



UNIVERSIDAD
NACIONAL
DE LOJA

TT-CIS-XB-006



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“Desarrollo de una aplicación web de análisis estadístico y monitoreo de una cuenta de Twitter”

Tesis previa a la obtención del título de Ingeniero en Sistemas

Autor:

- Franklin de Jesús Michay Cuenca

Director:

- Ing. Luis Roberto Jácome Galarza, Mg. Sc.

LOJA-ECUADOR

2017

CERTIFICACIÓN DEL DIRECTOR

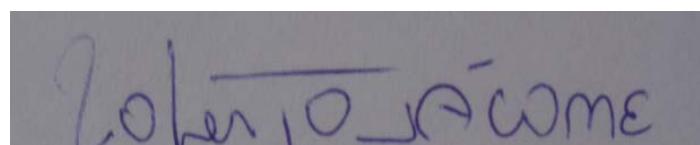
Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS

CERTIFICA:

Que el egresado Franklin de Jesús Michay Cuenca, autor del presente trabajo de titulación cuyo tema versa sobre “DESARROLLO DE UNA APLICACIÓN WEB DE ANÁLISIS ESTADÍSTICO Y MONITOREO DE UNA CUENTA DE TWITTER”, ha sido dirigido, orientado y discutido bajo mi asesoramiento y reúne a satisfacción los requisitos exigidos en una investigación de este nivel por lo cual autorizo su presentación y sustentación.

Loja, 01 de diciembre de 2016

A handwritten signature in blue ink, appearing to read "Luis Roberto Jácome".

Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

AUTORÍA

Yo, **FRANKLIN DE JESÚS MICHAY CUENCA**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Firma:



Cédula: 1104993157

Fecha: 10 de mayo de 2017

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **FRANKLIN DE JESÚS MICHAY CUENCA**, declaro ser el autor de la tesis titulada: **“DESARROLLO DE UNA APLICACIÓN WEB DE ANÁLISIS ESTADÍSTICO Y MONITOREO DE UNA CUENTA DE TWITTER”**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los diez días del mes de mayo del dos mil diecisiete.

Firma:



Autor: Franklin de Jesús Michay Cuenca

Cédula: 1104993157

Dirección: Loja (Av. Pio Jaramillo Alvarado y México)

Correo Electrónico: franklinmichay@gmail.com, fdmichayc@unl.edu.ec

Teléfono: 2106111

Celular: 0969879852

DATOS COMPLEMENTARIOS:

Director de Tesis: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Tribunal de Grado: Ing. Mario Andrés Palma Jaramillo, Mg. Sc.

Ing. Alex Vinicio Padilla Encalada, Mgs.

Ing. Carlos Miguel Jaramillo Castro, Mg. Sc.

DEDICATORIA

El desarrollo del presente trabajo de titulación lo dedico a Dios por ser mi guía, a mis padres, Luis y Mercedes por su amor y apoyo incondicional y a todos quienes me brindaron su mano amiga para cumplir con este objetivo.

AGRADECIMIENTO

Mi agradecimiento va dirigido en primer lugar hacia Dios por acompañarme a lo largo de mi vida y ayudarme en cada dificultad. De la misma forma a la Universidad Nacional de Loja, la Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables, a la Carrera de Ingeniera en Sistemas, y a todos y cada uno de los docentes que forjaron mi vida estudiantil, al impartir los conocimientos necesarios para llegar a la culminación de mi carrera profesional.

De igual forma, al Ingeniero Luis Roberto Jácome Galarza, Mg. Sc., profesional de altos valores profesionales y humanos, quien con su paciencia, responsabilidad y dedicación supo guiarme en el desarrollo del presente trabajo de titulación.

Índice de Contenidos

CERTIFICACIÓN DEL DIRECTOR.....	ii
AUTORÍA.....	iii
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.	iv
DEDICATORIA	v
AGRADECIMIENTO.....	vi
Índice de Contenidos	vii
Índice de Figuras.....	xii
Índice de Tablas	xiii
a. <i>Título</i>	1
b. <i>Resumen</i>	2
<i>Summary</i>	4
c. <i>Introducción</i>	6
d. <i>Revisión de Literatura</i>	8
CAPÍTULO I.....	8
1.1. Aplicaciones Web	8
1.1.1. Páginas Web	8
1.1.2. Sitio Web	8
1.1.3. Estándares Web.....	8
1.1.4. Beneficios del uso de estándares web	10
1.2. Redes Sociales	10
1.2.1. Twitter	11
1.2.1.1. Interacciones entre usuarios y glosario de términos comunes.....	11
1.2.1.2. Características	12
1.2.1.3. Temporalidad de la información	13
1.2.1.4. Información disponible.....	14
1.2.2. Herramientas para la Gestión de una Cuenta de Twitter	16
1.2.2.1. Autenticación OAuth	16
1.2.2.2. API de Twitter	17
1.2.2.3. Limitaciones de los APIs de Twitter	17
1.2.2.4. Rest API de Twitter	18
1.2.3. Herramientas de análisis de datos de Twitter	20
1.2.4. Twitonomy.....	20

1.2.5.	Trendinalia.....	21
1.3.	Sistema Integrado de Seguridad ECU911 Loja.....	22
1.3.1.	Objetivos estratégicos	23
1.3.2.	Entidades coordinadas.....	23
1.3.3.	Estadísticas.....	24
1.3.4.	Servicios que presta ECU 911 para una Seguridad Ciudadana Integral	24
1.3.5.	Geolocalización	25
1.3.6.	Aplicación Smartphone ECU911	26
1.3.7.	Feriado seguro	27
1.3.8.	Estadio Seguro:	27
CAPÍTULO II.....	28	
2.1.	Herramientas para el desarrollo de Aplicaciones Web	28
2.1.1.	Herramientas para el almacenamiento de datos	28
2.1.1.1.	Mysql	28
2.1.1.2.	PostgreSQL	30
2.1.2.	Comparativa entre herramientas de Bases de Datos para el desarrollo de Aplicaciones Web	31
2.2.	Lenguaje de programación	32
2.2.1.	Java	32
2.2.2.	PHP.....	33
2.3.	Comparativa entre lenguajes de programación para el desarrollo de aplicaciones web.....	34
2.4.	Librerías para el desarrollo de la aplicación web.....	35
2.4.1.	Twconnect	35
2.4.2.	Twitter Bootstrap.....	35
2.4.3.	HightCharts	36
CAPÍTULO III.....	37	
3.1.	Metodologías para el Desarrollo de Aplicaciones Web	37
3.1.1.	Metodología Scrum.....	37
3.1.1.1.	Roles en Scrum - ¿Quién? y ¿Qué?	38
3.1.1.2.	El Sprint - ¿Dónde? ¿Cuándo?	39
3.1.1.3.	Herramientas Scrum - ¿Por qué? ¿Cómo?	41
3.1.2.	Metodología RUP	42
3.1.3.	Metodología XP.....	44
3.1.3.1.	Roles XP	45
3.1.3.2.	Valores XP.....	46
3.1.3.3.	Actividades de XP	47
3.1.3.4.	Fases de la Metodología XP	48
3.1.3.5.	Ventajas y Desventajas de la Metodología XP.....	50
3.1.4.	Comparativa de metodologías para el desarrollo de aplicaciones web.	51
e.	<i>Materiales y Métodos.....</i>	52
1.	Materiales	52
2.	Métodos.....	52
3.	Técnicas.....	52
4.	Metodología.....	53

