  app/models/station.rb
  invoke  test_unit
  create  test/unit/station_test.rb
  create  test/fixtures/stations.yml
yunue:estacion-citec su$ rails g model Registro id fecha syncID
  invoke  active_record
  create  db/migrate/20131211182054_create_registros.rb
  create  app/models/registro.rb
  invoke  test_unit
  create  test/unit/registro_test.rb
  create  test/fixtures/registros.yml
yunue:estacion-citec su$ rake db:create
estacion1 already exists
yunue:estacion-citec su$ rake db:migrate
== CreateStations: migrating =====
-- create_table(:stations)
NOTICE: CREATE TABLE will create implicit sequence "stations_id_seq" for serial column "stations.id"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "stations_pkey" for table "stations"
-> 0.0173s
== CreateStations: migrated (0.0174s) =====

== CreateRegistros: migrating =====
-- create_table(:registros)
NOTICE: CREATE TABLE will create implicit sequence "registros_id_seq" for serial column "registros.id"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "registros_pkey" for table "registros"
-> 0.0077s
== CreateRegistros: migrated (0.0079s) =====
```

Imagen 12 La terminal muestra los comandos para la creación de los modelos y esquema de la base de datos.

Para disponer la aplicación modo desarrollo a producción el primer paso es establecer la variable de entorno “RAILS_ENV” en el estado producción como se observa en la *Imagen 13*, esta se toma como referencia para definir el modo en que se va a ejecutar la aplicación por parte del servidor http-apache2.

```
yunue:estacion-citec su$ set RAILS_ENV = production
yunue:estacion-citec su$
```

Imagen 13 Cambio de entorno de "development" a "production", para iniciar el servidor en modo producción

El ultimo paso para publicar la aplicación especificando el entorno de producción con el siguiente comando en la terminal o línea de comandos, entonces el sistema será puesto en funcionamiento como se observa en la Imagen 14, y es posible acceso a través de la dirección *URL* ⁶que este especificada en el servidor apache.

ruby s webrick -e production

```
=> Rails 3.2.13 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
[2013-12-11 07:47:58] INFO WEBrick 1.3.1
[2013-12-11 07:47:58] INFO ruby 2.0.0 (2013-05-14) [x86_64-darwin12.4.0]
[2013-12-11 07:47:58] INFO WEBrick::HTTPServer#start: pid=45938 port=3000
```

Imagen 14 Navegador muestra la aplicación en funcionando como *localhost* en el puerto 3000

⁶ Un localizador de recursos uniforme, más comúnmente denominado URL (sigla en inglés de *uniform resource locator*), para localizar un dirección particular en internet.

6 Pruebas

La fase de pruebas, mediante la técnica caja blanca para evaluar los componentes individuales del sistema y caja negra para probar el funcionamiento e interacción de los dichos componentes, trabajando como un sistema integral, complementado con una entrevista después de mostrar el funcionamiento básico del sistema con el personal de los encargados de energías renovables nos arrojan siguiente información.

6.1 Caja blanca

Este método de pruebas se designo a tres módulos como se observan en la *Tabla 4*, cada uno de los cuales desempeña un rol crucial en el funcionamiento del sistema de información presente y se añaden a esta como componentes externos de *Ruby Gems*, a continuación la prueba individual los componentes.

Tabla 5 Componentes utilizados para mejoras de interfaz, importación de datos y registros ala base de datos

#	Componente	Descripción
1	<i>rails_pitfall</i>	Integración para las vistas, CSS, JavaScript para las transiciones y creación de tablas dinámicas, y diseño responsivo.
2	<i>wlparams</i>	Abre el archivo de texto en modo lectura, identifica y ordena los datos capturados en el archivo de salida de la estación <i>WeatherLink</i>
3	<i>wltodb</i>	Migra los datos capturados por la aplicación al sistema de base de datos.

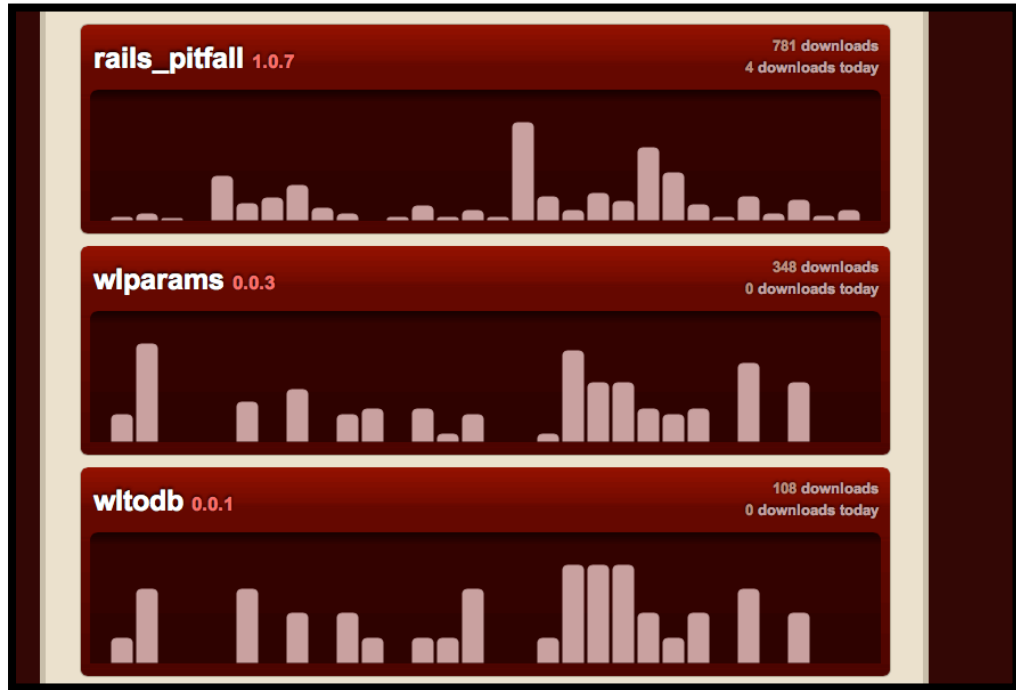


Imagen 14 Portal *rubygems.org* con repositorio de componentes utilizados en el presente proyecto.

Para iniciar con las pruebas del primer componente *rails_pitfall*, se puso el sistema de producción y se verifico cargaran correctamente los temas a partir del código HTML, además de el correcto funcionamiento de las tablas dinámicas sus controles como buscar y ordenar por parámetros, por ultimo el diseño adaptativo a diferentes resoluciones de pantalla como se observa en la Imagen 17.

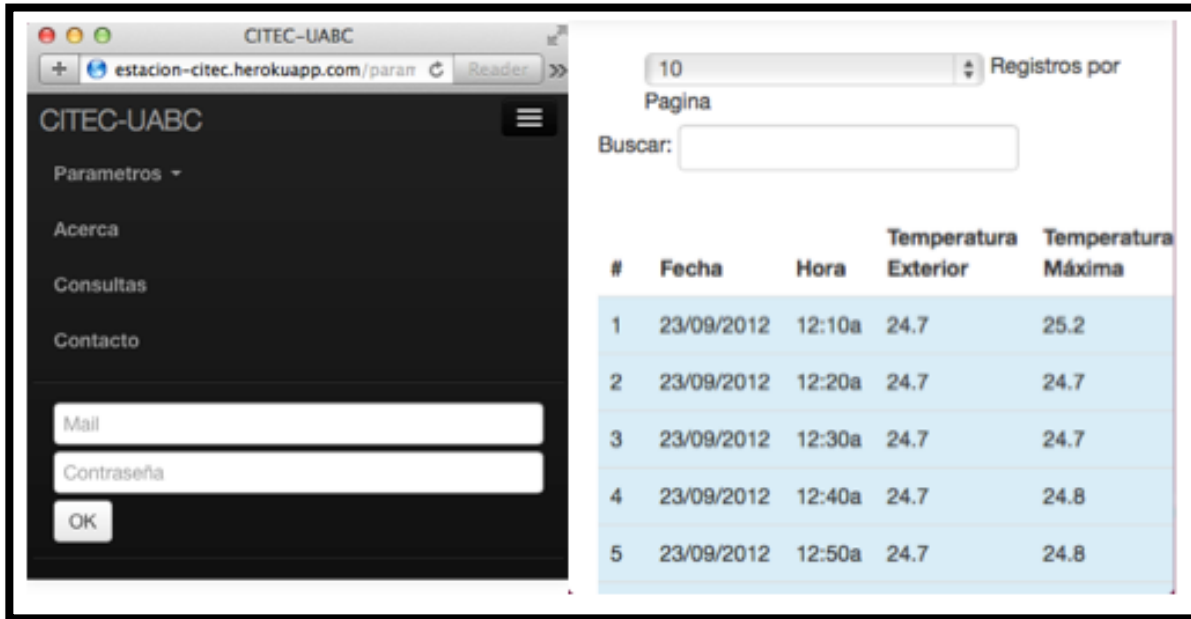


Imagen 15 Vista para despliegue en dispositivos móviles.

Para continuar con las pruebas de los siguientes dos componentes *wlparams* y *wltodb* se desarrollo una aplicación *Imagen 18*, con el propósito de corroborar el correcto funcionamiento al leer el archivo de texto en el que la estación meteorológica guarda los registros de las variables, desarrolla un análisis sintáctico, es decir, el proceso en el que se analizar una cadena de símbolos se identifica el tipo y se asigna a una variable dentro del sistema, lo que permite que se pueda utilizar en la presentación de datos y operaciones por parte del sistema.

