

**DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SOFTWARE DE SISTEMA
DE ALERTAS TEMPRANAS PARA LA PROBLEMÁTICA DE LOS ARROYOS
EN LA CIUDAD DE BARRANQUILLA**

**Luis Javier Rodríguez Zarate
Diego Armando Hernández Villalba**

**UNIVERSIDAD DE LA COSTA CUC
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BARRANQUILLA
2014**

**DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SOFTWARE DE SISTEMA
DE ALERTAS TEMPRANAS PARA LA PROBLEMÁTICA DE LOS ARROYOS
EN LA CIUDAD DE BARRANQUILLA**

**Luis Javier Rodríguez Zarate
Diego Armando Hernández Villalba**

**Trabajo de Grado presentado como requisito para optar por el título de:
Ingeniero De Sistemas**

Asesor: Ingeniero de sistemas Jorge Sepúlveda

**UNIVERSIDAD DE LA COSTA CUC
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BARRANQUILLA**

2014



Barranquilla 15 de Agosto de 2014

Señores: Corporación Universidad de la Costa (CUC)
Atención: Departamento Admisiones y Registros y Biblioteca
Asunto: Recepción de proyecto de grado

El día 15 de Agosto de 2014, el estudiante **Diego Armando Hernandez Villalba** identificado con **C.C. 1140837122**, **Luis Javier Rodriguez Zarate** identificado con **C.C. 1048212092**, hicieron la entrega de los documentos y monografía de proyectos de grado junto con los anexos respectivos al programa de ingeniería de sistemas. A continuación se relaciona la información del proyecto de grado:

Título del proyecto: Diseño y Desarrollo de un Prototipo de Software de Sistema de Alertas Tempranas para la Problemática de los Arroyos en la Ciudad de Barranquilla

Calificación: 3.9

Decisión del jurado evaluador: Aprobado

Tutor: Jorge Sepulveda Ojeda

Cotutor: Alexis de la Hoz Manotas

El jurado calificó el proyecto presentado como un proyecto destacable dado los objetivos alcanzados con el desarrollo del trabajo de grado. Esta calificación incluye el desempeño durante todo el proyecto y la sustentación final asignada por los jurados que por su director de proyecto de grado lo que incluye:

Atentamente

Paola Ariza Colpas
Ing. Paola Patricia Ariza Colpas
Director de Programa
Ingeniería de Sistemas

NOTA DE ACEPTACION

JURADO 1

JURADO 2

DEDICATORIA

A mis padres porque han sido en todo mi proceso de formación profesional el pilar para sobrellevar cualquier dificultad presentada y que de manera significativa promovieron la persona quien soy.

Diego Hernández Villalba

A mis padres por su amor y apoyo incondicional en todo momento
En mi formación y a todas las personas que compartieron conmigo
Y aportaron de alguna u otra forma parte en mi formación
Integral como persona y como profesional.

Luis Rodríguez Zarate

AGRADECIMIENTOS

Agradecemos este trabajo a Dios, por haber dado la vida a profesores, amigos y familiares que estuvieron en todo nuestro camino de formación como profesionales, de ser guías y personas de confianza brindando sus conocimientos y experiencias en el campo y otorgando un mayor acoplamiento en los intereses conceptuales de la carrera de la vida y de la profesión.

Al Ingeniero Jorge Sepúlveda en su condición de asesor, quien nos brindó todo el apoyo y nos motivó a investigar y a enfocarnos en la elaboración total de nuestro proyecto.

RESUMEN

El desarrollo de la investigación presenta una aplicación web, en el cual se alerta al ciudadano local sobre los arroyos y su nivel de peligrosidad en la ciudad de Barranquilla.

Para el desarrollo del proyecto se utilizó el lenguaje de programación Ruby con el framework Rails el cual usa como patrón de diseño la arquitectura MVC, la metodología utilizada para el proceso del desarrollo del ciclo de vida del software fue la metodología de Rumbaugh por medio de ella se realizó el proceso de análisis del objeto en donde se plasmó todo el proceso de levantamiento de información y captura de requisitos, el proceso de diseño el cual una vez realizado el levantamiento de información, se realizaron los diagramas de casos de uso y los diagramas de clases para determinar, delimitar y detallar el esquema final del prototipo y por último el proceso de implementación el cual se aplicaron los diagramas realizados y los requisitos funcionales y no funcionales; finalmente se realizan las pruebas de la herramienta para la comprobación funcional de todo el sistema, dando como resultado un prototipo final que permite a los usuarios estar informado de los niveles de peligrosidad que cuentan los arroyos de la ciudad en temporadas invernales.

Palabras claves: Arroyos, nivel de peligrosidad, mapas, sensores, casos de uso, MVC, Ingeniería del software, aplicación web, sistema de alertas tempranas.

ABSTRACT

The development of the research presents a web application, which alerts the local citizen on streams and their level of hazard in the city of Barranquilla.

For the project the programming language Ruby with the Rails framework which uses as the MVC design pattern architecture was used, the methodology used for the development process of the software life cycle methodology Rumbaugh was through her the scan of the object in which the whole process of information gathering and capture requirements, the design process which once made the information gathering was held was reflected, diagrams and use case diagrams were performed classes to determine, define and detail the final outline of the prototype and finally the implementation process which diagrams made and implemented functional and non-functional requirements; finally testing tool for functional testing of the complete system are made, resulting in a final prototype that allows users to be informed of the danger levels that have streams of the city in winter seasons.

Keywords: streams, danger level, maps, sensors, cases of use, MVC, Software Engineering, Web App, early warning system

CONTENIDO

1. DESCRIPCIÓN DE LA COMPOSICIÓN DEL PROYECTO	14
1.2 PARTICIPANTES DEL PROYECTO.....	15
1.1 DEFINICIÓN DEL PROBLEMA.....	16
1.1.1 INTRODUCCIÓN	16
1.1.2 PLANTEAMIENTO DEL PROBLEMA.....	16
1.1.2.1 Área Temática	16
1.1.2.2 FORMULACIÓN DEL PROBLEMA	16
1.1.2.3 IDENTIFICACIÓN Y DELIMITACIÓN DEL PROBLEMA.....	19
1.1.2.3.1 Delimitación Temporal	19
1.1.2.3.2 Delimitación Espacial.....	19
1.1.3 FINES DEL PROYECTO.....	19
1.1.3.1 Justificación	19
1.1.3.2 Beneficios tangibles.....	19
1.1.3.3 Beneficios Intangibles	20
1.1.4 OBJETIVOS	21
1.1.4.1 OBJETIVOS GENERALES.....	21
1.1.4.2 OBJETIVOS ESPECIFICOS.....	21
1.1.5 METODOLOGIA USADA	22
1.1.5.1 Fases del proyecto	22
1.1.5.1.2 Especificación.....	25
1.1.6 ESTADO DEL ARTE	26
1.1.7 PROPUESTA DE SOLUCIÓN.....	27
2. ANÁLISIS	28
2.1 INTRODUCCIÓN	28
2.2 ALCANCE Y PROPÓSITO.....	28
2.3 ÁMBITO	29
2.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	29
2.5 DESCRIPCIÓN GENERAL	30

2.5.1	Funciones del Producto	30
2.6	RESTRICCIONES GENERALES	31
2.6.1	Políticas Regulatoras	31
2.7	LIMITACIONES DE HARDWARE	31
2.7.1	Funciones de Control.....	31
2.7.2	Requisitos del Lenguaje.....	31
2.7.3	Credibilidad de la Aplicación	31
2.7.4	Consideraciones de Seguridad	31
2.8	Especificar requerimientos	32
2.8.1	Requerimientos Funcionales.....	32
	Ingreso a la Aplicación	32
2.8.2	Obligaciones del Diseño de interfaces	34
2.8.2.1	Estándares Cumplidos	34
2.8.3	Requisitos no Funcionales	34
3.	DISEÑO.....	35
3.1	INTRODUCCIÓN.....	35
3.2	DIAGRAMA DE CLASES	35
3.2.1	Diccionario de Entidades	36
3.3	CASOS DE USO	37
3.3.1	Resumen Actores	37
3.3.1.1	Actores	38
3.3.2	Casos de uso Gestionar Sensor	38
3.3.3	Casos de uso Gestionar Punto	39
3.3.4	Casos de uso Visualizar Arroyo	40
3.4	Diccionario de Entidades base de datos	41
4.	ARQUITECTURA	43
4.1	INTRODUCCIÓN	43
4.2	ARQUITECTURA DEL SISTEMA.....	43
4.3	CAPAS DE APLICACION.....	44
4.3.1	Descripción de la arquitectura.....	44

4.4	DIAGRAMA DE INTERFAZ.....	44
4.4.1	Interfaz Principal	44
4.4.2	Interfaz para visualizar uno o todos los arroyos registrados en el sistema.....	45
4.4.3	Interfaz para crear un sensor.....	46
4.4.4	Interfaz para visualizar los sensores que se encuentran registrados en el sistema.....	46
4.4.5	Interfaz para crear un punto.....	47
4.4.6	Interfaz para visualizar los puntos registrados en el sistema.....	47
5.	IMPLEMENTACION	48
5.1	INTRODUCCIÓN	48
5.2	CONCEPTOS UTILIZADOS.....	48
	MODELO	49
	VISTA	49
	CONTROLADOR	49
6.	PRUEBAS	50
6.1	INTRODUCCIÓN	50
6.2	VALIDACIONES.....	50
6.2.1	Validación de base de datos	50
6.2.2	Validación de navegadores.....	50
6.3	PRUEBAS	53
6.3.1	Gestión de sensores	53
6.3.2	Gestión de puntos.....	53
6.4	CONCURRENCIA DE LA INFORMACIÓN.....	54
7.	CONCLUSIONES.....	55
	BIBLIOGRAFÍA.....	56

LISTADE TABLAS

Tabla N. 1 Resumen de actividades	23
Tabla N. 2 Actividades de administración	23
Tabla N. 3 Cronograma de actividades	24
Tabla N. 4 Especificación de requerimientos	25
Tabla N. 5 Requerimiento ingreso a la aplicación	32
Tabla N. 6 Requerimiento visualizar mapa localizador de arroyo	32
Tabla N. 7 Requerimiento crear sensores	33
Tabla N. 8 Requerimiento crear puntos	33
Tabla N. 9 Diccionario de entidades clase arroyo	36
Tabla N. 10 Diccionario de entidades clase sensor	36
Tabla N. 11 Diccionario de entidades clase punto	36
Tabla N. 12 Identificador gestionar sensor	39
Tabla N. 13 Identificador gestionar punto	40
Tabla N. 14 Identificador visualizarArroyo	41
Tabla N. 15 Diccionario de entidades tabla arroyo	53
Tabla N. 16 Diccionario de entidades tabla sensor	53
Tabla N. 17 Diccionario de entidades tabla punto	53
Tabla N. 18 Diccionario de entidades tabla usuario	53
Tabla N. 19 Prueba gestión de sensores	53
Tabla N. 20 Prueba gestión de puntos	53

LISTA DE GRÁFICOS

Gráfico N° 1 metodología de Rumbaugh	22
Gráfico N° 2 Cronograma de actividades	24
Gráfico N° 3 Diagrama de clases	35
Gráfico N° 4 Caso de uso gestionar sensor	38
Gráfico N° 5 Caso de uso gestionar punto	39
Gráfico N° 6 Caso de uso visualizar arroyo	40
Gráfico N° 7 Arquitectura del sistema	43
Gráfico N° 8 Interfaz principal	44
Gráfico N° 9 Interfaz visualizar arroyos	45
Gráfico N° 10 Interfaz crear sensor	46
Gráfico N° 11 Interfaz visualizar sensor	46
Gráfico N° 12 Interfaz para crear punto	47
Gráfico N° 13 Interfaz para visualizar puntos	47
Gráfico N° 14 Arquitectura MVC	48
Gráfico N° 15 Validación de navegador internet Explorer	50
Gráfico N° 16 Validación de navegador google Chrome	51
Gráfico N° 17 Validación de navegador Safari	52

INTRODUCCIÓN

Los arroyos representan una de las problemáticas más graves que enfrenta Barranquilla, la ciudad más importante del caribe colombiano ubicada en el margen izquierdo del río Magdalena, el más importante, extenso y caudaloso del país, son corrientes de aguas que se forman durante y después de las lluvias, y alcanzan velocidades superiores a los 90 nudos que equivalen a unos 160 km por hora, la velocidad del agua sumada a su fuerza convierte los arroyos en ríos incontrolables que arrastran todo lo que encuentren en el camino, ocasionando de esta manera accidentes fatales, contaminación ambiental, parálisis del tránsito vehicular, entre otras. Un estudio de factibilidad y diseño de soluciones alternativas al drenaje pluvial señala que hay 6 factores que originan la gravedad del problema de los arroyos en la ciudad, los cuales son: crecimiento no planificado, cultura e idiosincrasia, condiciones pluviométricas, condiciones topográficas, condiciones geomorfológicas y la falta de inversión en proyectos que den solución en eliminar o mitigar el peligro que generan los arroyos.