<i>f.</i>	<i>Resultados</i>	54
✓	HIPÓTESIS.....	54
1.	PRIMERA FASE: Exploración	54
1.1.	Historias de Usuario	54
2.	SEGUNDA FASE: Planeación	63
2.1.	Cronograma de Actividades	63
2.2.	Módulos del Sistema	64
2.3.	Arquitectura de la aplicación	65
3.	TERCERA FASE: Iteraciones	66
3.1.	ITERACIÓN I	66
3.1.1.	Cronograma de Actividades	66
3.1.2.	Tareas por historia: Iteración I.....	67
3.1.3.	Pruebas Iteración I.....	90
3.1.4.	Resultados de la Iteración I	98
3.1.5.	Demo de la Versión: Iteración I	99
3.1.6.	Pruebas de Aceptación.....	105
3.1.7.	Incidencias.....	106
3.2.	ITERACIÓN II	107
3.2.1.	Cronograma de Actividades	107
3.2.2.	Tareas por historia: Iteración II.....	108
3.2.3.	Pruebas Iteración II.....	118
3.2.4.	Resultados de la Iteración II	122
3.2.5.	Demo de la Versión: Iteración II	123
3.2.6.	Pruebas de Aceptación.....	130
3.2.7.	Incidencias.....	130
3.3.	ITERACIÓN III	131
3.3.1.	Cronograma de Actividades	131
3.3.2.	Tareas por historia: Iteración III.....	132
3.3.4.	Pruebas Iteración III.....	145
3.2.5.	Resultados de la Iteración III	151
3.2.6.	Demo de la Versión: Iteración III	151
3.2.7.	Pruebas de Aceptación.....	156
3.2.8.	Incidencias	156
3.2.9.	Diagrama de Clases.....	157
3.2.10.	Modelo Entidad Relación	158
4.	CUARTA FASE: Producción	159
4.1.	Pruebas de aceptación de la aplicación	159
4.1.1.	Pruebas de Carga o stress.....	159
4.1.2.	Pruebas de Usabilidad	169
4.2.	Alojamiento de la aplicación en un servidor	169
<i>g.</i>	<i>Discusión</i>	171
1.	Desarrollo de la propuesta alternativa	171
1.1.	Objetivo específico 1: Obtener y analizar la meta data necesaria para el desarrollo de la aplicación web.....	172
1.2.	Objetivo específico 2: Determinar las herramientas necesarias para el desarrollo de la aplicación web.....	172

1.3. Objetivo específico 3: Establecer la herramienta adecuada para el almacenamiento de los datos.	172
1.4. Objetivo específico 4: Diseñar la aplicación web para el análisis estadístico y monitoreo de una cuenta de Twitter.....	173
h. <i>Conclusiones</i>	174
i. <i>Recomendaciones</i>	175
j. <i>Bibliografía</i>	176
k. <i>Anexos</i>	179
1. Anexo 1: Artículo Científico	180
2. Anexo 2: Licencia Creative Commons	181
3. Anexo 3: Anteproyecto	182
4. Autorización del desarrollo del proyecto de titulación y pruebas de campo.	183
5. Encuesta dirigida al jefe del Área de Comunicación del Servicio Integrado de Seguridad Ecu911 de la ciudad de Loja.	184
6. Certificado de culminación y aprobación de la aplicación web desarrollada.	187

Índice de Figuras

FIGURA 1. DISTINTOS DISPOSITIVOS Y REDES COMUNICÁNDOSE A TRAVÉS DE PROTOCOLOS Y LENGUAJES COMUNES [3].....	9
FIGURA 2. EJEMPLO DE TWEET.....	15
FIGURA 3. EJEMPLO PERFIL DE USUARIO EN TWITTER.....	16
FIGURA 4. EJEMPLO DE APLICACIÓN DEVELOPER EN TWITTER	18
FIGURA 5. APLICACIÓN TWITONOMY.....	21
FIGURA 6. APLICACIÓN TRENDINALIA	22
FIGURA 7. INCIDENTES ATENDIDOS POR EL ECU911 LOJA DESDE AGOSTO A DICIEMBRE DE 2014	24
FIGURA 8. INCIDENTES ATENDIDOS POR EL ECU911 LOJA DESDE ENERO A DICIEMBRE DE 2015	24
FIGURA 9. ROLES DE LA METODOLOGÍA SCRUM.....	38
FIGURA 10. SPRINT DE LA METODOLOGÍA SCRUM.....	39
FIGURA 11. FASES DE LA METODOLOGÍA XP	48
FIGURA 12. MÓDULOS DE LA APLICACIÓN	64
FIGURA 13. ARQUITECTURA DE LA APLICACIÓN.....	65
FIGURA 14. INGRESO AL SISTEMA	99
FIGURA 15. AUTENTICACIÓN CON TWITTER.....	99
FIGURA 16. AUTORIZACIÓN DE LA APLICACIÓN	100
FIGURA 17. ACTIVIDAD DE UNA CUENTA DE TWITTER	100
FIGURA 18. ESTADÍSTICA DE SEGUIDORES DE UNA CUENTA DE TWITTER	101
FIGURA 19. ESTADÍSTICA DE SIGUIENDO DE UNA CUENTA DE TWITTER	101
FIGURA 20. INFORMACIÓN DEL PERFIL DE UNA CUENTA DE TWITTER.....	102
FIGURA 21. SEGUIDORES DE UNA CUENTA DE TWITTER	102
FIGURA 22. AMIGOS DE UNA CUENTA DE TWITTER	103
FIGURA 23. TWEETS DE TU COMUNIDAD DE UNA CUENTA DE TWITTER	103
FIGURA 24. TUS TWEETS DE TWITTER	104
FIGURA 25. TWEETS DE TUS ALERTAS DE UNA CUENTA DE TWITTER.....	104
FIGURA 26. TUS TWEETS FAVORITOS DE UNA CUENTA DE TWITTER	105
FIGURA 27. ESTADÍSTICAS DE USUARIOS QUE MÁS RETUITEAN UNA CUENTA DE TWITTER.....	123
FIGURA 28. ESTADÍSTICA DE MENCIONES DE UNA CUENTA DE TWITTER	124
FIGURA 29. ESTADÍSTICAS DE TWEETS MÁS RETUITEADOS DE UNA CUENTA DE TWITTER.....	124
FIGURA 30. MENCIONAR Y/O SEGUIR UNA CUENTA DE TWITTER.....	125
FIGURA 31. ALERTAS DE UNA CUENTA DE TWITTER	126
FIGURA 32. TWEETS MÁS RETUITEADOS DE ALERTAS DE UNA CUENTA DE TWITTER	126
FIGURA 33. ESTADÍSTICA DE TWEETS DE UNA CUENTA DE TWITTER (POR DÍA).....	127
FIGURA 34. ESTADÍSTICA DE TWEETS DE UNA CUENTA DE TWITTER (POR MES).....	127
FIGURA 35. ESTADÍSTICA DE TWEETS DE UNA CUENTA DE TWITTER (POR SELECCIÓN DE RANGO DE FECHAS).....	128
FIGURA 36. ACTIVIDAD DE UNA CUENTA DE TWITTER (REDISEÑADA)	129
FIGURA 37. PUBLICACIÓN DE TWEET	129
FIGURA 38. PROGRAMACIÓN DE TWEET	130
FIGURA 39. PUBLICAR TWEETS.....	151
FIGURA 40. PROGRAMAR TWEETS	152
FIGURA 41. LISTA DE ALERTAS.....	153
FIGURA 42. CREAR NUEVA ALERTA.....	153
FIGURA 43. LOGIN: ADMINISTRACIÓN DE USUARIOS	154