```

1 ##### Estación Citec #####
2 #
3 # Launch this Ruby file from the command line
4 # to get started.
5 #
6
7 APP_ROOT = File.dirname(__FILE__)
8
9 # require "#{APP_ROOT}/lib/guide"
10 # require File.join(APP_ROOT, 'lib', 'guide.rb')
11 # require File.join(APP_ROOT, 'lib', 'guide')
12
13 $:.unshift( File.join(APP_ROOT, 'lib') )
14 require 'guide'
15
16 guide = Guide.new('Archivo.xls')
17 guide.launch!
18

```

Imagen 16 Código fuente de la aplicación *backend* utilizada para pruebas.

El registro periódico de la base de datos se puso a prueba con una terminal Unix comprobando que los registros se desarrollaran cada veinticuatro horas como se observa en la *Imagen 19*.

```

Comandos: list, find, add, quit
> list

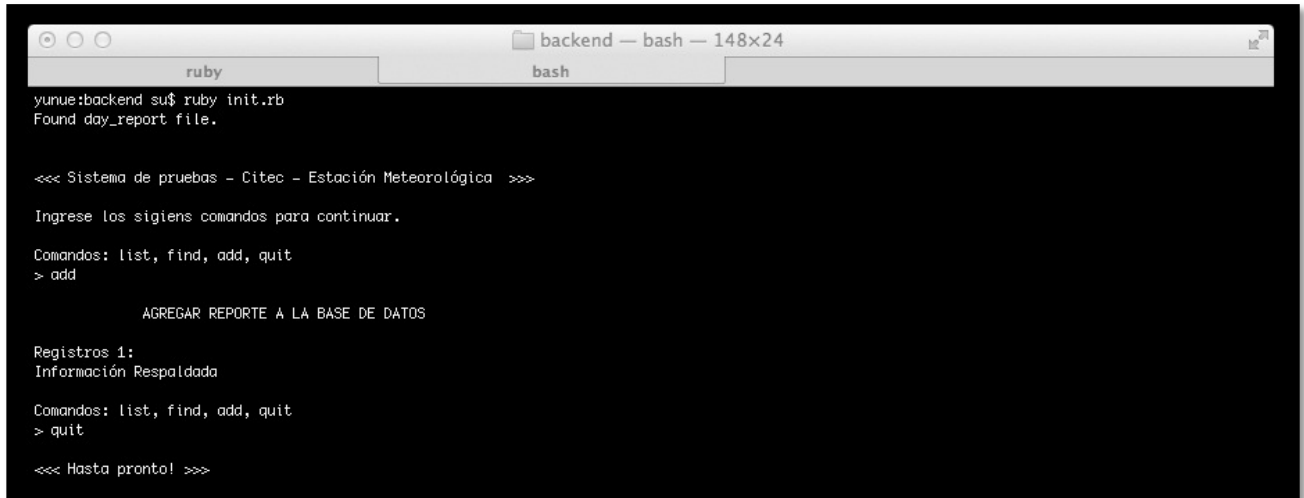
LISTANDO PARAMETROS DEL DIA


```

Wind	Heat	THW	THSW	Date	Rain	Rain	Rate	Solar	Solar	Hi	Low	Out	Dew	Wind	Wind	Wind	Hi	Hi
Chill	Index	Index	Index	Bar	Rain	Rate	Rad.	Time	Out	Temp	Temp	Hum	Pt.	Speed	Dir	Run	Speed	Dir
1	24.7	24.3	24.3	22.8	1011.8	0	0	12:10a	24.7	25.2	24.7	41	10.6	0	---	0	0	---
2	24.7	24.3	24.3	22.8	1011.8	0	0	12:20a	24.7	24.7	24.5	42	10.9	0	---	0	0	---
3	24.7	24.3	24.3	22.8	1011.8	0	0	12:30a	24.7	24.7	24.7	42	10.9	0	---	0	0	---
4	24.7	24.3	24.3	22.8	1011.9	0	0	12:40a	24.7	24.8	24.6	42	10.9	0	---	0	0	---
5	24.7	24.3	24.3	22.8	1011.8	0	0	12:50a	24.7	24.8	24.7	42	10.9	0	---	0	0	---
6	23.9	23.6	23.6	22.1	1011.9	0	0	1:00a	23.9	24.6	23.9	42	10.2	0	---	0	0	---
7								1:10a	23.9	23.9	23.9	43	10.6	0	---	0	0	---

Imagen 17 La aplicación muestra una lista ordenada de los datos extraídos del archivo de registro de la estación meteorológica.

Por ultimo probar el componente *wltodb* y verificar que efectivamente los parámetros se almacenan en el sistema de base de datos.