Buscar una solución que permita impulsar una cultura ciudadana en mantener informado en temporadas invernales de los niveles de peligrosidad que tienen los arroyos en la ciudad, esto con el fin de evitar accidentes por las fuertes corrientes pluviales que recorren las principales calles, con base en esta problemática se crea una herramienta web que permita informar a los ciudadanos del nivel de peligrosidad que tienen los arroyos antes, durante y después de las lluvias, con lo anterior se realiza un análisis extenso de los diferentes sistemas de alertas tempranas y se diseña un prototipo enfocado a resultados en brindar información de los arroyos y posterior a esto su desarrollo e implementación de la herramienta, dando como resultado una plataforma web que todos los ciudadanos puedan ingresar sin ningún costo ni tipo de acceso y poder estar informado las 24 horas en tiempo real del estado de peligro en se encuentran los arroyos

1. DESCRIPCIÓN DE LA COMPOSICIÓN DEL PROYECTO

Se describe la composición del proyecto y plantea la problemática de los arroyos delimitando el proyecto a nivel temporal y espacial, se justifica la necesidad de una herramienta para la solución al problema planteado y por último se establecen los objetivos y la propuesta de solución.

Se realiza el análisis con la descripción de la propuesta de solución plasmada en el capítulo anterior en donde se determina las funciones finales del producto, restricciones generales, credibilidad de la aplicación, seguridad y por último los requerimientos funcionales y no funcionales de la aplicación.

Se realiza el diseño de clases y de casos de usos con base en la captura de requisitos descrita en el capítulo anterior

Se presenta la arquitectura del proyecto y la interacción de los medio para su completo funcionamiento.

Se presenta como se desarrolló el proyecto y bajo qué paradigmas de programación se implementó el código.

Se describen las pruebas que se requirieron para la aprobación del producto final.

Finalmente se realizan las respectivas conclusiones y fases futuras del proyecto

1.2 PARTICIPANTES DEL PROYECTO

Para llevar adelante el presente trabajo se cuenta con los siguientes participantes:

- Ingenieros de Sistemas:

Luis Rodríguez Zarate y Diego Hernández Villalba, autores del presente Trabajo.

- Usuarios que se integrarán al proyecto, en calidad de usuarios especialistas, los cuales son las siguientes:

Jorge Sepúlveda – Ingeniero de sistemas

1. DEFINICIÓN DEL PROBLEMA

1.1 INTRODUCCIÓN

Este capítulo se centra en plasmar específica y generalmente el problema en el cual se realizan una serie de estudios y análisis para dar soluciones al problema expuesto en las siguientes secciones. Dando la solución más adecuada y práctica para que cualquier persona tenga la posibilidad de obtener información pertinente y en tiempo real de estados de los arroyos en la ciudad.

Seguido se presentan los objetivos y la metodología a utilizar con base en un exhausto análisis.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 Área Temática

La ingeniería web es la base para cualquier proceso de desarrollo del ciclo de vida del software a nivel web, con base en estos conocimientos se plasma el proceso y metodologías utilizadas para la generación y proceso de desarrollo del requerimiento planteado.

1.2.2 FORMULACIÓN DEL PROBLEMA

Los arroyos de Barranquilla han existido desde siempre y a través de la historia, se han vinculado con el desarrollo urbanístico de la ciudad. De acuerdo con la tradición recogida por (Arroyos de Barranquilla, 2013) Domingo Malabeth, se da por sentado que para la época del nacimiento de la ciudad las barrancas en que se estableció estaban separadas por arroyos poderosos que bajaban las aguas de la sierra del noroeste y las de los altos areniscos del viejo camino de Soledad, hacia una gran ciénaga espaciosa, comunicada con el río.

La ciénaga, que tenía su borde occidental en lo que hoy es la plaza del paseo de Bolívar y con anchura probable desde la carrera del Cuartel, hasta más allá de la plaza de la Tenería, recibía tributo del extinguido caño de Soledad y era una especie de lago que tenía gran caudal de aguas profundas, había sido primero parte del mar y antes de la formación del terreno conocido hoy como La Loma, era parte del Río Magdalena. Cuando La Loma, completó su formación, quedó siendo un brazo del río que recibía sus aguas en cercanías de Ponedera para arrojarlo otra vez al río por el caño de la tablaza.

Las tierras altas del occidente, y parte de las del suroeste vertían sus aguas a la ciénaga, a través de un gran arroyo que bajaba por el callejón de Progreso, cruzando por la Calle Ancha (llamada así por el arroyo que se formaba en su centro) hasta la laguna. La existencia de la ciénaga, explica la forma irregular de los callejones formados en la banda oriental de la Calle del Comercio y la banda occidental de la Calle Real.(Arroyos de Barranquilla, 2013)

Los arroyos también han tenido que ver con la movilidad de sus habitantes, la comunicación con las poblaciones vecinas y especialmente, con la creación de algunos caños. En un escrito publicado en 1872(Arroyos de Barranquilla, 2013) se afirma:

El año 1687, cuando aún no se había formado la faja de terreno conocida hoy como Barranquillita, la ciénaga fue parte de la ribera occidental del Río Magdalena... Cuando los depósitos de aluvión al solidificarse formaron La Loma, no quedaron caños de comunicación con el río en frente de la ciudad; apenas había algunas vertientes o desagües naturales, formados por la presión de las aguas en las épocas de creciente, cuyos nombres no ha conservado la tradición. Por consiguiente, la acción natural de las aguas, no interrumpida o trastornada por ninguna corriente transversal, tendía siempre a profundizar su cauce, ganando fuerzas en cada invierno, para volver a arrojar al río los objetos flotantes que éste introducía desde las cercanías de Ponedera. Las necesidades del tráfico hicieron sin duda que los pueblos de la ribera occidental, aprovechando esas mismas vertientes, establecieran una comunicación regular por el cañón de Arriba, el de los Tramposos, el de Trupillos y el que conducía a otros pueblos de la provincia. La desviación del río en el curso de muchos años fue impulsada en épocas de sequía por las enormes cantidades de arena, de escombros y de basuras que ha venido arrastrando las aguas pluviales en su atropellado descenso por el plano inclinado de la población.

Si bien en la época colonial la comunicación entre las diferentes poblaciones no era problema, por el buen estado en que ordinariamente se encontraban los caminos, y por la poca distancia, que separaba cada población, las inundaciones y la formación de grandes arroyos ya impedía la movilidad de las personas (Arroyos de Barranquilla, 2013)

Actualmente la ciudad de Barranquilla afronta una serie de problemas generados a partir de los grandes volúmenes y caudales de agua que circulan durante la temporada invernal por vías convertidas en canales y por todas las corrientes

naturales (arroyos). Esta problemática fue advertida a través de un estudio realizado por HERBARD & COMPANY INC, quienes plantearon en el año de 1920 la necesidad del programa de alcantarillado pluvial, sin embargo la situación financiera de Barranquilla impidió ejecutar el plan propuesto. Posteriormente, en 1956 la TOWN PLANNING COLABORATIVE (Alcaldía Distrital, 2013) estableció el primer plan regulador y recomienda la canalización de algunos arroyos, y por las mismas razones anteriores, nada pudo realizarse.

Algunas desventajas vistas para la ciudad dado por el origen de fuertes corrientes pluviales por las grandes lluvias presentadas en temporadas invernales son:

1. Situaciones de emergencia por levantamiento de placas de concreto.
2. Creación de grandes corrientes pluviales originada por las fuertes lluvias que atentan contra el tránsito vehicular y peatonal.
3. Generación de enfermedades mortales debido a las cantidades de aguas albergadas y se contaminan con el tiempo.

De esta necesidad y debido al gran impacto que ha tenido la tecnología sobre la sociedad en los últimos años, surge una pregunta general:

¿Cómo pueden ayudar las TIC, en la generación de una solución en la ayuda al tránsito vehicular y peatones que puedan estar informados en tiempo real del estado de los arroyos en temporada invernal?

Y de estas se deriva las siguientes preguntas:

¿Qué herramienta puede ser la más viable para la creación de un prototipo que genere solución a estas necesidades?

¿Qué tecnología se puede considerar para el desarrollo e implementación de un producto altamente confiable y eficiente?

1.2.3 IDENTIFICACIÓN Y DELIMITACIÓN DEL PROBLEMA

1.2.3.1 Delimitación Temporal

El proceso de desarrollo y ejecución de las etapas de análisis, diseño, desarrollo e implementación del proyecto se elaboraron en el periodo de septiembre/2013a Marzo/2014.

1.2.3.2 Delimitación Espacial

El Proyecto de investigación se realizó en la ciudad de barranquilla bajo la directiva de la universidad de la Costa CUC y como ente aprobatorio COLCIENCIAS, como proyecto de desarrollo y ejecución en innovación, dando una solución aplicada a nivel social en la ciudad de barranquilla.

1.3 FINES DEL PROYECTO

1.3.1 Justificación

Referente a la situación actual que los ciudadanos de la ciudad a raíz de su problemática de los arroyos y la gran necesidad que se presenta en tener el conocimiento de estos eventos que producen los arroyos, se analiza mediante un proyecto propuesto la tecnología más eficiente para una óptima solución.

A raíz de la problemática de los arroyos y de las afectaciones que ha hecho a lo largo de los años, y que hasta el momento no han dado una solución que de fin a este problema, buscar soluciones que impulsen a nivel social el manejo de información pertinente de los niveles de peligrosidad de los arroyos para así poder estar informados y prevenir una eventual tragedia por estas fuertes corrientes.

Por lo anterior, establecer una aplicación web en donde el usuario puede interactuar de manera completamente visual y estéticamente agradable toda la información pertinente para estar prevenido e informado de los niveles de peligro que tienen los arroyos existentes en los diferentes puntos, aminorando de esta manera el porcentaje de desastres que se presentan actualmente en la ciudad de Barranquilla.

1.3.2 Beneficios tangibles

El beneficio más importante es tener la información en tiempo real de los arroyos más peligrosos y sus rutas alternas para evitar cualquier accidente que genera la presencia de los arroyos, de esta manera generar una conciencia ciudadana en la prevención y cuidado que se debe tener con estas corrientes pluviales.

1.3.3 Beneficios Intangibles

Promover una conciencia y cultura ciudadana en la prevención de accidentes que históricamente han sucedido en la ciudad por imprudencias o simplemente por no estar informado del peligro que puede tener y hacer un arroyo, como así también promover el uso de las TIC para el desarrollo de soluciones que generen un impacto positivo a nivel social.