FIGURA 44. LISTA DE USUARIOS CREADOS	154
FIGURA 45. LISTA DE ALERTAS CREADAS POR UN USUARIO.....	155
FIGURA 46. LISTA DE TWEETS PROGRAMADOS POR UN USUARIO	155
FIGURA 47. DIAGRAMA DE CLASES DE LA APLICACIÓN	157
FIGURA 48. MODELO ENTIDAD RELACIÓN DE LA APLICACIÓN.....	158
FIGURA 49. AÑADIR GRUPO DE HILOS EN JMETER.....	160
FIGURA 50. INGRESO DE DATOS EN JMETER	160
FIGURA 51. VALOR POR DEFECTO PARA LA CONFIGURACIÓN HTTP EN JMETER	161
FIGURA 52. INGRESO DEL NOMBRE DEL SERVIDOR EN JMETER.....	161
FIGURA 53. NUEVA PETICIÓN HTTP EN JMETER	162
FIGURA 54. NUEVA RUTA DE PETICIÓN HTTP EN JMETER	163
FIGURA 55. NUEVA ÁRBOL DE RESULTADOS EN JMETER.....	163
FIGURA 56. ÁRBOL DE RESULTADOS EN JMETER	164
FIGURA 57. NUEVA TABLA DE RESULTADOS EN JMETER	165
FIGURA 58. TABLA DE RESULTADOS EN JMETER.....	166
FIGURA 59. NUEVA GRÁFICO DE RESULTADOS EN JMETER	167
FIGURA 60. GRÁFICO DE RESULTADOS AL REALIZAR UNA PETICIÓN HTTP	168

Índice de Tablas

TABLA I. CUADRO COMPARATIVO DE LAS HERRAMIENTAS DE BASES DE DATOS	31
TABLA II. CUADRO COMPARATIVO ENTRE LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE APLICACIONES WEB	34
TABLA III. ROLES DE LA METODOLOGÍA XP	45
TABLA IV. COMPARATIVA DE METODOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB	51
TABLA V. HISTORIA DE USUARIO CONTROL Y ACCESO AL SISTEMA	55
TABLA VI. HISTORIA DE USUARIO CONTROL Y ACCESO A UNA CUENTA DE TWITTER	56
TABLA VII. HISTORIA DE USUARIO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER.....	57
TABLA VIII. HISTORIA DE USUARIO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER	58
TABLA IX. HISTORIA DE USUARIO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER....	59
TABLA X. HISTORIA DE USUARIO PUBLICAR Y PROGRAMAR TWEETS	60
TABLA XI. HISTORIA DE USUARIO ADMINISTRAR ALERTAS	61
TABLA XII. CRONOGRAMA DE ACTIVIDADES DE LA APLICACIÓN.....	63
TABLA XIII. CRONOGRAMA DE ACTIVIDADES: ITERACIÓN I.....	66
TABLA XIV. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA	67
TABLA XV. TAREA 1: CONFIGURACIÓN DEL ENTORNO DE DESARROLLO WEB DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA.....	68
TABLA XVI. TAREA 2: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA	69
TABLA XVII.TAREA 3: DISEÑO DE INTERFÁZ GRÁFICA LOGIN DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA	69
TABLA XVIII. TAREA 4: MAPEO OBJETO RELACIONAL ENTIDAD USUARIO DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA	70
TABLA XIX. TAREA 5: CREACIÓN DE ARCHIVOS PHP DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA.....	70
TABLA XX. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER.....	71
TABLA XXI. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER	72
TABLA XXII. TAREA 2: CREACIÓN DE UNA APLICACIÓN DE DESARROLLO PARA ACCEDER A LOS DATOS DE UNA CUENTA DE TWITTER DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER	73
TABLA XXIII. TAREA 3: CONFIGURACIÓN DE CREDENCIALES PARA EL ACCESO AL API DE TWITTER DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER.....	74
TABLA XXIV. TAREA 4: CREACIÓN DE FUNCIONES PHP PARA LA VERIFICACIÓN DE DATOS DE USUARIO DE UNA CUENTA DE TWITTER DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER	75
TABLA XXV. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR DATOS DEL USUARIO AUTENTICADO DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER	76
TABLA XXVI.CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER	77
TABLA XXVII. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER.....	78