```
yunue:backend su$ ruby init.rb
Found day_report file.

<<< Sistema de pruebas - Citec - Estación Meteorológica >>>

Ingrese los sigiens comandos para continuar.

Comandos: list, find, add, quit
> add

          AGREGAR REPORTE A LA BASE DE DATOS

Registros 1:
Información Respaldata

Comandos: list, find, add, quit
> quit

<<< Hasta pronto! >>>
```

Imagen 18 La aplicación muestra la opción "*add*" encargada de guardar los registros extraídos del archivo en la base de datos.

6.3 Caja Negra:

Se puso a prueba la interacción del componente utilizando el sistema en producción y verificando el correcto funcionamiento de cada uno de los componentes captura de registros desde archivos texto, graficas, tablas de reportes y eventualmente la observación de la interacción transparente de cada uno de los módulos del sistema.

Fuente : *<http://estacion-citec.herokuapp.com>*

6.3 Resultado:

En base a las pruebas se observó que los componentes utilizados por la aplicación funcionan correctamente y cumplen satisfactoriamente con su objetivo individual y no se identificaron conflictos en la integración y funcionamiento global del sistema de información. Por otra parte, se realizó una entrevista a los coordinadores con el intención de evaluar la usabilidad y funcionalidad del sistema, y se detectaron posibilidades menores de mejora en el diseño y organización de los datos, el cambio más notable fue la clasificación de las mediciones por parte de la estación, dando un enfoque mayor a las variables primarias, los administradores del área de energías renovables mencionaron la importancia de mantener una perspectiva simple hacia los datos comunes, clasificándolos en primarios para las mediciones básicas, derivados para los que se calculan a partir de los parámetros primarios, y de estado, para aquellos que miden registran la calidad de los datos y conexión por parte de la estación.

7.1 Ajustes

Como se ha señalado en los resultados de las pruebas, una modificación importante fue el re ordenamiento de la tabla en la que se muestran el total de los parámetros registrados a lo largo de un día, desplegando por defecto los parámetros primarios y conservando el total de variables en una vista detallada de la misma tabla.

#	Fecha	Hora	Temperatura Exterior	Velocidad de Viento	Dirección de Viento	Enfriamiento del Viento	Índice THSW	Presión Barométrica	Lluvia	Radiación Solar	Índice UV	Evapotranspiración
1	23/09/2012	12:10a	24.7	10.6	0	---	24.3	22.8	1011.8	0	0	0
2	23/09/2012	12:20a	24.7	10.9	0	---	24.3	22.8	1011.8	0	0	0
3	23/09/2012	12:30a	24.7	10.9	0	---	24.3	22.8	1011.8	0	0	0
4	23/09/2012	12:40a	24.7	10.9	0	---	24.3	22.8	1011.9	0	0	0
5	23/09/2012	12:50a	24.7	10.9	0	---	24.3	22.8	1011.8	0	0	0
6	23/09/2012	1:00a	23.9	10.2	0	---	23.6	22.1	1011.9	0	0	0.05
7	23/09/2012	1:10a	23.9	10.6	0	---	23.7	22.2	1011.9	0	0	0
8	23/09/2012	1:20a	23.7	10.7	0	---	23.4	21.9	1011.9	0	0	0
9	23/09/2012	1:30a	24.1	10.7	0	---	23.8	22.3	1011.9	0	0	0
10	23/09/2012	1:40a	24.1	10.3	0	---	23.7	22.2	1012	0	0	0

Imagen 19 Tabla de parámetros después de ajuste, mostrando por default solo los parámetros primarios

Como se observa en la Imagen 22, se agregó una interfaz para dar de alta nuevas estaciones meteorológica y satisfacer uno de los objetivos específicos del presente sistema, concluyendo así con la etapa de ajustes.