1.4 OBJETIVOS

1.4.1 OBJETIVOS GENERALES

- Desarrollar un prototipo de software de un sistema de alerta tempranas para identificar el nivel de peligrosidad de los arroyos de la ciudad de Barranquilla a través de una aplicación web

1.4.2 OBJETIVOS ESPECIFICOS

- Realizar el proceso de Identificación de los datos correspondientes a las rutas de los arroyos.
- Identificar en la ruta del arroyo un punto que hace referencia a un sensor para informar el nivel de peligrosidad.
- Generar una herramienta web que muestre información en tiempo real de la ruta y del estado de peligrosidad en el cual se encuentre el arroyo.
- Mostrar al usuario las rutas alternas para evitar secciones donde el arroyo tenga el nivel de peligrosidad alta.
- Realizar pruebas que permitan validar el prototipo de software.

1.5 METODOLOGIA USADA

La metodología de Rumbaugh (Rumbaugh, 1996) es la usada para el proyecto ya que es una de las principales metodologías orientadas a objeto, ajustándose al proceso de análisis, diseño y desarrollo del software. Por lo anterior se realiza la integración del proceso de desarrollo del ciclo de vida del software de la mejor manera y permitiendo tener un análisis del requerimiento a groso modo.

1.5.1 Fases del proyecto

De la metodología de Rumbaugh (Rumbaugh, 1996) fueron tomadas las 4 fases para el análisis y desarrollo del requerimiento: análisis de objetos, diseño del sistema, diseño de objetos e implementación. A continuación como se muestra en el gráfico N°1 se presenta los ciclos del proceso para la integración e interacción del producto final:

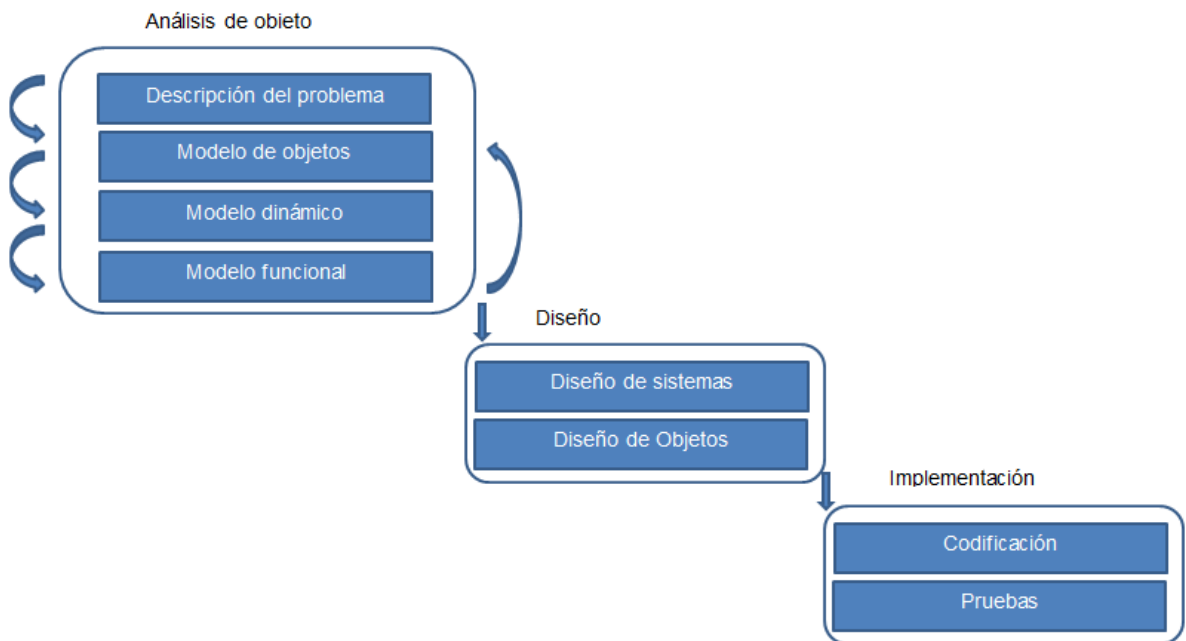


Gráfico N° 1 Metodología de Rumbaugh

Consecuente de lo anterior, se describe brevemente en la siguiente *tabla N°1* el resumen de las actividades de las fases del proyecto expuestas en el gráfico anterior.

Fase	Descripción
Análisis de Objetos	Una vez obtenida toda la información necesaria y pertinente para aterrizar la idea del proyecto, se realizó la identificación de los requerimientos en el levantamiento de información con los usuarios.
Diseño del Sistema	Con la información capturada y los análisis de casos de uso, se diseña una arquitectura con todas las interacciones que tendrá la aplicación.
Diseño de Objetos	Con toda la información detallada se realiza el diseño de clases para la identificación de los entes principales.
Implementación	Se plasma en código toda la información detallada en el diagrama de casos de uso y diagrama de clases

Tabla N. 1 Resumen de actividades

Actividades de Administración del proyecto	
Actividades	Objetivos
Planificar	<ul style="list-style-type: none"> • Levantamiento de información del requerimiento • Identificación y creación de cada uno de los objetos considerados en el levantamiento de información. • Creación de la base de conocimientos de la aplicación. • Creación de la estructura de la herramienta • Determinación de tecnologías a utilizar
Proceso de calidad	<ul style="list-style-type: none"> • Se realizaron revisiones minuciosas del levantamiento de información. • Se realizan casos de pruebas.
Controlar	<ul style="list-style-type: none"> • Se hizo el control mediante el uso de cronogramas de actividades (<i>ver tabla N°3</i>)

Tabla N. 2 Actividades de administración

El siguiente *Tabla N°3* presenta el diagrama de las actividades realizadas para llevar a cabo todo el proceso desarrollo del requerimiento:

Cronograma de actividades 2014			
ActividadNo.	Concepto	Inicia	Finaliza
1	Levantamiento de información	15-sep-13	15-oct-13
2	Captura de requisitos	15-oct-13	20-ene-14
3	Creación casos de uso	02-feb-14	10-feb-14
4	Diseño de clases	10-feb-14	20-feb-14
5	Diseño de base de datos	20-feb-14	05-mar-14
6	Documentación	05-mar-14	20-abr-14
7	Creación de interfaces	20-abr-14	30-abr-14
8	Creación de lógica de negocio	05-may-14	20-jun-14
10	Pruebas	20-jun-14	05-jul-14

Tabla N. 3 Cronograma de actividades

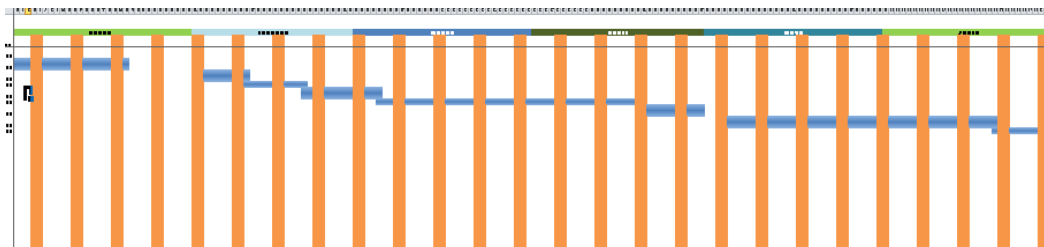


Gráfico N° 2 Cronograma de actividades

1.5.1.2 Especificación

Para la especificación del requerimiento se da formato a la siguiente información:

Documento de especificación de requerimientos	
Nombre	Sistema de Alertas Tempranas SAT.
Fecha Inicio	03/22/2013
Profesional(es) funcional	Diego Hernández Villalba, Luis Rodríguez Zarate.
Profesional de desarrollo	Luis Rodríguez Zarate, Diego Hernández Villalba.
Propósito →	Construir una aplicación web que permita a usuarios obtener información completa y en tiempo real de los arroyos de la ciudad.
Nivel de formalidad	Formal.
A quien va dirigido	Audidores, consultores y usuarios.
Fuentes de conocimiento	Expertos en auditoria de sistemas de información de la Universidad de la costa, CUC y usuarios.

Tabla N. 4 Especificación de requerimientos

1.6 ESTADO DEL ARTE

SATCA - Sistema de Alerta Temprana para Centroamérica, tiene como propósito fundamental fortalecer la capacidad de anticipar posibles amenazas naturales en Centroamérica para mejorar la preparación, mitigación y respuesta humanitaria. Con este propósito, SATCA fortalece los sistemas de alerta temprana en la región centroamericana a través de una plataforma Web regional dedicado al monitoreo de posibles amenazas naturales(satcaweb, 2012).

SIATA –Sistema de Alerta Temprana- es un proyecto del Área Metropolitana del Valle de Aburrá; La Alcaldía de Medellín, en cabeza del Departamento Administrativo de Gestión del Riesgo DAGRED; EPM e ISAGEN. El objetivo principal del proyecto es alertar de manera oportuna a la comunidad sobre la probabilidad de ocurrencia de un evento hidro-meteorológico extremo que pueda generar una situación de emergencia y así reducir los impactos de los fenómenos mediante la implementación de medidas de respuesta ante una amenaza inminente (siata).

SIAT - El Sistema de Alerta Temprana consta de un conjunto de procedimientos e instrumentos automatizados que permiten, por una parte, detectar oportunamente a los alumnos de educación media superior que están en riesgo de abandonar los estudios; y por otra, poner en marcha, con la debida oportunidad, las intervenciones adecuadas para lograr su permanencia en la escuela(sems, 2013).

FamineEarlyWarningSystems Red: Es un sistema de información diseñado para identificar problemas en el sistema de suministro de alimentos que potencialmente lleva a la hambruna o de otras condiciones que padecen inseguridad alimentaria en el África subsahariana, Afganistán, América Central y Haití (fews, 1985).

SLDS: Es una herramienta basada en datos que utiliza indicadores para ayudar a predecir el estado de los estudiantes que están en el camino correcto hacia una eventual graduación u otros objetivos de. A través de sistemas de alerta temprana, las partes interesadas a nivel de la escuela, del distrito y del estado pueden ver los datos de una amplia gama de perspectivas y obtener una comprensión más profunda de los datos de los estudiantes (nces, 2012).

EN CHILE: Las alertas de emergencia recibidas en los terminales móviles compatibles, serán en formato similar a un mensaje de texto SMS, focalizadas por área geográfica, según la ubicación y tipo de emergencia que determine la ONEMI (culturadigital, 2011).

La plataforma **Terra MA2**, instalada en Madre de Dios (Perú), permitirá realizar alertas a tiempo real de inundaciones e incendios forestales y otros tipos de desastres naturales y estará integrada con Brasil y Bolivia, permitiendo de esta forma fortalecer los datos con los que se cuentan para realizar la alerta de eventos extremos (sems, 2013)

1.7 PROPUESTA DE SOLUCIÓN

Por medio de un grupo de investigación, se desea implementar el desarrollo de una aplicación web, capaz de alertar al ciudadano en tiempo real de los posibles arroyos más cercanos a su punto de localización geográfica, esta información estará acompañada del nivel de peligrosidad del arroyo, dirección y puntos geográficos exactos, entre otros.

2. ANÁLISIS

2.1 INTRODUCCIÓN

En la presente sección busca definir las especificaciones funcionales y no funcionales del sistema, así mismo el alcance que puede tener este. La aplicación tendrá 2 perfiles de usuarios, un perfil el cual es el encargado de realizar la creación, eliminación y edición de los arroyos, puntos que es la trazabilidad del arroyo y sensores que son los que brindan información del nivel de peligrosidad del arroyo en un lugar específico, el otro perfil es un usuario que desea visualizar o eliminar la información sin ingresar a la plataforma.