TABLA XXVIII. TAREA 2: CREACIÓN DE FUNCIONES PHP PARA LA EXTRACCIÓN DE DATOS DEL PERFIL Y LA COMUNIDAD DE UN USUARIO DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER	79
TABLA XXIX. TAREA 3: DISEÑO DE INTERFACES GRÁFICAS PERFIL DE USUARIO Y GENERAL DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER	80
TABLA XXX. TAREA 4: DISEÑO DE INTERFACES GRÁFICAS SEGUIDORES Y SIGUIENDO DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER	81
TABLA XXXI. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR, EXTRAER Y MOSTRAR DATOS DEL PERFIL Y LA COMUNIDAD DE UN USUARIO DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER.....	82
TABLA XXXII. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA XXXIII. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER	¡ERROR! MARCADOR NO DEFINIDO.
TABLA XXXIV. TAREA 2: CREACIÓN DE FUNCIONES PHP PARA LA EXTRACCIÓN DE DATOS DE TWEETS FAVORITOS, DE USUARIO Y DE LA COMUNIDAD DE UN USUARIO DE TWITTER DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER	¡ERROR! MARCADOR NO DEFINIDO.
TABLA XXXV. TAREA 3: DISEÑO DE INTERFACES GRÁFICAS TUS TWEETS FAVORITOS Y TUS TWEETS DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER	¡ERROR! MARCADOR NO DEFINIDO.
TABLA XXXVI. TAREA 4: DISEÑO DE INTERFACES GRÁFICAS TWEETS DE TU COMUNIDAD Y TWEETS DE TUS ALERTAS DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER	¡ERROR! MARCADOR NO DEFINIDO.
TABLA XXXVII. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR, EXTRAER Y MOSTRAR DATOS DE TWEETS DE LA COMUNIDAD Y FAVORITOS DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER.....	¡ERROR! MARCADOR NO DEFINIDO.
TABLA XXXVIII. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN 1.0.....	90
TABLA XXXIX. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN 1.1.....	92
TABLA XL. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN 1.2	94
TABLA XLI. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN 1.3	96
TABLA XLII. CRONOGRAMA DE ACTIVIDADES ITERACIÓN II	107
TABLA XLIII. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER	108
TABLA XLIV. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER.....	109
TABLA XLV. TAREA 2: CREACIÓN DE FUNCIONES PHP PARA LA GENERACIÓN DE ESTADÍSTICAS DE UNA CUENTA DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER	110
TABLA XLVI. TAREA 3: DISEÑO DE INTERFACES GRÁFICAS ESTADÍSTICAS DE TU COMUNIDAD Y MENCIONES Y RT DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER.....	111
TABLA XLVII. TAREA 4: DISEÑO DE INTERFACES GRÁFICAS ESTADÍSTICAS DE TUS ALERTAS Y TWEETS DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER.....	112
TABLA XLVIII. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR, EXTRAER Y MOSTRAR DATOS ESTADÍSTICOS DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER.....	113
TABLA XLIX. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER	114

TABLA L. TAREA 1: REDISEÑO DE LA INTERFÁZ GRÁFICA GENERAL DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER.....	115
TABLA LI. TAREA 2: DISEÑO DE FORMULARIOS PUBLICAR Y PROGRAMAR TWEETS DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER.....	116
TABLA LII. TAREA 3: CREACIÓN DE FUNCIONES PHP PARA GUARDAR DATOS DE TWEETS PROGRAMADOS DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER	117
TABLA LIII. HISTORIAL DE REVISIONES ITERACIÓN II VERSIÓN 1.4.....	118
TABLA LIV. HISTORIAL DE REVISIONES ITERACIÓN II VERSIÓN 1.5	120
TABLA LV. CRONOGRAMA DE ACTIVIDADES ITERACIÓN III	131
TABLA LVI. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS	132
TABLA LVII. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS	133
TABLA LVIII. TAREA 2: DISEÑO DE INTERFACES GRÁFICAS PARA PUBLICAR Y PROGRAMAR TWEETS DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS	134
TABLA LIX. TAREA 3: CREACIÓN DE FUNCIONES PHP PARA CREAR, MOSTRAR, ELIMINAR Y PROGRAMAR TWEETS DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS	135
TABLA LX. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO ADMINISTRAR ALERTAS	135
TABLA LXI. TAREA 1: DISEÑO DE BASE DE DATOS DEL CASO DE USO ADMINISTRAR ALERTAS	136
TABLA LXII. TAREA 2: DISEÑO DE INTERFACES GRÁFICAS PARA ALERTAS DEL CASO DE USO ADMINISTRAR ALERTAS	137
TABLA LXIII. TAREA 3: MAPEO OBJETO RELACIONAL ENTIDAD ALERTA DEL CASO DE USO ADMINISTRAR ALERTAS	138
TABLA LXIV. TAREA 4: CREACIÓN DE FUNCIONES PHP PARA ADMINISTRAR ALERTA DEL CASO DE USO ADMINISTRAR ALERTAS	139
TABLA LXV. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO ADMINISTRAR USUARIOS	140
TABLA LXVI. TAREA 1: DISEÑO DE BASE DE DATOS DEL CASO DE USO ADMINISTRAR USUARIO.....	141
TABLA LXVII. TAREA 2: DISEÑO DE INTERFACES GRÁFICAS PARA USUARIO DEL CASO DE USO ADMINISTRAR USUARIOS	142
TABLA LXVIII. TAREA 3: MAPEO OBJETO RELACIONAL ENTIDAD USUARIO DEL CASO DE USO ADMINISTRAR USUARIOS	143
TABLA LXIX. TAREA 4: CREACIÓN DE FUNCIONES PHP PARA ADMINISTRAR USUARIO DEL CASO DE USO ADMINISTRAR USUARIOS	144
TABLA LXX. HISTORIAL DE REVISIONES ITERACIÓN III VERSIÓN 1.6	145
TABLA LXXI. HISTORIAL DE REVISIONES ITERACIÓN III VERSIÓN 1.7	147
TABLA LXXII. HISTORIAL DE REVISIONES ITERACIÓN III VERSIÓN 1.8	149

a. Título

**“DESARROLLO DE UNA APLICACIÓN WEB DE
ANÁLISIS ESTADÍSTICO Y MONITOREO DE UNA
CUENTA DE TWITTER”**

b. Resumen

Este trabajo de titulación, consiste en el desarrollo de una aplicación web que permita el análisis estadístico y monitoreo de la información de la cuenta de Twitter asociada al Sistema Integrado de Seguridad ECU911 Loja, la herramienta muestra las estadísticas de la comunidad (seguidores y siguiendo) de un perfil de Twitter, la localidad de su comunidad, estadísticas diarias de tweets, retweets, nuevos seguidores y menciones. Se incluyen el monitoreo de los tweets mediante la creación de alertas (patrones de búsqueda de tweets de acuerdo a un acontecimiento que se está publicando por Twitter), y la programación de tweets (publicación de tweets a horas específicas), con la finalidad de complementar el Plan de Seguridad Ciudadana, que como institución tiene planteado, por el imperante crecimiento de la delincuencia e inseguridad en nuestra comunidad, logrando con ello, que las emergencias que los usuarios comunican a través de esta red social, sean atendidas a tiempo, por cuanto, al momento, la falta de una herramienta que les permita controlar la información que se recoge de esta red social, es un obstáculo que impide lograr los objetivos trazados por el ECU911- Loja.