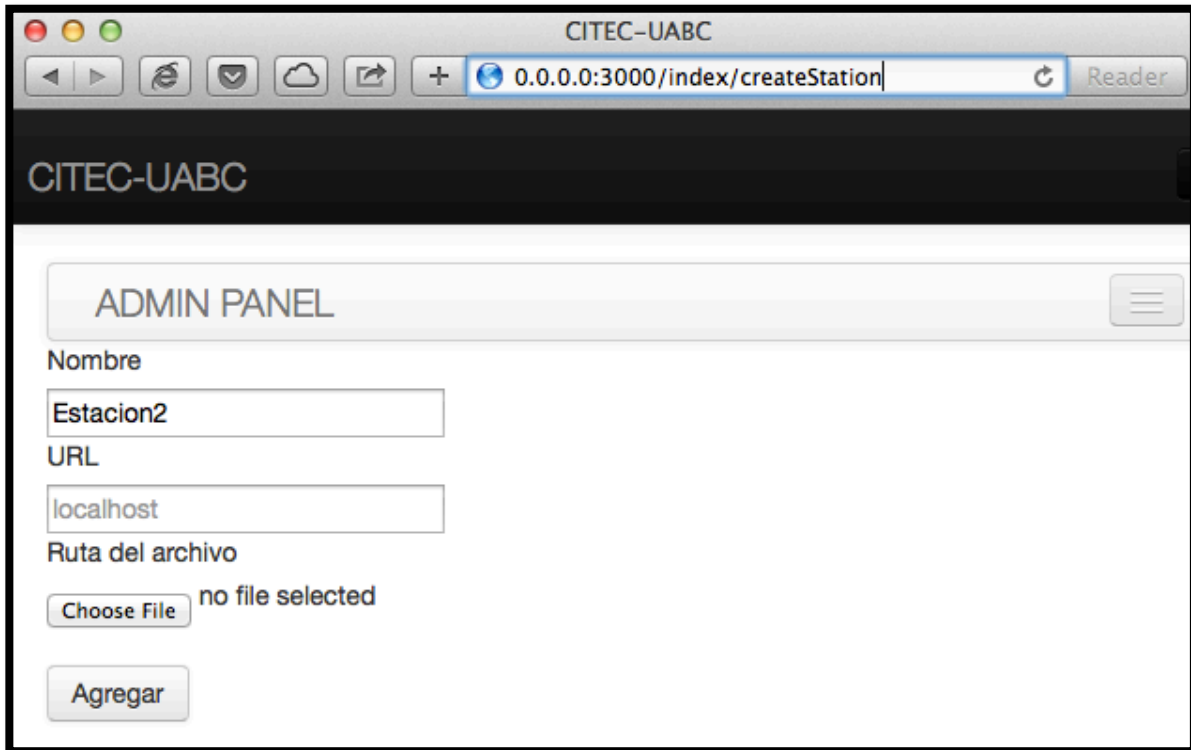


Imagen 20 Interfaz para agregar una nueva estación meteorológica.

Capítulo VI

RESULTADOS Y CONCLUSIONES

7.1 RESULTADOS

Durante el procedimiento de incorporación de funcionalidades, modelado de las interfaces y gestión de datos, se utilizaron componentes con el objetivo de satisfacer los requerimientos dictados por las normas de calidad de software, los cuales se ven reflejados en el sistema de información web, por ejemplo, en los procedimientos para importar, ordenar y almacenar los parámetros registrados por la estación meteorológica, enfocados en aspectos tales como la transparencia, usabilidad, escalabilidad, entre otros; y cumplir con los requisitos de un sistema de información eficiente.

El principal problema al que el analista se enfrenta, cuando se desea incorporar herramientas de usabilidad a los sistemas desarrollados o de un nuevo sistema, es el tiempo que se invierte en descubrir e incorporar las técnicas y herramientas durante el desarrollo, pero una vez implementados estos mecanismos van a reducir los tiempos en los que se desarrolla una tarea dentro del sistema por parte de los usuarios finales.

En lo que respecta a la integración de componentes se coincide en que, “Para la incorporación de mecanismos de usabilidad es recomendable utilizar componentes independientes pero funcionales, es decir, que cumplan con su objetivo, sin afectar la estructura principal del sistema, en el caso de las opciones de visualización es importante brindar flexibilidad en lo que respecta a los dispositivos de computo desde los cuales se puede acceder, la agregación de diseño responsivo es una solución eficiente para este problema debido a que se adapta a diferentes tipos de resolución sean computadores de escritorio o dispositivos móviles” (Quiroz, 2013).

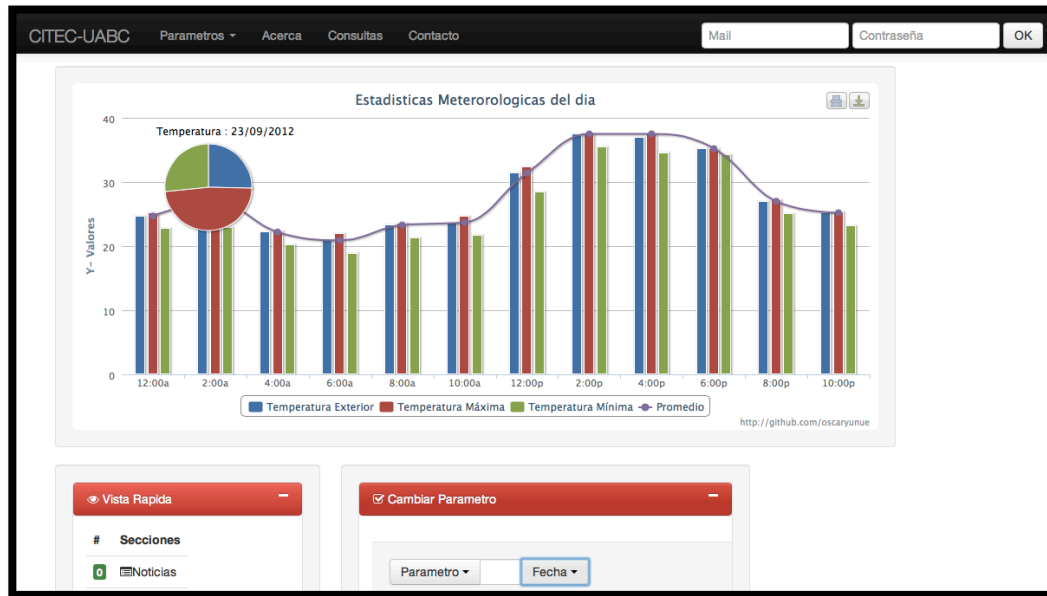


Imagen 21 Herramienta web de reporte de datos meteorológicos con componente de reporte grafico.

Los datos son la parte medular del sistema de información y basado en los requerimientos particulares del presente sistema se implemento un sistema de base de datos para el registro de las variables capturadas por la estación meteorológica *WeatherLink*, el *dynamic-type*⁷ es una característica que permite al sistema saber que tipo de dato se esta capturando y asignarle una variable correspondiente al parámetro que se esta capturando, entero, cadena de caracteres, punto flotante o fecha.

En el caso de las investigaciones meteorológicas en las cuales se procesas cantidades elevadas de parámetros como en el caso del presente proyecto donde intervenían ciento cuarenta y cuatro registros diarios, cada uno con cuarenta y cinco parámetros diferentes, se convierten en una limitan para los investigadores cuando se requiere consultar diferentes tipos de datos y fechas simultáneamente, como solución a este problema, se encontró que las librerías que permiten

⁷ Es la asignación automática de un de dato, a un tipo de variable en memoria por parte del lenguaje de programación, evitando tener que declarar un variable antes de usarla .

consultas dinámicas como *ActiveRelations* reducen el número de pasos y eventualmente tiempo para obtener la información solicitada por parte de los investigadores.