2.2 ALCANCE Y PROPÓSITO

Desarrollo e implementación de una aplicación web en la cual un usuario pueda obtener información de los estados de los arroyos más peligrosos de la ciudad y la plataforma debe tener la capacidad de brindarle las rutas alternas en dos puntos elegidos por el usuario, además debe existir un usuario administrador el cual es encargado de la creación de los arroyos en la plataforma. Para esta fase se simularan los sensores creándolos en la aplicación ya que no se encuentran instalados en los diferentes puntos estratégicos por donde pasan los arroyos

La aplicación fue desarrollada en el lenguaje de programación RUBY con el framework Rails con la versión disponible a la fecha 3.2.11 para el desarrollo del proyecto. La Aplicación constará con los siguientes procesos:

1. Visualizar todos los arroyos.
2. Visualizar un arroyo determinado.
3. Crear sensores.

Mediante este sistema No se podrá:

1. Modificar la descripción del estado de un punto.
2. Modificar el color de presentación del arroyo.
3. Modificar tamaño en ancho y alto del marco del mapa.
4. Modificar colores del puntero en el mapa.

2.3 ÁMBITO

El desarrollo del requerimiento está enfocado a brindar información pertinente y relevante para el usuario de los estados de los arroyos, para esta fase se contara con la simulación de la creación de los sensores en la misma aplicación para efectos de muestreo y pruebas de funcionalidad del requerimiento.

2.4 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

A continuación se enumera la jerga o vocabulario común en el desarrollo del proyecto:

Interfaz: Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario (Kimmel, Manual de UML, 2008)

GUI: Interfaz gráfica de usuarios, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz (Kimmel, Manual de UML, 2008).

Bitbucket: Es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de revisiones Mercurial y Git. Bitbucket ofrece planes comerciales y gratuitos. Se ofrece cuentas gratuitas con un número ilimitado de repositorios privados (que puede tener hasta cinco usuarios en el caso de cuentas gratuitas) desde septiembre de 2010, los repositorios privados no se muestran en las páginas de perfil - si un usuario sólo tiene depósitos privados, el sitio web dará el mensaje *"Este usuario no tiene repositorios"*. El servicio está escrito enPython.

Git: Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux(git-scm, 2009).

JSON: Acrónimo de *JavaScript Object Notation*, Es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML (JSON LIB, 2010).

SOAP: (siglas de *Simple Object Access Protocol*) Es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web (W3C, 2000)

2.5 DESCRIPCIÓN GENERAL

2.5.1 Funciones del Producto

Dado el alcance del requerimiento se presentan las siguientes funciones que debe realizar:

- **Creación de los sensores:** Para esta fase del requerimiento se desarrolla la inclusión de los sensores como simulador y no como sensores físicos ubicados en los puntos estratégicos donde se genera mayor incidencia por los arroyos.
- **Creación de los puntos:** La creación de los puntos, permite identificar el estado de las posiciones exactas a los que el usuario desea obtener información con referencia al determinado arroyo.
- **Modificar y eliminar sensores:** Estas funciones se puede modificar o eliminar los sensores ya registrados por un usuario.
- **Modificar y eliminar puntos:** Estas funciones se puede modificar o eliminar los puntos que brindan información del estado del arroyo ya registrados por un usuario.
- **Creación de arroyos:** esta función permite la creación de un arroyo con un nombre y una ubicación.
- **Usuario administrador:** Es aquel usuario que accede a la aplicación a través de una cuenta única y puede realizar edición, creación y eliminación de información de todos los arroyos.

- **Usuario final:** el usuario final puede visualizar información en el mapa, crear rutas alternas en cualquier espacio del mapa.

2.6 RESTRICCIONES GENERALES

2.6.1 Políticas Regulatoras

Para esta fase del requerimiento, el software será desarrollado en lenguaje Ruby en el framework Rails de aplicaciones web de código abierto, por lo tanto no hay costo en licencia de herramientas de desarrollo de software, el servidor para alojar la herramienta será un servidor disponible por la universidad de la costa CUC sin ningún costo.

2.7 LIMITACIONES DE HARDWARE

Para la utilización de la aplicación web, se encuentra publicado en un servidor gratuito en la web con acceso <http://arroyoquilla.herokuapp.com/>.

Este sistema no tiene relación con ningún otro sistema, por tanto no se desarrollan interfaces con otras aplicaciones.

2.7.1 Funciones de Control

El control se realiza a través de la elección de un punto geográfico en el mapa al momento de crear un sensor o un punto de información del arroyo.

2.7.2 Requisitos del Lenguaje

La aplicación está desarrollada en lenguaje inglés.

2.7.3 Credibilidad de la Aplicación

Para esto se realizó un caso de pruebas y certificación con el usuario final para las debidas validaciones y el completo desarrollo del requerimiento.

2.7.4 Consideraciones de Seguridad

Para mantener la seguridad del sistema el usuario solo tendrá visualización de la información, por ningún momento tendrá acceso a manipulación directa de base de datos.

2.8 Especificar requerimientos

2.8.1 Requerimientos Funcionales

Ingreso a la Aplicación

Requerimiento No. 1				
Código	Nombre	Grado de Necesidad		
ERF_01	Ingreso a la aplicación	Esencial		
Descripción	Para el ingreso a la aplicación deberá seguir el siguiente enlace http://arroyoquilla.herokuapp.com/			
Entradas	Fuente	Salida	Destino	Restricciones
Usuario	Creación de sensores y puntos de los arroyos.	Página de inicio de la aplicación.	Demanda por parte del usuario a que arroyo visualizar o modificar	Modificaciones en la interfaz y visualización de los arroyos.
Proceso	La herramienta muestra información registrada por los usuarios de los arroyos y el estado de cada uno de ellos.			
Efecto colateral				

Tabla N. 5 Requerimiento ingreso a la aplicación

Requerimiento No. 2				
Código	Nombre	Grado de Necesidad		
ERF_01	Visualizar mapa localizador de arroyos	Esencial		
Descripción	El tamaño del mapa debe ser adecuado para una correcta visualización de los arroyos y el mapa debe estar en modo satélite.			
Entradas	Fuente	Salida	Destino	Restricciones
Usuario	Definición del tamaño estándar que va tener el mapa.	Visualización correcta del mapa	Usuarios en general.	No es modificable.
Proceso	La herramienta muestra información registrada por los usuarios de los arroyos y el estado de cada uno de ellos en el mapa con un tamaño fijo e invariable			
Efecto colateral				

Tabla N. 6 Requerimiento visualizar mapa localizador de arroyo

Requerimiento No. 3				
Código	Nombre	Grado de Necesidad		
ERF_01	Crear sensores	Esencial		
Descripción	Este requerimiento indica la creación y actualización de un sensor en el sistema SAT.			
Entradas	Fuente	Salida	Destino	Restricciones
Usuario	Coordenadas x y coordenadas y	Punto geográfico en el mapa.	Información almacenada en base de datos.	El sensor es creado localizando las coordenadas con el puntero del mouse sobre el mapa.
Proceso	La herramienta muestra la interfaz de creación o actualización de un sensor y el usuario selecciona con el puntero del mouse sobre el mapa en donde quiere localizarlo.			
Efecto colateral				

Tabla N. 7 Requerimiento crear sensores

Requerimiento No. 4				
Código	Nombre	Grado de Necesidad		
ERF_01	Crear puntos	Esencial		
Descripción	Este requerimiento indica la creación y actualización de un punto en el sistema SAT.			
Entradas	Fuente	Salida	Destino	Restricciones
Usuario	Coordenadas x y coordenadas y, estado del punto	Información del arroyo en el punto creado.	Información almacenada en base de datos.	El punto es creado localizando las coordenadas con el puntero del mouse sobre el mapa y seleccionando en la lista, el estado actual.
Proceso	La herramienta muestra la interfaz de creación o actualización de un punto y el usuario selecciona con el puntero del mouse sobre el mapa en donde quiere localizarlo y luego selecciona su estado.			
Efecto colateral				

Tabla N. 8 Requerimiento crear puntos

2.8.2 Obligaciones del Diseño de interfaces

2.8.2.1 Estándares Cumplidos

Se cumple con los estándares del desarrollo para plataformas web con base en el framework y el proceso que tiene para la organización de código orientado a objetos.

2.8.3_Requisitos no Funcionales

Se han determinado los siguientes requisitos no funcionales para el sistema, pero si para el usuario.

- Tamaño del marco del mapa
- Interfaz e interactividad de la aplicación con el usuario.
- Alertas de mensajes cuando se realizan creación o actualización de un arroyo.

3. DISEÑO

3.1 INTRODUCCIÓN

Como se dijo anteriormente en la metodología a emplear, para realizar el diseño de esta aplicación, se ha optado por seguir las pautas definidas por el lenguaje unificado de modelado UML (por sus siglas en inglés), este modelo dispone de diagramas que ayudan a comprender en un futuro la complejidad del sistema, por eso mismo es tan importante la especificación de los requisitos, ya este permite plasmar en un lenguaje las funcionalidades y requerimientos que se ha detectado en el sistema.

Aunque UML define una gran cantidad de diagramas para representar los distintos aspectos del desarrollo de la aplicación, en nuestro caso nos hemos centrado en tres tipos de diagramas: el diagrama de clases y el diagrama de casos de usos.

3.2 DIAGRAMA DE CLASES

El propósito de este diagrama es el de representar los objetos fundamentales del sistema, es decir los que percibe el usuario y con los que espera tratar para completar su tarea en vez de objetos del sistema o de un modelo de programación.

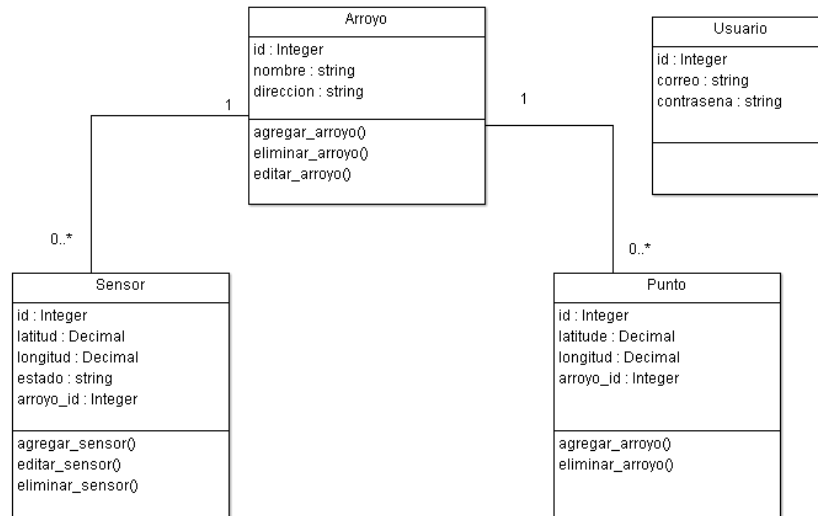


Gráfico N° 3 Diagrama de clases

3.2.1 Diccionario de Entidades

Clase Arroyo		
Atributos:	name, addres	Tipo:String
Archivo		.rb
rutaPerfiles		Rutas
Descripción de la clase		
Esta clase se pueden agregar los puntos y los sensores para tener la información pertinente de construir un arroyo.		

Tabla N. 9 Diccionario de entidades clase arroyo

Clase Sensor		
Atributos:	latitude, longitudo	Tipo: String
Archivo		.rb
rutaPerfiles		Rutas
Descripción de la clase		
En esta clase se gestionan los sensores que van a simular la ubicación del arroyo, el usuario puede crear, actualizar y eliminar uno varios sensor.		

Tabla N. 10 Diccionario de entidades clase sensor

Clase Punto		
Atributos:	latitude, longitudo status	Tipo: String
Archivo		.rb
rutaPerfiles		Rutas
Descripción de la clase		
En esta clase se gestionan los puntos que dan información del arroyo, donde el usuario puede crear, actualizar y eliminar uno varios puntos.		