El ECU911-Loja, posee cuentas tanto en Facebook como en Twitter, sin embargo la cuenta oficial que mantiene en Twitter, se ha vuelto una herramienta muy importante al momento de obtener y brindar información, por su alto grado de aceptación y afinidad que tienen los usuarios con la misma, logrando un excelente nivel de interconexión entre la institución y la comunidad. Sin embargo, la información que se recibe de los usuarios en esta cuenta, es abundante y generalizada, lo cual provoca que un considerable número de emergencias que se denuncian en este medio, no sean atendidas a tiempo o en muchos casos no sean atendidas.

La finalidad de ésta aplicación, es contribuir con el óptimo manejo de la información que el Sistema Integrado de Seguridad ECU911 Loja, recibe de los usuarios a través de la cuenta oficial que mantiene en la red social Twitter, por cuanto, muchos de los usuarios, toman como una alternativa al servicio de llamadas 911, comunicar las emergencias a través del Twitter.

A través de ésta aplicación, la información que ingresa de la cuenta de Twitter se actualiza cada 3 minutos, misma que es almacenada en una base de datos, desde donde la aplicación la trabaja dependiendo de la necesidad del requirente, a través de una extracción y eficiente clasificación de la información más relevante, consiguiendo que las emergencias que los usuarios dan a conocer mediante esta red social, sean conocidas y atendidas a tiempo.

Para el desarrollo de ésta herramienta ha sido necesario abordar la conceptualización relacionada con aplicaciones web, la red social Twitter, la conexión a la API de Twitter, la integración de tecnologías para la parte gráfica y metodologías de aplicaciones web. Se ha utilizado la metodología XP, metodología ágil que no exige un nivel alto de documentación, que permite al programador centrarse más en el desarrollo de la aplicación y terminar en un menor tiempo.

Todas y cada una de las fases desarrolladas dentro del proyecto ha permitido obtener los resultados necesarios para cumplir con cada uno de los objetivos planteados y así tener como resultado una aplicación que permita monitorear una cuenta de Twitter, sabiendo que Twitter es una red social muy poderosa con gran acogida en todo el país y que a través de ella se puede estar al tanto de la mayoría de las noticias con mayor rapidez.

Summary

This titration work, consist in the development of a web app that allows the statistical analysis and control of a twitter account associated with de SECURITY SISTEM ECU911 LOJA, this tool shows the statistics of the twitter community (followers and people followed) by an specific profile, as well as the daily statistic of tweets, retweets and new followers. Is included the tweets monitoring by an alert (search patterns of tweets of an event that is been posted in tweeter) and the programming of tweets (posting of tweets on specific time), with the intention of make the local security program better and of course more effective because the increase in delinquency and insecurity in the community, achieving with this idea that the emergencies communicated by the users of this social network would be attended in a more effective way, because at this time the absence of a tool that helps to stock and control the information of this social network, is an impediment to achieve the objectives that the ECU911-LOJA has.

The ECU911-LOJA actually has accounts in different social networks as Facebook and tweeter, but in the latest months the official tweeter account has been such an important implement at the time of getting and also giving information, because this social network is been very accepted by the community and its users have such an incredible affinity with it. Of course this fact has made that now the interconnection between the community and the security system is excellent and for sure easier. However as the amount of information that is received in this way have grown and is also generalized, it is very difficult for the system to attend all the complaints in the precise moment, causing that a big number of emergencies are not even met.

The idea of this app is to contribute to the correct handling of the information received by the SECURITY SYSTEM ECU911-LOJA and that is send by people using the official account this company has in the social network “Tweeter” because many users prefer the social network as an option to communicate an emergency instead of using the 911 call center.

With the use of this app, the information that is entered in Tweeter is updated every 3 minutes, and it is also saved in a database where the app works depending on the user necessity through the classification of the most important information, making that the

different emergencies which the users communicate in the social network are known and met at the right time.

For the development of this tool, it has been necessary to approach concepts related to web app, tweeter social network, tweeter API connection, integration of technologies for graphics and web app methodologies. XP methodology has been used because it is a type of methodology which do not need a lot of documentation, letting the programmer focus on the development of the app, and spend less time to finish it.

Every developed step during this project have allowed to obtain the necessary results to achieve the objectives we started with, and also having as a result an app used to monitor a Tweeter account, knowing that Tweeter is a very powerful social network and nowadays it is been very used in the country, and through it, is possible to have knowledge of the most important new and have information about an specific event.

c. Introducción

En la actualidad, el uso de las redes sociales por parte de las instituciones públicas o privadas es fundamental para dar a conocer sus servicios e interactuar con el público. Esto no es una excepción para el Servicio Integrado de Seguridad Social ECU 911 Loja, ya que mediante esta red social ofrece y recibe importante información, que le permite estar más cerca de la comunidad.

En Ecuador, existen más de un millón de usuarios de Twitter a través del cual se difunden noticias con gran rapidez, es por esto, que las personas no dudan en hacer conocer algún percance, accidente, delito, etc, mediante esta red social; muchas de las veces prefieren etiquetar al ECU 911-Loja, antes que llamar a su línea única para emergencias, provocando que muchas emergencias no sean atendidas a tiempo, por la falta de control de la información que reciben en esta cuenta.

Por ello, propongo la realización del presente trabajo, cuyo objetivo principal es desarrollar una aplicación web para el análisis estadístico y monitoreo de la cuenta de Twitter asociada al Sistema Integrado de Seguridad ECU911-Loja, con la finalidad de mejorar el servicio brindado por esta entidad, ayudando a clasificar la información de su cuenta de una manera automática, ahorrando tiempo para el Ecu911 como para el ususario.

Dentro de este proyecto, para el cumplir el objetivo general se establecieron cuatro objetivos específicos, los cuales son: a) obtener y analizar la meta data necesaria para el desarrollo de la aplicación web, estos datos son obtenidos a través del API de Twitter; b) Determinar las herramientas necesarias para el desarrollo de la aplicación web; c) Establecer la herramienta adecuada para el almacenamiento de los datos; d) Diseñar la aplicación web para el análisis estadístico y monitoreo de una cuenta de Twitter, cuyo cumplimiento se dio, gracias a la utilización de herramientas como Codeigniter, Xampp, además librerías como Twconnect y Higcharts.