#	Fecha	Hora	Temperatura Exterior	Temperatura Máxima	Temperatura Mínima	Humedad Exterior	Punto de Rocío	Velocidad de Viento	Dirección de Viento	Carrera del viento	Velocidad Máxima	Dirección Máxima	Enfriamiento del Viento	Índice de Calor
1	23/09/2012	12:10a	24.7	25.2	24.7	41	10.6	0	---	0	0	---	24.7	24.3
2	23/09/2012	12:20a	24.7	24.7	24.5	42	10.9	0	---	0	0	---	24.7	24.3
3	23/09/2012	12:30a	24.7	24.7	24.7	42	10.9	0	---	0	0	---	24.7	24.3
4	23/09/2012	12:40a	24.7	24.8	24.6	42	10.9	0	---	0	0	---	24.7	24.3
5	23/09/2012	12:50a	24.7	24.8	24.7	42	10.9	0	---	0	0	---	24.7	24.3
6	23/09/2012	1:00a	23.9	24.6	23.9	42	10.2	0	---	0	0	---	23.9	23.6
7	23/09/2012	1:10a	23.9	23.9	23.9	43	10.6	0	---	0	0	---	23.9	23.7
8	23/09/2012	1:20a	23.7	24.1	23.7	44	10.7	0	---	0	0	---	23.7	23.4
9	23/09/2012	1:30a	24.1	24.1	23.7	43	10.7	0	---	0	0	---	24.1	23.8
10	23/09/2012	1:40a	24.1	24.2	24.1	42	10.3	0	---	0	0	---	24.1	23.7

Mostrando 1 al 10 de 143 registros

← Anterior 1 2 3 4 5 Siguiete →

Imagen 22 Lista de parámetros detallada aplicando herramientas de usabilidad.

7.2 Conclusión

En base a los resultados obtenidos se puede llegar a las siguientes conclusiones:

Utilizar una metodología en cascada permite invertir tiempo en el transcurso del análisis de requerimientos y diseño desde una perspectiva global, siendo necesaria una definición correcta al momento de la validación de estas etapas debido a la rigidez de dicha metodología, evitando tener que regresar y aplicar ajustes a etapas previamente completadas.

Por otra parte, la integración de componentes al sistema de información como se hizo con los archivos fuente para las vistas, de lectura de archivos y registros a la base de datos, así como la complejidad en la estructura del sistema reduciendo el riesgo de problemas de integración durante la etapa de codificación, así como el componente *Active Record* permite la creación, escritura, actualización y lectura a la base de datos con métodos de clase, es decir, con sintaxis simplificada y sin la utilización de lenguaje de consulta estructurado *SQL* dentro del código de la aplicación.

Por último, el control de versiones facilita el proceso de instalación del sistema web en producción, puesto que permite clonar una versión funcional de la aplicación desde un repositorio remoto.

La importancia de los puntos anteriores además de optimar el proceso de desarrollo del sistema, este puede generar consultas complejas y brinda flexibilidad en la presentación de datos y por tanto mejora la experiencia en la utilización de la herramienta por parte de los investigadores. con el fin de obtener un sistema de software de calidad. Este trabajo va encaminado en esa dirección.

7.3 Trabajo Futuro

La metodología aquí desarrollada, al igual que los componentes y datos recabados servirán para todo aquel interesado en mejorar la aplicación web y haciendo uso del gestor de versiones es posible agregar nuevas funcionalidades, herramientas de visualización e incorporar nuevas estaciones meteorológicas para el desarrollo de investigaciones mas precisas, utilizando como base el sistema actual.

Capítulo VII

BIBLIOGRAFIA

7.1. Bibliografía

- 1) Abraham Silberschatz, H. F. (2002). *Fundamentos de Bases de Datos - Cuarta Edición*. McGRAW HILL .