Tabla N. 11 Diccionario de entidades clase punto

3.3 CASOS DE USO

3.3.1 Resumen Actores

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar (Alonso Amo, 2005).

Su ventaja principal es la facultad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. Los elementos básicos para la creación de un caso de uso son:

Actores: Representan un tipo de usuario del sistema. Se entendió con usuario cualquier cosa externa que interactúe con el sistema. No tiene que ser un ser humano, puede ser otro sistema informático, unidades organizativas o empresas(Alonso Amo, 2005).

Casos de uso: Es una tarea que debe poder llevarse a cabo con el apoyo con el sistema que se está desarrollando. Se representan mediante un óvulo. Cada caso de uso debe detallarse, habitualmente mediante una descripción textual(Alonso Amo, 2005).

Asociaciones: Hay una asociación entre actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso(Alonso Amo, 2005)..

Escenario: Es una interacción entre el sistema y los actores, que puede ser descrito mediante una secuencia de mensajes. Un caso de uso es una generalización de un escenario(satcaweb, 2012)

3.3.1.1 Actores

3.3.2 Casos de uso Gestionar Sensor

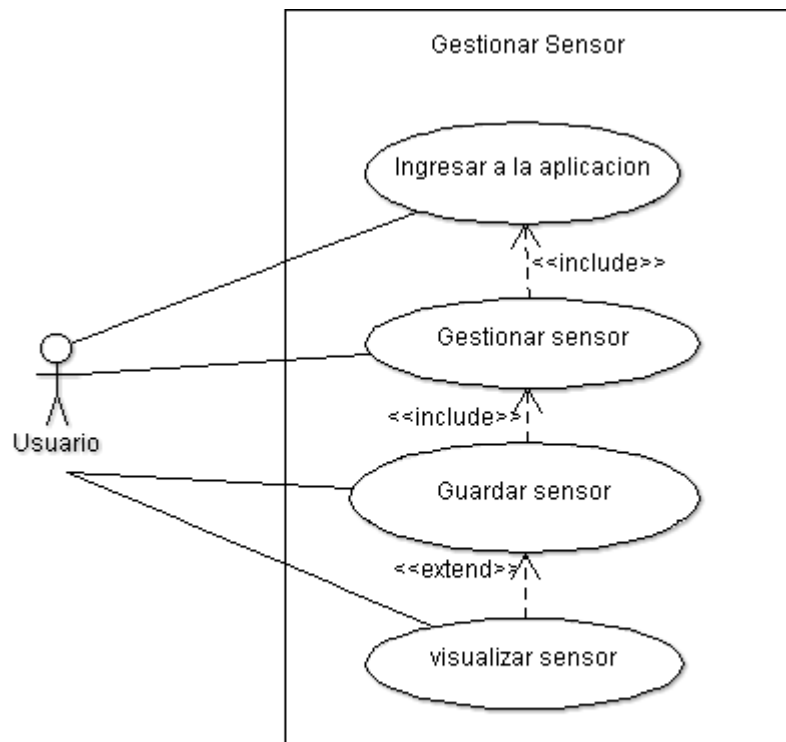


Gráfico N° 4 Caso de uso gestionar sensor

Identificador: GestionarSensor		
Descripción	El usuario es un visitante el cual no necesita de ningún registro para utilizar la herramienta, el módulo de gestionar un sensor debe permitir la creación, modificación, eliminación y visualizarlo en el mapa.	
Precondición	Ingreso al sistema	
Secuencia normal	Paso01	Ingreso al sistema
	Paso02	Gestionar sensor: crear, eliminar, actualizar y visualizar
	Paso03	Guardar cambios
	Paso04	Visualizar cambios
Importancia	Alta	
Urgencia	De inmediato – requerimiento funcional	

Tabla N. 12 Identificador gestionar sensor

3.3.3 Casos de uso Gestionar Punto

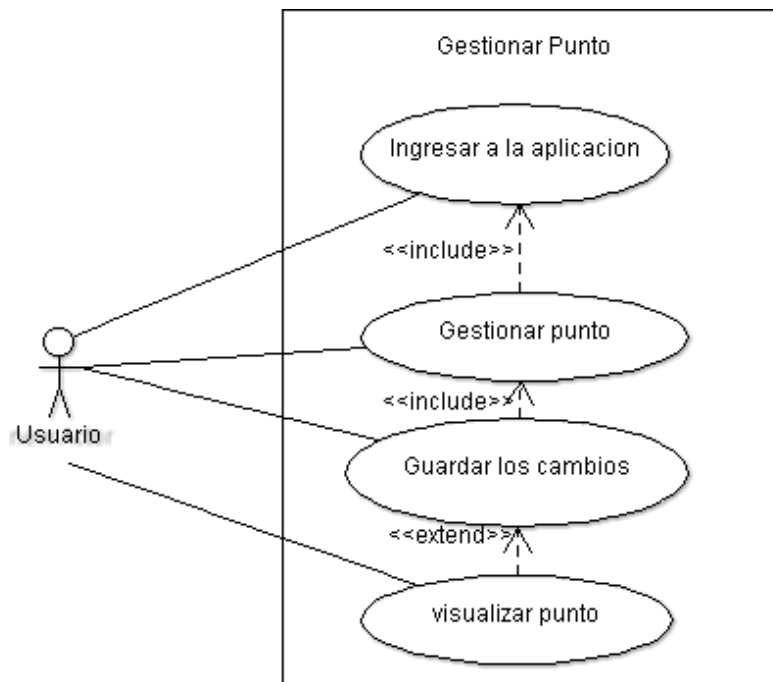


Gráfico N° 5 Caso de uso gestionar punto

Identificador: GestionarPunto		
Descripción	El usuario es un visitante el cual no necesita de ningún registro para utilizar la herramienta, el módulo de gestionar un punto debe permitir la creación, modificación, eliminación y visualizarlo en el mapa.	
Precondición	Ingreso al sistema	
Secuencia normal	Paso01	Ingreso al sistema
	Paso02	Gestionar punto: crear, eliminar, actualizar y visualizar
	Paso03	Guardar cambios
	Paso04	Visualizar cambios
Importancia	Alta	
Urgencia	De inmediato – requerimiento funcional	

Tabla N. 13 Identificador gestionar punto

3.3.4 Casos de uso Visualizar Arroyo

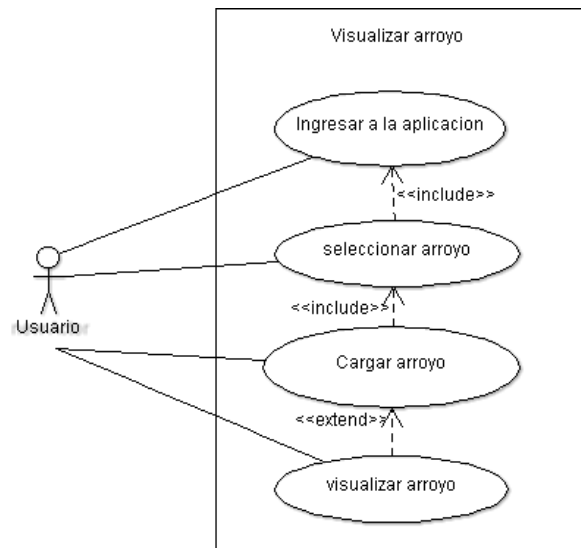


Gráfico N° 6 Caso de uso visualizar arroyo

Identificador: VisualizarArroyo		
Descripción	Visualizar el mapa y adyacente una lista desplegable donde pueda seleccionar los arroyos que desea ver como puede ser uno como pueden ser todos.	
Precondición	Ingreso al sistema	
Secuencia normal	Paso01	Página principal
	Paso02	Seleccionar uno o todos los arroyos que se desea visualizar
	Paso03	Visualizar arroyo en mapa
Importancia	Medio	
Urgencia	Baja necesidad– requerimiento no funcional	

Tabla N. 14 Identificador visualizarArroyo

3.4 Diccionario de Entidades base de datos

Tabla Arroyo		
Campo	Tipo	Descripción
Id	Integer	Identificador del arroyo
Name	String	Nombre del arroyo
Address	String	Dirección del arroyo
Archivo	arroyo.rb	
Relaciones	Has_many :sensors, has_many :points	
Descripción de la tabla		
Se almacenan todos los arroyos indicando el nombre y dirección que son registrados por un administrador del sistema.		

Tabla N. 15 Diccionario de entidades tabla arroyo

Tabla Sensor		
Campo	Tipo	Descripcion
Id	Integer	Identificador del sensor
Latitude	String	Coordenada latitud del sensor
Longitude	String	Coordenada longitud del sensor
Archivo		sensor.rb
Relaciones		Belongs_to :arroyo
Descripción de la tabla		
Se almacenan todos los sensores para indicar el nivel de peligrosidad del arroyo.		

Tabla N. 16 Diccionario de entidades tabla sensor

Tabla Punto		
Campo	Tipo	Descripción
Id	Integer	Identificador del punto
latitude	String	Coordenada latitud del punto
longitude	String	Coordenada longitud del punto
Archivo		point.rb
Relaciones		Belongs_to :arroyo
Descripción de la tabla		
Se almacenan todos los puntos para trazar la ruta del arroyo.		

Tabla N. 157 Diccionario de entidades tabla punto

Tabla Usuario		
Campo	Tipo	Descripcion
Id	Integer	Identificador del usuario administrador
Name	String	Nombre del usuario
Password		Contraseña del usuario
Confirm_password	String	Confirmación de la contraseña
Correro		Correo del usuario
Archivo		user.rb
Relaciones		Belongs_to :arroyo
Descripción de la tabla		
Se almacenan el usuario administrador del sistema.		

Tabla N. 168 Diccionario de entidades tabla usuario

4. ARQUITECTURA

4.1 INTRODUCCIÓN

El diseño es el proceso que extiende, refina y reorganiza los aspectos detectados en el proceso de modelado conceptual para generar una especificación rigurosa del sistema de información siempre orientada a la obtención de la solución del sistema software.

Para el diseño del sitio web se ha escogido una arquitectura de tres capas (presentación, negocio y conocimiento).

4.2 ARQUITECTURA DEL SISTEMA



Gráfico N° 7 Arquitectura del sistema

4.3 CAPAS DE APLICACION

4.3.1 Descripción de la arquitectura

Grafica N°7 muestre en detalle cada uno de los procesos que intervienen en la aplicación, desde el usuario realizando una petición ya sea por medio de un dispositivo móvil o desde cualquier browser de una computadora, esta información es enviada al servidor de aplicaciones el cual la recibe y por medio de una formato de datos se reciben del sensor las coordenadas donde ubicado, la solicitud es devuelta y validada en el servidor de aplicaciones con la información almacenada de ese punto en la base de datos, para ser presentada al usuario en forma gráfica.