La Universidad Nacional de Loja y el área de la Energía, las Industrias y los Recursos Naturales no Renovables poseen lineamientos establecidos que rigen la estructura del proyecto de titulación, el cual tiene el siguiente orden descrito:

El Resumen presenta una síntesis del trabajo desarrollado, resaltando los resultados obtenidos, los Índices correspondientes al contenido, figuras, tablas y diagramas; la introducción que contiene una descripción del tema tratado junto al contenido del presente documento.

Así mismo, se encuentra inmersa la Revisión de Literatura, dónde se ha incluido toda la información bibliográfica utilizada en tres capítulos, los cuales son: Aplicaciones web, Herramientas para el desarrollo de aplicaciones web, y Metodologías para el desarrollo de aplicaciones web.

Posee una sección denominada Materiales y Métodos que describe los materiales y metodología aplicados para la obtención del presente trabajo.

En el apartado resultado, están todos los documentos obtenidos a lo largo del desarrollo del presente trabajo, de acuerdo a las fases de desarrollo de la metodología XP.

En la sección Discusión, se hace un análisis de los resultados obtenidos y su debida justificación, como una propuesta alternativa a la problemática encontrada para la culminación del presente trabajo.

Finalmente, después de haber desarrollado y culminado exitosamente lo planteado dentro del proyecto de tesis, arribamos a Conclusiones y Recomendaciones.

Para certificar todo lo desarrollado dentro del presente trabajo de Tesis, Anexo: encuesta realizada al jefe del área de comunicación de Sistema Integrado de Seguridad ECU911-Loja, solicitud para el desarrollo y realización de pruebas de campo al jefe local del centro operativo CIS ECU911-Loja, certificado de haber culminación con éxito la aplicación planteada, artículo científico de la aplicación y finalmente el anteproyecto.

d. Revisión de Literatura

Toda la información bibliográfica utilizada para el desarrollo del presente proyecto de titulación está organizada en tres capítulos: Aplicaciones web, Herramientas para el desarrollo de aplicaciones web y Metodologías para el desarrollo de aplicaciones web.

CAPÍTULO I

1.1. Aplicaciones Web

1.1.1. Páginas Web

Una página web, también conocida como página de Internet es un documento adaptado para la Web y que normalmente forma parte de un sitio web. Su principal característica son los hiperenlaces a otras páginas, siendo esto la característica fundamental de la Web.

Una página web está compuesta principalmente por información (sólo texto o multimedia) e hiperenlaces; además puede contener o asociar datos de estilo para especificar cómo debe visualizarse, o aplicaciones embebidas para hacerla interactiva.

Las páginas web son escritas en un lenguaje de marcado de hipertexto que provee la capacidad de insertar hiperenlaces, por lo general HTML. [1]

1.1.2. Sitio Web

Un sitio web es un conjunto de archivos electrónicos y páginas Web que se refieren a un tema en particular, que incluye una página inicial de bienvenida, con un nombre de dominio y dirección en Internet específicos, empleados por las instituciones públicas y privadas, organizaciones e individuos para la comunicación con todo el mundo. En un caso común, las empresas utilizan este medio con la finalidad de ofrecer sus bienes y servicios a través de Internet. [2]

1.1.3. Estándares Web

Se define como estándares web al conjunto de recomendaciones dadas por el **World Wide Web Consortium** (W3C) y otras organizaciones internacionales acerca de cómo crear e interpretar documentos basados en la Web, con el objetivo de establecer un

mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí.

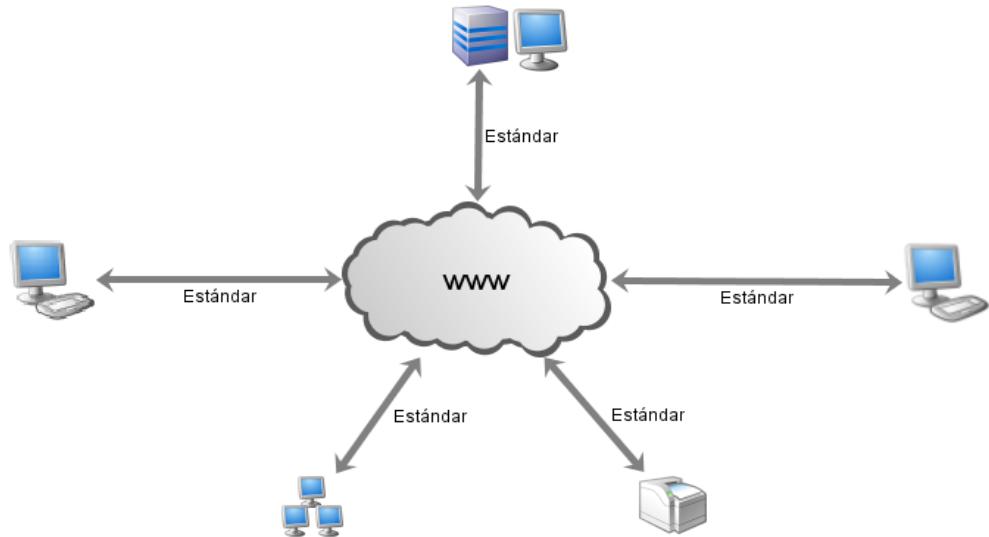


Figura 1. Distintos dispositivos y redes comunicándose a través de protocolos y lenguajes comunes [3]

La Web está basada en varios estándares, entre los más conocidos y ampliamente utilizados tenemos:

- El Identificador de Recurso Uniforme (URI), que es un sistema universal para referenciar recursos en la Web, como páginas web,
- El Protocolo de Transferencia de Hipertexto (HTTP), que especifica cómo se comunican el navegador y el servidor entre ellos.
- El Lenguaje de Marcado de Hipertexto (HTML), usado para definir la estructura y contenido de documentos de hipertexto.
- XML (eXtensible Markup Language), utilizado para estructurar la información de documentos, fue estandarizado por el W3C, que autorizó su uso libremente, convirtiéndolo en una pieza clave en la interoperabilidad de la mayoría de los sistemas de información. Muchos otros lenguajes y tecnologías están basados en XML, lo que no se habría podido conseguir si este lenguaje tuviese alguna patente comercial que restringiese su uso.

- CSS (Cascading Style Sheets), permite asignar estilos para la representación de los documentos.

1.1.4. Beneficios del uso de estándares web

Los beneficios de un sitio basado en estándares web son muy bastos, a continuación, se detallan algunos de ellos:

- Compatibilidad con todos los navegadores actuales, y lo será con versiones futuras. Funcionará tan bien en un PC, un navegador y un teléfono móvil dentro de un buen tiempo.
- Facilidad de mantener y actualizar, el código es más simple, de esta forma se elimina la dependencia de un solo desarrollador.
- Accesibilidad, permitiendo a personas con discapacidades utilizar su contenido.