- 2) Abran, A. (2008). *Metrology, Software Metrics & Software*. New Jersey, E.U.A.: WILEY.
- 3) Busto, H. d. (2012). <http://revistatelematica.cujae.edu.cu>. Obtenido de Revista Telemática: <http://revistatelematica.cujae.edu.cu/index.php/tele/article/download/23/21>
- 4) Burbeck, S. (1992). *How to use Model-View-Controller*. Obtenido de <http://st-www.cs.illinois.edu>: <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
- 5) Carlos Gómez, L. P. (2007). *Diseño e implementación de estación meteorológica automática*.
- 6) CITEC. (2010). Obtenido de <http://citecuvp.tij.uabc.mx>: http://citecuvp.tij.uabc.mx/index.php?option=com_content&view=article&id=109&Itemid=151
- 7) Cloves Carneiro, R. A. (2010). *Beginning Rails 3*. Apress.
- 8) Fernández, L. A. (2013). <http://microsoft.com>. Obtenido de <http://msdn.microsoft.com/>: <http://msdn.microsoft.com/es-es/library/bb972232.aspx>
- 9) Gil, C. L. (2006). *Estimación de parámetros meteorológicos secundarios aplicando minería de datos*. Politécnico Jose Antonio Echeverria Habana, Cuba.
- 10) Ismael Guardado Dávila, I. P. (2007). *Monitoreo de una Estación Meteorológica vía internet*.
- 11) ISO 25010. (2011). *ISO/IEC 25010:2011*.

- 12) ISO 9000 . (2005). *ISO 9000:2005*. Suiza.
- 13) Kabir, M. J. (2002). *Apache Server 2 Bible*. Hungry Minds.
- 14) Kevin Marshall, C. P. (2007). *Pro ActiveRecors*. USA.
- 15) Kofler, M. (2005). *The Definitive Guide to MySQL5* . Apress.
- 16) Matsumoto, Y. (2000). *informit*. Obtenido de <http://www.informit.com>:
<http://www.informit.com/articles/article.aspx?p=18225>
- 17) O'Brien, J., & Marakas, G. (2006). *Sistemas de información gerencial*. McGraw Hill .
- 18) Rich Bowen, D. L. (2002). *Apache Administrator's Handbook* . Indianapolis, Indiana, E.U.A: SAMS.

ANEXOS

Entrevista semi estructurado para el análisis de requerimientos.

El objetivo de la siguiente secuencia de preguntas es el de identificar las necesidades así como las restricciones del sistema actual, expectativas de los usuarios finales del sistema para considerarlas como criterio en la definición de requerimientos, gestión de datos, y funcionalidad para ser tomados como referencia en el análisis de requerimientos, entrevista desarrollada a la coordinadora de energías renovables en UABC CITEC Dra. María Cristina Castañón Bautista.

I. ¿Qué hará el sistema?

El sistema será utilizado como herramienta por alumnos y encargados del área de energías renovables para desarrollar tareas e investigaciones.

II. ¿Existen varios modos de operación?

Los usuarios van a desarrollar las mismas tareas de acceso a los datos con la excepción del administrador que será el responsable de agregar las estaciones meteorológicas al sistema.

III. ¿Como y cuando puede cambiarse o mejorarse un sistema?

Que tiene que poder acceder los datos por internet, guardar los registros capturados por la estación meteorológica en una base de datos, que sea fácil de utilizar, que permite exportar datos y también que se puedan agregar más estaciones meteorológicas en el futuro porque se está planeando agregar más.

IV. ¿Existen restricciones de la velocidad de ejecución, tiempo de respuesta o rendimiento?

Los legisladores están capturando se guardan rápido y se puede acceder a ellos por el programa que ya tienen la estación *WeatherLink*, sin embargo el