4.4 DIAGRAMA DE INTERFAZ

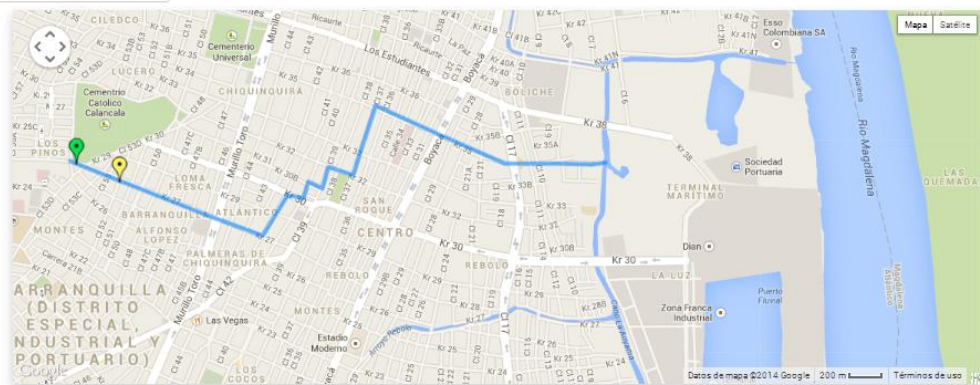
4.4.1 Interfaz Principal



SAT sistema de alertas tempranas

Listar todos los arroyos

Todos los arroyos



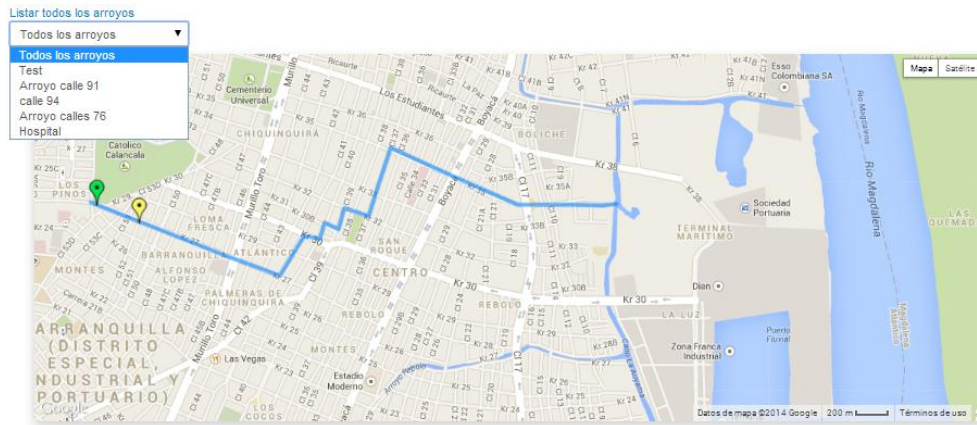
Proyecto de grado, simulación sistema de alertas tempranas (SAT)
Universidad de la costa CUC
Luis Rodriguez, Diego Hernandez
Barranquilla - Colombia

Gráfico N° 8 Interfaz principal

4.4.2 Interfaz para visualizar uno o todos los arroyos registrados en el sistema.



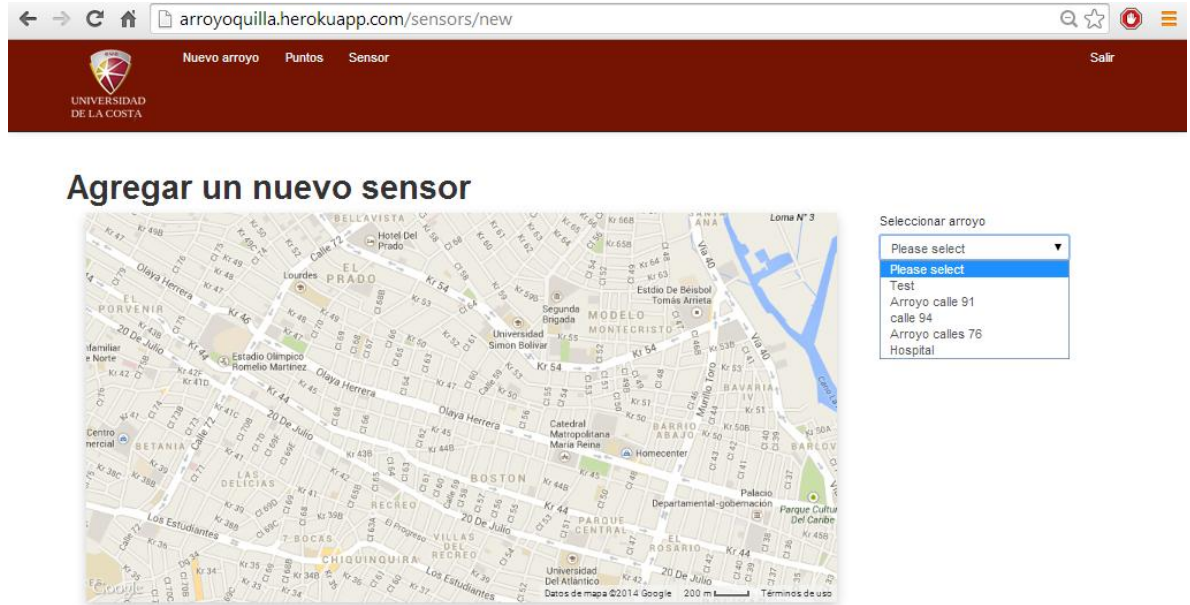
SAT sistema de alertas tempranas



Proyecto de grado, simulación sistema de alertas tempranas (SAT)
Universidad de la costa CUC
Luis Rodríguez, Diego Hernández
Barranquilla - Colombia

Gráfico N° 9 Interfaz visualizar arroyos

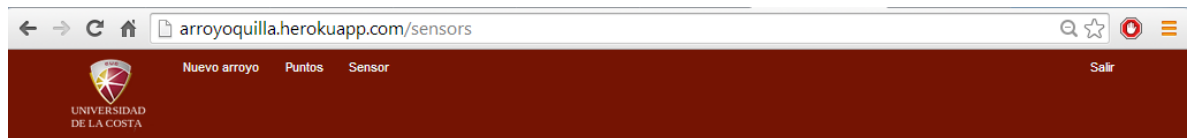
4.4.3 Interfaz para crear un sensor.



Proyecto de grado, simulación sistema de alertas tempranas (SAT)
 Universidad de la costa CUC
 Luis Rodríguez, Diego Hernández
 Barranquilla - Colombia

Gráfico N° 10 Interfaz crear sensor

4.4.4 Interfaz para visualizar los sensores que se encuentran registrados en el sistema.

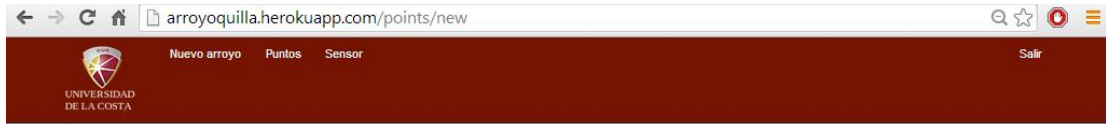


Latitud	Longitud	Estado	Arroyo		
10.9723281068722	-74.797334522009	good	Hospital		
10.9714012274137	-74.7951029241086	medium	Hospital		

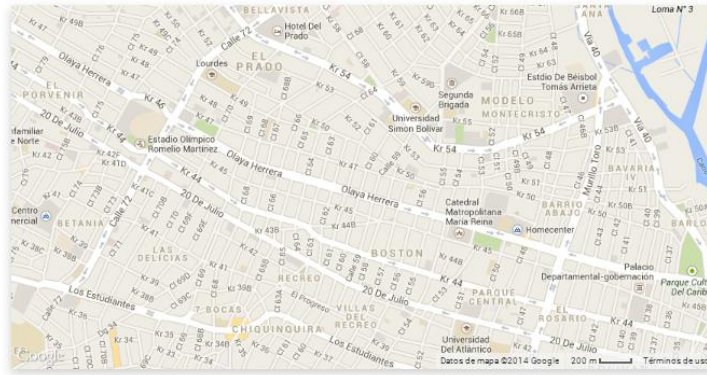
Proyecto de grado, simulación sistema de alertas tempranas (SAT)
 Universidad de la costa CUC
 Luis Rodríguez, Diego Hernández
 Barranquilla - Colombia

Gráfico N° 11 Interfaz visualizar sensor

4.4.5 Interfaz para crear un punto.



Agregar un nuevo punto



Seleccionar un arroyo

Please select
Please select
Test
Arroyo calle 91
calle 94
Arroyo calles 76
Hospital

Proyecto de grado, simulación sistema de alertas tempranas (SAT)
Universidad de la costa CUC
Luis Rodríguez, Diego Hernández
Barranquilla - Colombia

Gráfico N° 12 Interfaz para crear punto

4.4.6 Interfaz para visualizar los puntos registrados en el sistema.

Latitud	Longitud	Arroyo	
11.0045985500084	-74.8257444798946	Arroyo calle 91	✘
11.0045985500084	-74.8257444798946	Arroyo calle 91	✘
11.0045985500084	-74.8257444798946	Arroyo calle 91	✘
11.0044300451157	-74.8259161412716	Arroyo calle 91	✘
11.0081792561944	-74.824542850256	Arroyo calle 91	✘
11.009948530242	-74.8231695592403	Arroyo calle 91	✘
11.0139504204196	-74.8199079930782	Arroyo calle 91	✘
11.0158881582211	-74.8186810036373	Arroyo calle 91	✘
11.0181628776228	-74.8156164586544	Arroyo calle 91	✘
11.0214485523811	-74.8151014745235	Arroyo calle 91	✘
11.0234704876388	-74.8148439824581	Arroyo calle 91	✘
11.0260821338991	-74.8146294057369	Arroyo calle 91	✘

Gráfico N° 13 Interfaz para visualizar puntos

5. IMPLEMENTACION

5.1 INTRODUCCIÓN

Con base en el levantamiento de la información y el proceso de análisis del requerimiento se realiza el desarrollo del proyecto. Para esta primera fase se crea la simulación de los sensores, creación, modificación y eliminación de los mismos ya que en una posterior fase estos simuladores serán implementados de manera estratégica que brinden información relevante de los arroyos más peligrosos de la ciudad de barranquilla.

5.2 CONCEPTOS UTILIZADOS

El framework utilizado, Ruby on Rails contiene el modelo de arquitectura mencionado, esto facilita la programación y mantiene el código organizado, limpio y reutilizable. Al crear un proyecto RoR, el framework realiza la creación automática de carpetas en donde se almacenará la información, llevando el patrón MVC.

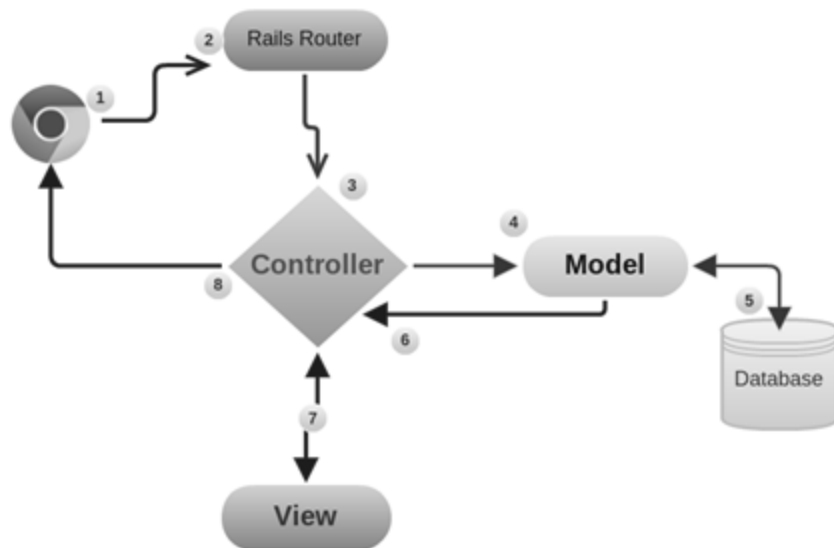


Gráfico N° 14 Arquitectura MVC

MODELO

En las aplicaciones web orientadas a objetos sobre bases de datos, el Modelo consiste en las clases que representan a las tablas de la base de datos.

En Ruby onRails, las clases del Modelo son gestionadas por **ActiveRecord**. Por lo general, lo único que tiene que hacer el programador es heredar un de la clase ActiveRecord:Base, y el programa averiguará automáticamente qué tabla usar y qué columnas tiene(Sommerville, 2005).

VISTA

En MVC,es la lógica de visualización, o cómo se muestran los datos de las clases del *Controlador*. Con frecuencia en las aplicaciones web la vista consiste en una cantidad mínima de código incluido en HTML.