1.2. Redes Sociales

Una red social es una estructura social compuesta por personas interconectadas entre sí por relaciones de diverso tipo como pueden ser amistad, parentesco o intereses comunes. Con la aparición de internet tenemos hoy en día servicios de redes sociales SNS (Social Networking Service) que trasladan estas estructuras sociales al mundo digital. Algunos ejemplos de estos SNS son Facebook, Twitter, Instagram. Una SNS permite a los usuarios crear un perfil con información básica que le identifica dentro de la red y le proporciona una serie de servicios adicionales centrados en el intercambio de información como pueden ser fotos, enlaces u opiniones. Los servicios de redes sociales se pueden clasificar en: [4]

- **Generalistas:** su propósito es permitir la comunicación entre usuarios, proporcionándoles un medio para compartir información y relacionarse. Un ejemplo de este tipo de red social es Twitter.
- **De propósito específico:** los usuarios comparten información orientada a una temática o formato concreto. Ejemplos de este tipo de red social son YouTube para compartir material multimedia.

En la actualidad el uso de los servicios de redes sociales se ha instalado y arraigado profundamente en la sociedad, sólo Facebook tiene un total de 1.110 millones de usuarios registrados y Twitter unos 500 millones de perfiles de los que 288 son usuarios activos.

1.2.1. Twitter

Twitter es una red social creada en 2006 en Estados Unidos, con la intención de facilitar el intercambio de mensajes breves entre sus usuarios. El éxito de Twitter se centra en una limitación del servicio ofrecido a los usuarios: los mensajes (tweets) que pueden publicar los usuarios tienen una extensión limitada a 140 caracteres. De esta forma se consigue un intercambio de información fluida, concisa y rápida. En un principio se creó con la idea de comunicar a pequeños grupos de personas partiendo de la idea de los antiguos SMS de los dispositivos móviles.

El crecimiento de Twitter desde que se fundó ha sido exponencial, ha pasado de tener a principios del año 2010 unos 30 millones de perfiles activos a tener 288 millones de perfiles activos en el último trimestre del año 2014. [5]

1.2.1.1. Interacciones entre usuarios y glosario de términos comunes

Las relaciones entre usuarios son de dos tipos: following y follower (seguidor y seguido). Se denomina following al conjunto de usuarios a los que un miembro sigue, suscribiéndose a sus publicaciones. Del mismo modo estos usuarios seguidos pasan a tener un nuevo follower o seguidor. La situación en la que dos usuarios se siguen mutuamente es la que se podría considerar como la relación clásica de "amistad" en otras redes sociales.

El lugar en el que aparecen los contenidos de los usuarios es el timeline o línea de tiempo. Cada miembro de esta red dispone de un espacio personal o perfil donde se muestran sus mensajes ordenados cronológicamente. Además, también se visualizan los mensajes de los usuarios que el usuario ha decidido seguir.

Las formas de interacción entre miembros de esta red son las siguientes: [6]

- **Menciones:** Es la forma en la que un usuario se dirige a otro, u otros, con el _n de iniciar una conversación, o notificarles algo en particular. Para llevar a cabo este proceso se precede el nombre de usuario con el símbolo @. Si la mención es la respuesta a otro tweet se denomina reply. Mensaje privado: A diferencia del anterior el contenido del mensaje sólo es visible para el destinatario del mensaje.
- **Hashtags:** Palabras o etiquetas que comienzan con el símbolo # y se emplean para agrupar mensajes cuyo contenido tienen un tema común. De esta forma se puede conocer cuál es la opinión de los usuarios sobre temas concretos.
- **Trending Topics:** Son las tendencias o temas de actualidad de los que los usuarios están hablando en un determinado momento. Se pueden visualizar a nivel mundial o bien restringido a zonas geográficas. Aunque la mayoría de trending topics están identificados por un hashtag, no todas las tendencias tienen este formato. La manera de generar éstos trending topics es mediante un algoritmo secreto que usa Twitter. Aunque este algoritmo es secreto, si se han dado algunas claves de cómo funciona, como por ejemplo que prevalece el número de usuarios twitteando al número de tweets o que se tiene preferencia por aquellas tendencias que son novedosas y que están ocurriendo en el momento.
- **Retweet:** Acción de copiar un tweet o mensaje de otro usuario al perfil propio, añadiéndolo al timeline o línea de tiempo.
- **Hacer Follow:** Acción de seguir un perfil de otro usuario, a partir de entonces, las actualizaciones del perfil de ese otro usuario las podrá ver el usuario que ha realizado la acción de hacer follow.

1.2.1.2. Características

Twitter es una red social no dirigida, no es necesario que exista una relación o conexión bidireccional entre dos usuarios, es decir es asimétrica, asemejándose al mundo real donde la comunicación entre entidades puede ser unidireccional, como pasa, por ejemplo, en los medios de comunicación.

En 2010 se hace un análisis de Twitter usando diversos algoritmos de teoría de grafos. Se estudia un grafo generado a partir de la recolección de datos de 41,7 millones de

perfiles de Twitter, 1,47 billones de relaciones sociales, 4.262 trending topics y 106 millones de tweets. Una vez representada la red y aplicados los distintos algoritmos de teoría de grafos, se obtienen algunas conclusiones muy interesantes sobre Twitter:

- **Reciprocidad:** Según el estudio, un 77,9% de los pares de usuarios están relacionados en una única dirección, es decir, un usuario sigue a otro, pero el otro no sigue al primero, sólo un 22,1% de las relaciones son bidireccionales, que es la relación típica de amistad en otras redes sociales como Facebook. Según los datos anteriores se puede afirmar que Twitter es una red social con bajo nivel de reciprocidad. Otro dato interesante sacado del estudio de las relaciones es que un 67,6% de los usuarios estudiados no son seguidos por ninguno de sus seguidores, con lo que los autores de dicho artículo conjeturan que, para este alto porcentaje de usuarios, Twitter es más una herramienta de información que una red social.
- **Homofilia:** Es la tendencia a relacionarse entre sí, aquellos usuarios con gustos similares. El estudio muestra que los usuarios cuya relación es recíproca están geográficamente cercanos. Esto indica que el contexto cultural y social es muy importante a la hora de establecer relaciones en Twitter.
- **La importancia del retweet.** El retweet es un mecanismo que ofrece Twitter para dotar al usuario del poder de difundir a todos sus seguidores un tweet específico. Esta característica de Twitter hace que aquella información a la que un usuario le parezca interesante se extienda de manera exponencial por la red al llegar a todos sus seguidores. La temporalidad está muy presente a la hora de hacer retweets ya que la mitad del total de retweets que llega a conseguir un tweet se produce en la primera hora desde que el tweet original se publicó, y un 75% del total se produce a lo largo de las primeras 24 horas, de lo que se deduce que el tiempo de vida de actualidad de un tweet viene a ser aproximadamente de un día.