sistema es poco flexible solamente se puede utilizar en la computadora la que está instalada la terminal, tiene otras limitaciones como que se vea un poco viejo no se pueden integrar más estaciones meteorológicas en un mismo sistema tanto nos han podido desarrollar tareas utilizando sus datos porque tendrían que todas las personas a estar consultando los datos directamente sobre la terminal lo que es poco práctico.

V. ¿Qué recursos materiales, personales o de otro tipo se requieren para construir, utilizar y mantener el sistema?

Solamente tener un servidor que pueda mandar la página en línea es decir un servidor que cuente con las especificaciones necesarias.

VI. ¿Qué habilidades deben tener los usuarios?

El sistema tiene que ser muy fácil de usar para que puedan hacer a ellos cualquier tipo de usuarios, porque comúnmente va ser utilizado por alumnos y coordinadores de energía renovables y otros usuarios que no son del área de sistemas por eso es conveniente que sea lo más práctico posible.

VII. ¿Cuanto espacio físico será ocupado por el sistema?

Solamente el servidor en el cual va a guardarse el sistema.

VIII. ¿Existe un cronograma prescrito para el desarrollo?

Todavía no.

IX. ¿Debe controlarse el acceso al sistema o a la información?

Sí que solamente tenga permiso usares autorizados en la parte de agregar más estaciones meteorológicas, fuera de ahí que pueda accesar cualquier usuario.

X. ¿Cuantos datos fluyen a través del sistema?

Son 46 registros, los que está guardando la estación cada 10 minutos.

XI. ¿Debe retenerse algún dato por algún periodo de tiempo?

Los datos que se registran deben de quedarse guardados siempre.

XII. ¿Para que se utiliza la estación meteorológica ?

Actualmente nos está utilizando más que para registrar los datos, percepciones a desarrollar investigaciones, estudios y tareas por parte de los alumnos de energías renovables.

Entrevista semi estructurada para validación.

La presente entrevista se desarrolla a la coordinadora de energías renovables en UABC CITEC Dra. María Cristina Castañón Bautista, con el objetivo de verificar que se cumplen con los objetivos planteados en el análisis de requerimientos.

I. ¿ Ha utilizado el sistema actual ?

Si

II. Se cumplen las expectativas en las siguientes categorías :

1. Gestión de información:

La gestión de información se desarrolla correctamente, desde la captura de los parámetros registrados por la estación y son guardados en un sistema de base de datos da mas confianza al utilizar, los datos se cargan rápido y por el sitio vía internet como se había planteado desde un inicio, y la opción para exportar datos es importante y útil también,

2. Usabilidad:

El es fácil de utilizar se puede acceder a los datos físicamente además las graficas facilitan la interpretación de los datos, tomando en cuenta que se están capturando registros cada diez minutos es mas fácil de verlos en una grafica, otro aspecto positivo es que se puede cambiar el diseño para acceder a la pagina donde esta los datos con *Smartphone*, lo cual da flexibilidad y practicidad para los usuarios.

3. Opinión general:

El sistema a grandes rasgos es practico y fácil de manejar pero debido a la