Existen en la actualidad muchas maneras de gestionar las vistas. El método que se emplea en Rails por defecto es usar Ruby Empotrado (archivos.rhtml, desde la versión 2.x en adelante de RoR archivos.html.erb), que son básicamente fragmentos de código HTML con algo de código en Ruby, siguiendo una sintaxis similar a JSP.

Es necesario escribir un pequeño fragmento de código en HTML para cada método del controlador que necesita mostrar información al usuario. El "maquetado" o distribución de los elementos de la página se describe separadamente de la acción del controlador y los fragmentos pueden invocarse unos a otros(Sommerville, 2005).

CONTROLADOR

En MVC, las clases del *Controlador* responden a la interacción del usuario e invocan a la lógica de la aplicación, que a su vez manipula los datos de las clases del *Modelo* y muestra los resultados usando las *Vistas*. En las aplicaciones web basadas en MVC, los métodos del *controlador* son invocados por el usuario usando el navegador web.

La implementación del *Controlador* es manejada por el ActionPack de Rails, que contiene la clase *ApplicationController*. Una aplicación Rails simplemente hereda de esta clase y define las acciones necesarias como métodos, que pueden ser invocados desde la web, por lo general en la forma `http://aplicacion/ejemplo/metodo`, que invoca a `EjemploController#método`, y presenta los datos usando el archivo de plantilla `/app/views/ejemplo/método.html.erb`, a no ser que el método redirija a algún otro lugar(Sommerville, 2005).

6. PRUEBAS

6.1 INTRODUCCIÓN

Para realizar las pruebas del requerimiento se realizan un proceso de certificación de pruebas unitarias y pruebas de certificación de usuario.

6.2 VALIDACIONES

6.2.1 Validación de base de datos

Se valida que los sensores y los puntos se almacenen correctamente en la base de datos y que la lógica de negocio sea la correcta y finalmente se visualice el resultado en el mapa de la aplicación.

6.2.2 Validación de navegadores

Se realizan pruebas de la aplicación en los diferentes navegadores más utilizados:

Internet Explorer 8

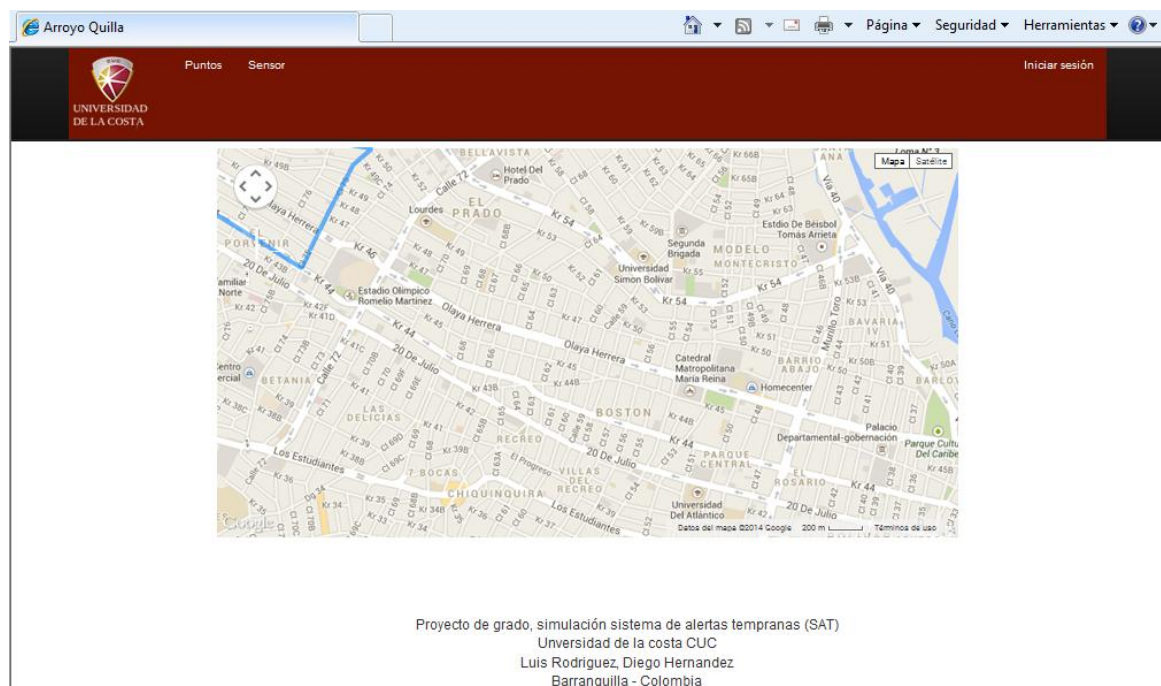


Gráfico N° 15 Validación de navegador internet Explorer

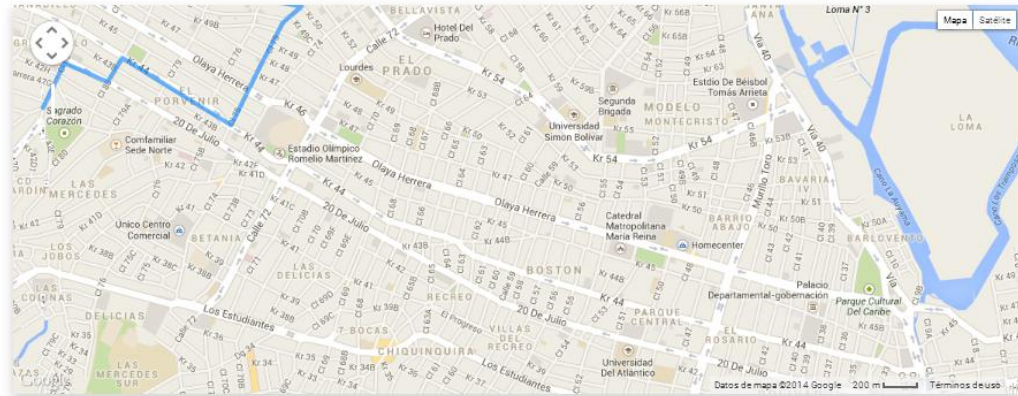
Google Chrome



SAT sistema de alertas tempranas

Listar todos los arroyos

Todos los arroyos



Proyecto de grado, simulación sistema de alertas tempranas (SAT)
Universidad de la costa CUC
Luis Rodriguez, Diego Hernandez

Gráfico N° 16 Validación de navegador google Chrome

Safari

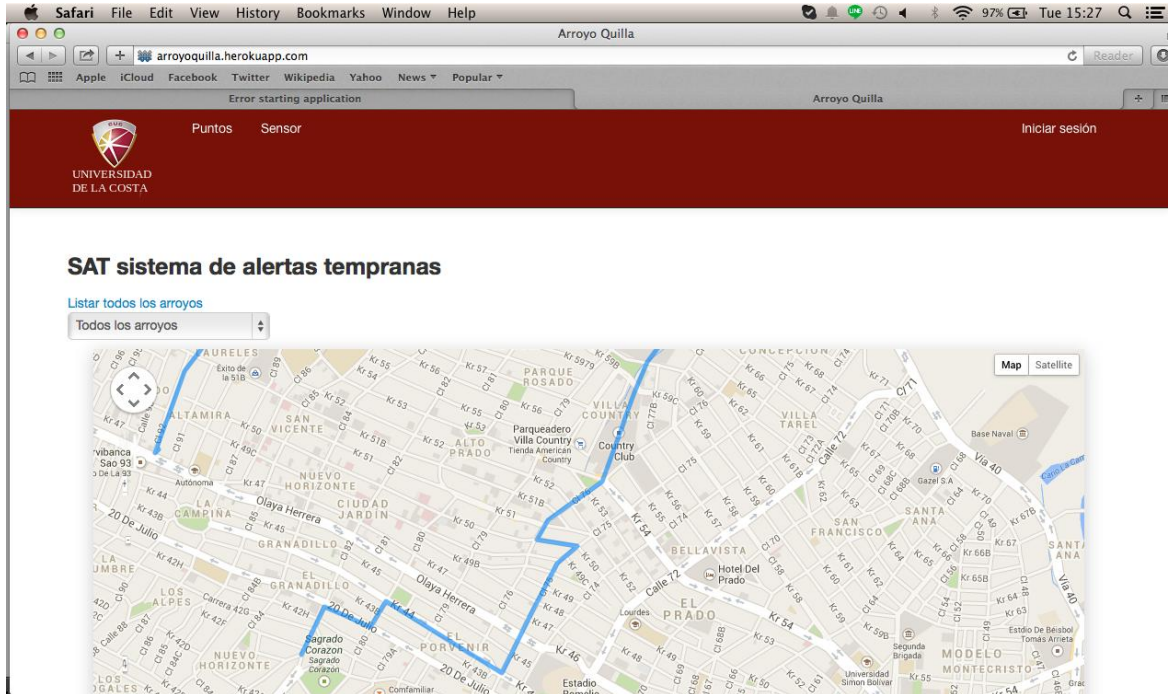


Gráfico N° 17 Validación de navegador Safari

6.3 PRUEBAS

6.3.1 Gestión de sensores

CASOS DE PRUEBA							
Registro de Sensores							
ID caso de prueba	Modulo a probar	Descripción	Pre-Requisito	Funcionamiento		Resultado obtenido /Observaciones	Estado
				Si	No		
CO_001	Sensores	La herramienta debe permitir que el usuario realice la creación, actualización y eliminación de un sensor.	El usuario debe ingresar a la interfaz de creación de los sensor por medio del link: http://arroyoquilla.herokuapp.com/points	x		Se registra de manera satisfactoria el punto en la base de datos	Exitoso

Tabla N. 19 Prueba gestión de sensores

6.3.2 Gestión de puntos

CASOS DE PRUEBA							
Registro de Puntos							
ID caso de prueba	Modulo a probar	Descripción	Pre-Requisito	Funcionamiento		Resultado obtenido /Observaciones	Estado
				Si	No		
CO_001	Puntos	La herramienta debe permitir que el usuario realice la creación, actualización y eliminación de un punto.	El usuario debe ingresar a la interfaz de creación de los punto por medio del link: http://arroyoquilla.herokuapp.com/points	x		Se registra de manera satisfactoria el punto en la base de datos	Exitoso

Tabla N. 17 Prueba gestión de puntos

6.4 CONCURRENCIA DE LA INFORMACIÓN

Se realizaron pruebas de creación, edición actualización de arroyos y sensores, como también creación y eliminación de puntos en los arroyos en 15 máquinas conectadas en línea, entre computadores de mesa, celulares y portátiles, dando un resultado satisfactorio en la visualización el funcionamiento en línea de la plataforma.

7. CONCLUSIONES

El prototipo de sistema de alertas temprana, es una herramienta que aporta información relevante de los arroyos que para esta fase se toma como piloto de prueba la ciudad de barranquilla, ya que cualquier ciudad con problemáticas de arroyos debido a fuertes lluvias y con problemas de alcantarillado puede aplicar. La herramienta brinda información que indica al usuario del estado sensible de los arroyos como: el nivel de peligrosidad de un determinado arroyo, la ubicación, el estado del arroyo, puntos geográficos del arroyo desde donde inicia hasta donde finaliza, una interfaz visual que el usuario puede sin ninguna dificultad ver uno o todos los arroyos registrados en el sistema; el sistema cuenta con una tecnología aplicada a un excelente rendimiento y escalabilidad para próximas fases del proyecto.

Con base en en los resultados obtenidos se logró de manera satisfactoria todo el desarrollo e implementación del proyecto, el cumplimiento de cada uno de los requerimientos funcionales y no funcionales, dando por completo éxito la culminación de herramienta aunque de manera social no da por completo una solución que acabe con los arroyos de la ciudad, esta herramienta ayudará a los usuarios a evitar grandes accidentes que hoy día se generan a raíz de falta de información y de estar prevenidos o alertados.