cantidad de parámetros, cuarenta y seis hay que buscar la forma de que se muestren solo los mas importantes.

III. En su opinión, cual es la mejor manera de manejar el volumen de parámetros.

Tal ves mostrando solo los primarios por default que son los mas importantes porque comúnmente son los que se utilizan por parte de los investigadores y alumnos además los demás suelen ser derivados de estos.

IV. Considera que se cumplen los objetivos planteados durante la definición de los objetivos del proyecto actual.

Si se cumplieron los objetivos que se plantearon en un inicio, pero es necesario hacer algunos ajustes como tener una forma de agregar una nueva estación meteorológica desde la pagina y reordenar la forma en la que se muestra los parámetros para cumplir completamente con los objetivos.

Apéndice

9.1. Procedimiento de instalación y configuración

El proceso empieza con la de cada uno de los servicios necesarios para el correcto funcionamiento del sistema, empezando con el interprete del lenguaje de programación Ruby, seguido por el servidor *Apache2*, el sistema gestor de base de datos *MySQL* y el gesto de versión GIT.

Características del servidor

9.1.1 Componentes de hardware

Las especificaciones del servidor asignado por CITEC para alojar el sistema web se pueden observar en la *Tabla 4*.

9.1.2 Componentes de Software

- Sistema operativo: Windows
- Ruby 2.0.0-p247
- Apache 2.4.4
- Git versión 1.8.4
- MySQL 5.5.3

9.1.3 Instalación de gestor de versiones GIT.

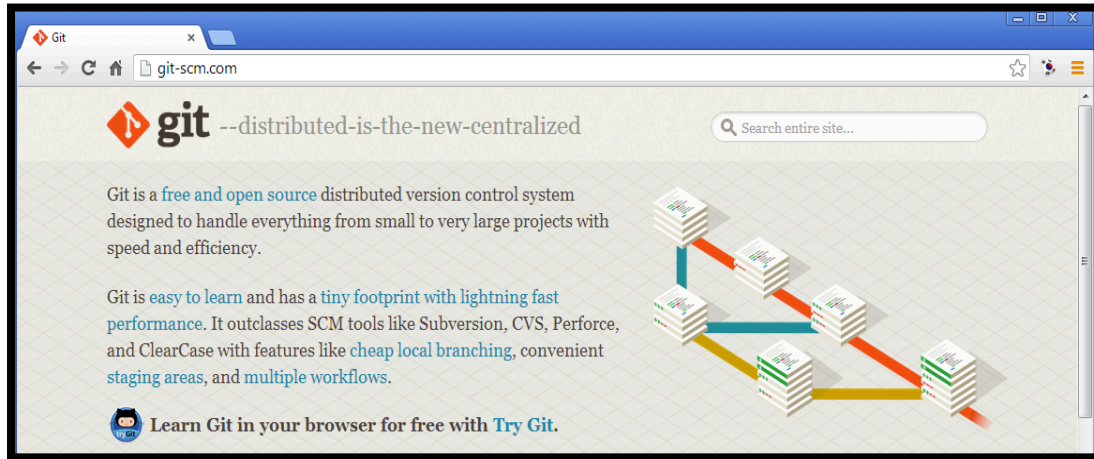


Imagen 23 Sitio oficial con las ultimas versiones del gestor de versiones GIT.



Imagen 24 Asistente de instalación para el gestor de versiones GIT.

Como requerimiento para iniciar el sistema en plataforma *Windows* es

necesario instalar las dependencias de Microsoft Visual C++ 2008

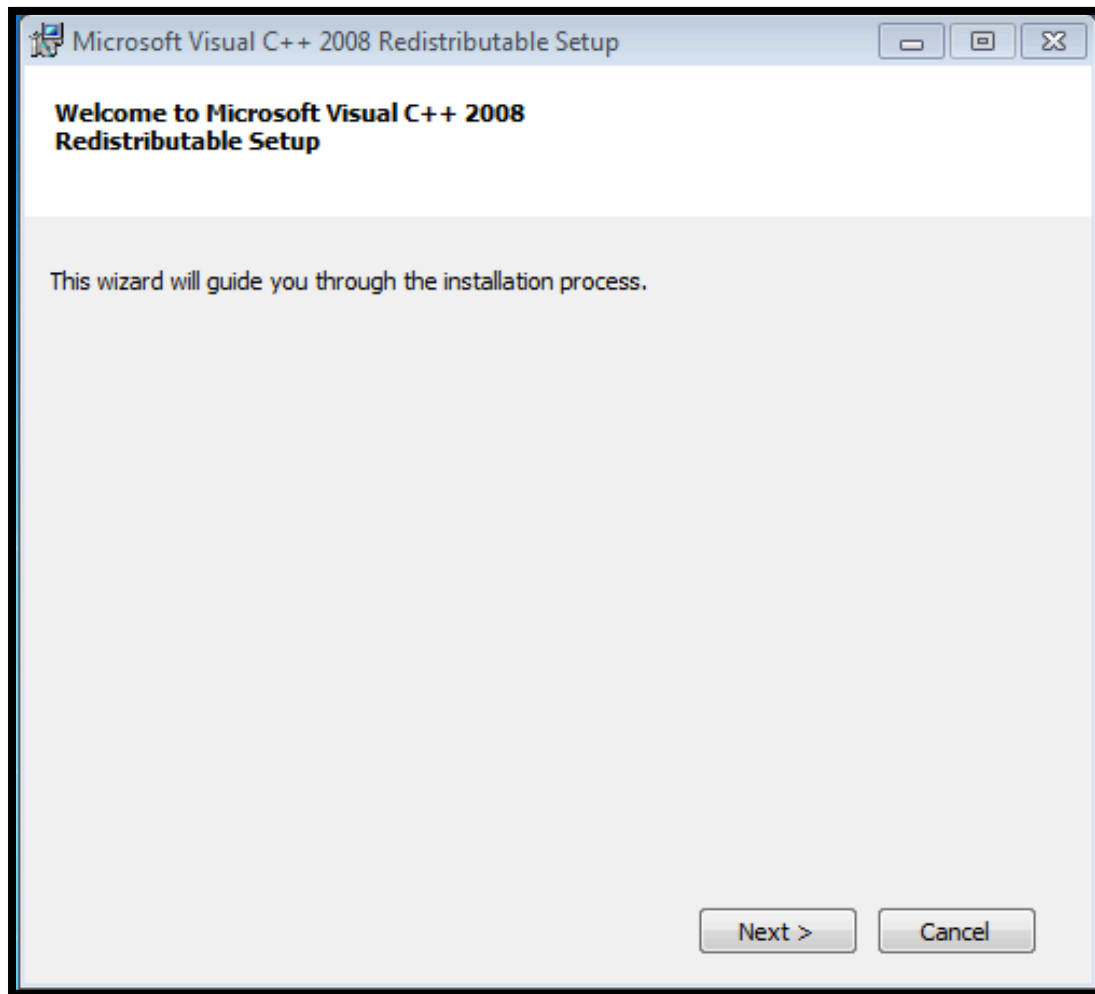


Imagen 25 Asistente de instalación de las dependencias de Microsoft Visual C++ 2008.

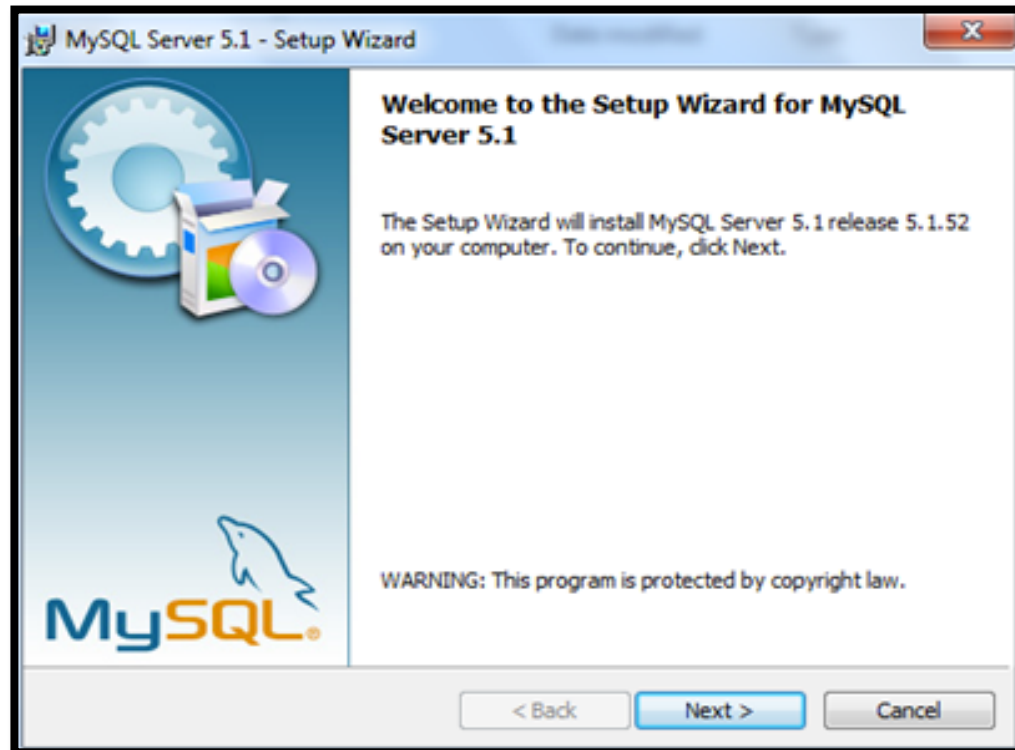


Imagen 26 Asistente de instalación para el sistema gestor de base de datos en Windows.

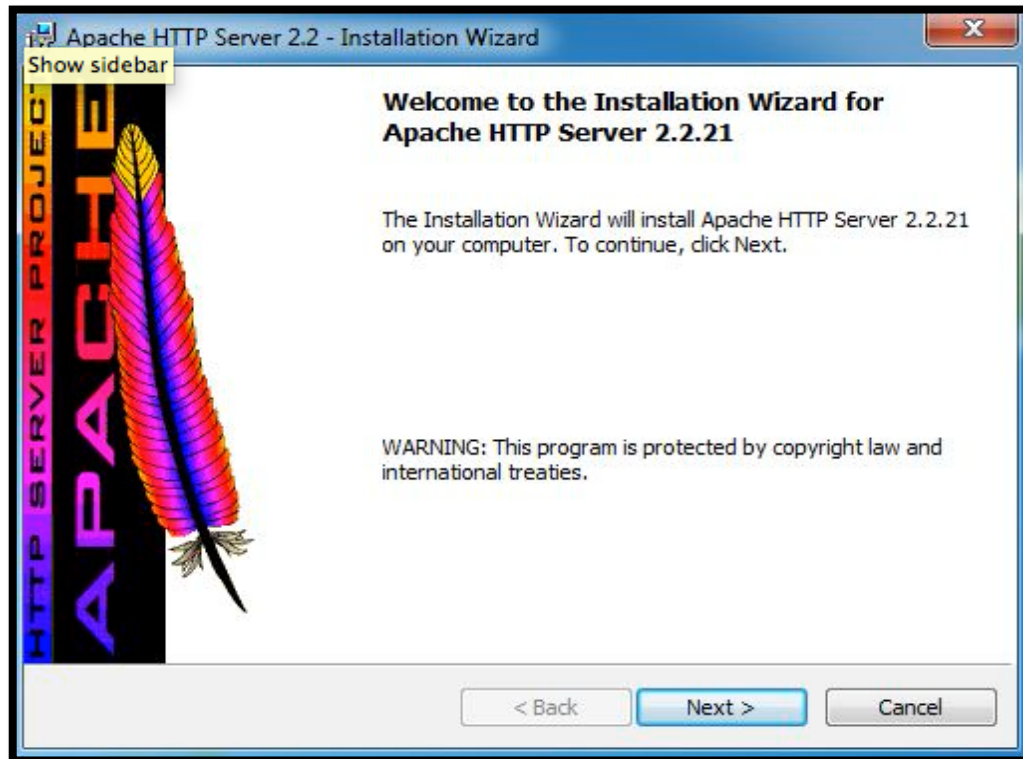


Imagen 27 Asistente de instalación para el servidor apache2.

Es necesario agregar el modulo *passenger_module* a la configuración del servidor apache, así como la configuración del host virtual, para habilitar el funcionamiento del sistema en *Rails 3* utilizado en el presente proyecto.

```
File Edit Format View Help
#BrowserMatch "MSIE 10.0;" bad_DNT
#</IfModule>
#<IfModule headers_module>
#RequestHeader unset DNT env=bad_DNT
#</IfModule>

# XAMPP: we disable operating system specific optimizations for a listening
# socket by the http protocol here. IE 64 bit make problems without this.

AcceptFilter http none

# AJP13 Proxy
<IfModule mod_proxy.c>
<IfModule mod_proxy_ajp.c>
Include "conf/extra/httpd-ajp.conf"
</IfModule>
</IfModule>

#Modulo para el sistema en ruby on rails
LoadModule passenger_module /somewhere/passenger/ext/apache2/mod_passenger.so

PassengerRoot /somewhere/passenger/
PassengerDefaultRuby /usr/bin/ruby
PassengerMaxPoolSize 10

<VirtualHost *:80>
  ServerName 0.0.0.0:3000
  DocumentRoot /webapps/foo/public
  RailsBaseURI /rails
</VirtualHost>
```

Imagen 28 Agregado del modulo "passenger" u "virtual host" al archivo de configuración del servidor apache.