BIBLIOGRAFÍA

- Alcaldía Distrital. (23 de Octubre de 2013). *Barranquilla*. Obtenido de Barranquilla: <http://www.barranquilla.gov.co/forohidrico/problematika.html>
- Alonso Amo, L. M. (2005). *Introducción a la ingeniería del software*. Madrid: Delta publicaciones.
- Arroyos de Barranquilla. (23 de Octubre de 2013). *Arroyos de Barranquilla*. Recuperado el Lunes de Noviembre de 2013, de Arroyos de Barranquilla: <http://www.arroyosdebarranquilla.co/pedagogia/antecedentes-historicos>
- bitbucket. (s.f.). *bitbucket*. Recuperado el Jueves de Febrero de 2014, de bitbucket: <https://bitbucket.org/features>
- Coral calero, M. P. (2010). *Calidad del producto y proceso del software*. España: RM-Editorial.
- culturadigital*. (abril de 2011). Recuperado el 4 de febrero de 2014, de <http://culturadigital.cl/wp/sistema-de-alerta-temprana-en-celulares-adjudicado-por-el-gobierno/>
- David Heinemeier Hansson. (s.f.). *guides.rubyonrails*. Recuperado el Enero de martes de 2014, de guides.rubyonrails: <http://guides.rubyonrails.org/>
- EL HERALDO. (7 de Septiembre de 2013). El problema crónico de los arroyos. *ELHERALDO.COM*, pág. 4.
- fews*. (1985). Recuperado el 1 de febrero de 2014, de <http://www.fews.net/es>
- git-scm. (2009). *git-scm*. Recuperado el Miercoles de Febrero de 2014, de git-scm: <http://git-scm.com/>
- Hartl, M. (2013). *Ruby on Rails Tutorial, 2nd Edition*. New York: Pearson.
- <http://earlywarning.usgs.gov>. (s.f.). Recuperado el 3 de Febrero de 2014, de <http://earlywarning.usgs.gov/fews/>
- <http://earlywarning.usgs.gov/>. (2012). *usgs*. Recuperado el 4 de Febrero de 2014, de <http://earlywarning.usgs.gov/>
- <http://www.sems.gob.m>. (s.f.). Recuperado el 2 de Febrero de 2014, de http://www.sems.gob.mx/en_mx/sems/sistema_alerta_temprana_siat

- JSON LIB. (14 de Diciembre de 2010). *JSON - LIB*. Recuperado el Miercoles de Octubre de 2013, de JSON - LIB: <http://json-lib.sourceforge.net/>
- Kimmel, P. (2008). *Manual de UML*. Mexico: Mac Graw Hill.
- Kimmel, P. (2008). *Manual de UML*. Mexico: Mac Graw Hill.
- Microsoft Corporation. (Diciembre de 2006). *La Arquitectura Orientada a Servicios (SOA) de Microsoft* .
- nces. (abril de 2012). Recuperado el 5 de 2 de 2014, de http://nces.ed.gov/programs/slds/pdf/Early_Warning_Systems_FINAL.pdf
- nces. (Abril de 2012). nces. Recuperado el 5 de Febrero de 201, de http://nces.ed.gov/programs/slds/pdf/Early_Warning_Systems_FINAL.pdf
- Rumbaugh, J. e. (1996). *Modelado y diseño orientados a objetos*. Madrid: Prentice Hall.
- satcaweb. (2012). *satcaweb*. Recuperado el 2 de Febrero de 2014, de satcaweb: <http://www.satcaweb.org/alertatemprana/inicio/satcaweb.aspx>
- sems. (2013). Recuperado el 2 de Febrero de 2014, de http://www.sems.gob.mx/en_mx/sems/sistema_alerta_temprana_siat
- SENA. (Jueves de Octubre de 2013). *Observatorio concejo visible barranquilla*. Recuperado el Martes de Septiembre de 2013, de Observatorio concejo visible barranquilla: <http://www.cvisible.com/cvquilla/inicio.html>
- siata. (s.f.). <http://siata.gov.co>. Recuperado el 2 de Febrero de 2014, de <http://siata.gov.co/newpage/web/nosotros.php>
- Sommerville, I. (2005). *Ingenieria del software*. Madrid: Pearson.
- W3C. (Lunes de Mayo de 2000). *W3C*. Recuperado el Viernes de Octubre de 2013, de W3C: <http://www.w3.org/TR/soap/>

ANEXO 1
CARTA DE ENTREGA Y AUTORIZACIÓN DE LOS AUTORES PARA LA CONSULTA, LA REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO DE TESIS Y TRABAJOS DE GRADO

Barranquilla, 15 de agosto de 2014

Fecha Marque con una X
Tesis Trabajo de Grado

Yo, Diego Armando Hernandez Villalba identificado con C.C. No. 1.140.837.122 y Luis Javier Rodriguez Zarate identificado con C.C No. 1.048.212.092, actuando en nombre propio y como autores de la tesis y/o trabajo de grado titulado DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SOFTWARE DE SISTEMA DE ALERTAS TEMPRANAS PARA LA PROBLEMÁTICA DE ARROYOS EN LA CIUDAD DE BARRANQUILLA presentado y aprobado en el año 2014 como requisito para optar al título de Ingeniero de Sistemas; se hace entrega del ejemplar respectivo y de sus anexos de ser el caso, en formato digital o electrónico (DVD) y autorizo a la UNIVERSIDAD DE LA COSTA, CUC, para que en los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia, utilice y use en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador de la obra objeto del presente documento.

Y se autoriza a la Unidad de información, para que con fines académicos, muestre al mundo la producción intelectual de la Universidad de la Costa, CUC, a través de la visibilidad de su contenido de la siguiente manera:

Los usuarios puedan consultar el contenido de este trabajo de grado en la página Web de la Facultad, de la Unidad de información, en el repositorio institucional y en las redes de información del país y del exterior, con las cuales tenga convenio la institución y Permita la consulta, la reproducción, a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato DVD o digital desde Internet, Intranet, etc., y en general para cualquier formato conocido o por conocer.

LUIS JAVIER RODRIGUEZ ZARATE Y DIEGO ARMANDO HERNANDEZ VILLALBA, manifiestan que la obra objeto de la presente autorización es original y se realizó sin violar o usurpar derechos de autor de terceros, por lo tanto la obra es de su exclusiva autoría y detenta la titularidad ante la misma. PARÁGRAFO: En caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, LUIS JAVIER RODRIGUEZ ZARATE Y DIEGO ARMANDO HERNANDEZ VILLALBA, asuman toda la responsabilidad, y saldrá en defensa de los derechos aquí autorizados; para todos los efectos, la Universidad actúa como un tercero de buena fe.

Para constancia se firma el presente documento en dos (02) ejemplares del mismo valor y tenor, en Barranquilla D.E.I.P., a los 15 días del mes de agosto de Dos Mil Catorce 2014.

LUIS JAVIER RODRIGUEZ ZARATE. _____

FIRMA

ARMANDO HERNANDEZ VILLALBA. _____

FIRMA

ANEXO 2
FORMULARIO DE LA DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADO

TÍTULO COMPLETO DE LA TESIS O TRABAJO DE GRADO: DISEÑO Y DESARROLLO DE UN PROTOTIPO DE SOFTWARE DE SISTEMA DE ALERTAS TEMPRANAS PARA LA PROBLEMÁTICA DE LOS ARROYOS EN LA CIUDAD DE BARRANQUILLA.

SUBTÍTULO, SI LO TIENE: NO APLICA

AUTOR AUTORES

Apellidos Completos	Nombres Completos
Rodríguez Zarate Hernández Villalba	Luis Javier Diego Armando

DIRECTOR (ES)

Apellidos Completos	Nombres Completos
Sepúlveda Ojeda	Jorge Antonio

JURADO (S)

Apellidos Completos	Nombres Completos
De la Hoz Manotas	Alexis Kevin

ASESOR (ES)

Apellidos Completos	Nombres Completos
Sepúlveda Ojeda	Jorge Antonio

TRABAJO PARA OPTAR POR EL TITULO DE: Ingeniero de Sistemas

FACULTAD: Ingeniería

PROGRAMA: Pregrado X Especialización

NOMBRE DEL PROGRAMA: Ingeniería de Sistemas

CIUDAD: Barranquilla **AÑO DE PRESENTACIÓN DEL TRABAJO DE GRADO:** 2014

NÚMERO DE PÁGINAS 58

TIPO DE ILUSTRACIONES:

- | | |
|--|--------------------------------------|
| <input type="checkbox"/> Ilustraciones | <input type="checkbox"/> Planos |
| <input type="checkbox"/> Láminas | <input type="checkbox"/> Mapas |
| <input type="checkbox"/> Retratos | <input type="checkbox"/> Fotografías |

Tablas, gráficos y diagramas

MATERIAL ANEXO (Vídeo, audio, multimedia o producción electrónica):

Duración del audiovisual: _____ minutos.

Número de casetes de vídeo: _____ Formato: VHS ___ Beta Max ___ ¼ ___ Beta Cam _____

Mini DV ___ DV Cam ___ DVC Pro ___ Vídeo 8 ___ Hi 8 _____

Otro. Cuál? _____

Sistema: Americano NTSC _____ Europeo PAL _____ SECAM _____

Número de casetes de audio: _____

Número de archivos dentro del DVD (En caso de incluirse un DVD diferente al trabajo de grado): _____

PREMIO O DISTINCIÓN (En caso de ser LAUREADAS o tener una mención especial): _____

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS: Son los términos que definen los temas que identifican el contenido. (En caso de duda para designar estos descriptores, se recomienda consultar con la Unidad de Procesos Técnicos de la Unidad de información en el correo biblioteca@cuc.edu.co, donde se les orientará).

ESPAÑOL

Arroyos, nivel de peligrosidad, mapas, sensores, casos de uso, MVC, Ingeniería del software, aplicación web, sistema de alertas tempranas.

INGLÉS

streams, danger level, maps, sensors, cases of use, MVC, Software Engineering, Web App, early warning system

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS:(Máximo 250 palabras-1530 caracteres):

El desarrollo de la investigación presenta una aplicación web, en el cual se alerta al ciudadano local sobre los arroyos y su nivel de peligrosidad en la ciudad de Barranquilla.

Para el desarrollo del proyecto se utilizó el lenguaje de programación Ruby con el framework Rails el cual usa como patrón de diseño la arquitectura MVC, la metodología utilizada para el proceso del desarrollo del ciclo de vida del software fue la metodología de Rumbaugh por medio de ella se realizó el proceso de análisis del objeto en donde se plasmó todo el proceso de levantamiento de información y captura de requisitos, el proceso de diseño el cual una vez realizado el levantamiento de información, se realizaron los diagramas de casos de uso y los diagramas de clases para determinar, delimitar y detallar el esquema final del prototipo y por último el proceso de implementación el cual se aplicaron los diagramas realizados y los requisitos funcionales y no funcionales; finalmente se realizan las pruebas de la herramienta para la comprobación funcional de todo el

sistema, dando como resultado un prototipo final que permite a los usuarios estar informado de los niveles de peligrosidad que cuentan los arroyos de la ciudad en temporadas invernales.

The development of the research presents a web application, which alerts the local citizen on streams and their level of hazard in the city of Barranquilla.

For the project the programming language Ruby with the Rails framework which uses as the MVC design pattern architecture was used, the methodology used for the development process of the software life cycle methodology Rumbaugh was through her the scan of the object in which the whole process of information gathering and capture requirements, the design process which once made the information gathering was held was reflected, diagrams and use case diagrams were performed classes to determine, define and detail the final outline of the prototype and finally the implementation process which diagrams made and implemented functional and non-functional requirements; finally testing tool for functional testing of the complete system are made, resulting in a final prototype that allows users to be informed of the danger levels that have streams of the city in winter seasons.