1.2.1.3. Temporalidad de la información

Twitter ofrece un servicio de streaming, es decir, presenta la información exactamente en el momento en que se genera y de manera continua, a esta característica la llamaremos temporalidad de la información. Twitter es una red social con un alto contenido de temporalidad, es decir, la mayoría de los sucesos que los usuarios twittean en la aplicación suelen ser aquellos que están ocurriendo en tiempo real. Por ejemplo, durante

un partido de fútbol suele surgir un determinado hashtag que identifica el partido, ese hashtag o bien lo crea la propia comunidad de usuarios y prevalece aquel que es más identificativo o popular sobre el tema, o bien la propia televisión suele facilitar un hashtag para que todos los usuarios que estén viendo el partido converjan rápidamente a un único hashtag y se pongan a twittear sobre el suceso en cuestión una mayor cantidad de usuarios, para así poder llegar a ser trending topic y producir una mayor repercusión en ese instante en la comunidad de usuarios de Twitter, lo que para la televisión podría repercutir en una mayor cantidad de espectadores. Los usuarios de Twitter que están viendo el partido se suelen dividir en dos grandes grupos: los usuarios que producen contenido y los usuarios que sólo consumen contenido, es decir, aquellos usuarios que estén twitteando de manera activa durante la retransmisión del partido pertenecerán al primer grupo, y aquellos que sólo consuman información, es decir, que apenas publiquen y sólo lean lo que los usuarios productores estén publicando pertenecerán al segundo grupo. Para ambos grupos de usuarios surge el mismo problema, y es que la producción de contenido se hace de manera constante en tiempo real, y cuanto más usuarios estén twitteando sobre el partido mayor será la cantidad de información publicada por unidad de tiempo, es decir, que mientras un usuario está leyendo o publicando contenido, en ese intervalo de tiempo que ha empleado para ello, se han podido llegar a publicar varias decenas de tweets. Esto produce que alguien que se acaba de conectar a Twitter y quiera informarse de los acontecimientos que han sucedido hasta ese momento en el partido pueda encontrarse con que no es capaz de encontrar lo que está buscando debido a la inmensa cantidad de tweets publicados, muchos de ellos opiniones personales o ruido que se cuela en medio de la tendencia. [7]

1.2.1.4. Información disponible

Además de la información básica que provee Twitter (el tweet). Twitter ofrece también la siguiente información derivada de los tweets, de los usuarios de la aplicación y del entorno de la aplicación en general:

Los elementos básicos que conforman un tweet son: cuerpo, creador, fecha y hora en las que se creó el tweet, número de retweets (cantidad de veces que dicho mensaje ha sido copiado en el perfil de otro usuario) y número de favoritos (cantidad de veces que dicho tweet ha sido seleccionado como favorito por los demás usuarios). Ver figura 2



Figura 2. Ejemplo de tweet

En la figura 3 se muestra el conjunto de características que nos permite conocer Twitter sobre un usuario de su aplicación:

- Foto de perfil con la que el usuario se identifica.
- Nombre de usuario, el cual puede no ser único y el usuario puede cambiar a su gusto.
- Identificador de usuario para identificar de forma única al usuario, es la información que se usa para realizar las menciones.
- Tweets que el usuario ha publicado, creado o retwitteado. Ésta lista se puede filtrar de manera que sólo muestre los tweets publicados, las fotos y vídeos o todos en conjunto.
- Lista de favoritos.
- Seguidores.
- Siguiendo (conjunto de otros perfiles a los que el usuario ha decidido seguir).



Figura 3. Ejemplo perfil de usuario en Twitter

Twitter ofrece en su interfaz la lista de los trending topics, los cuales se ordenan de manera jerárquica dependiendo de lo relevantes que sean para los usuarios, es decir, aquellos que más tweets generan se colocarán en las primeras posiciones. Esta lista de trending topics se puede filtrar a nivel global, nacional o metropolitano.

Toda esta información accesible desde Twitter permite crear diversas aplicaciones y servicios relacionados con la red social.

1.2.2. Herramientas para la Gestión de una Cuenta de Twitter

1.2.2.1. Autenticación OAuth

Open Authorization es un protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite la autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web.

Fue lanzado hacia finales de 2007 y define un mecanismo para que una aplicación web (cliente) pueda acceder a la información de un usuario en otra (proveedor) sin tener que informar a la primera del usuario y contraseña. [8]

1.2.2.2. API de Twitter

Uno de los motivos de la expansión de Twitter ha sido la existencia de APIs gratuitas que proporciona la empresa, que han propiciado la creación de software de terceros que permite conectarse y manejar datos de la aplicación.

El API de Twitter está limitado ya que el acceso a la aplicación está limitado a 150 o 350 solicitudes por hora dependiendo si registramos o no la aplicación en el apartado de desarrolladores de Twitter. Twitter usa OAuth para tener acceso a algunas APIs, OAuth es un protocolo abierto para permitir acceso seguro de manera simple y estándar. Hay tres APIs principales proporcionadas por Twitter: [9]

- **Search API:** Se encarga de suministrar los tweets buscados de hasta hace 7 días, con un máximo de 1.500 tweets. En esta API es posible filtrar por cliente utilizado, lenguaje y localización. [10]
- **Rest API:** Es una API web que funciona por HTTP a la cual accedemos a partir de URLs que devuelven contenido en formato JSON, XML, HTML, etc. A diferencia de la Search API no hay limitación temporal, pero sí una limitación del número de resultados devueltos establecido en 3.200 tweets. [11]
- **Streaming API:** Permite recibir información en tiempo real. Los contenidos devueltos tienen formato JSON. En esta API se pueden obtener muestras aleatorias o un filtrado por palabras clave o usuarios, aunque también existen métodos más interesantes como puedan ser el poder obtener el caudal de tweets, o filtrar solo por tweets con enlaces o tweets con retweets. [12]

A través de las API de Twitter cualquiera puede crear aplicaciones que comuniquen con los servidores de Twitter desde un script PHP, para recibir el estado de un usuario, sus seguidos o postear un mensaje automático a Twitter, entre otras.

1.2.2.3. Limitaciones de los APIs de Twitter

En el Streaming API el flujo es continuo y la velocidad de recepción de tweets tendrá fluctuaciones que dependerán del ancho de banda de los dos extremos de la conexión y la sobrecarga de los servidores de Twitter. En el Search API y en el REST API existe una limitación de 150 peticiones a la hora por usuario o por IP si la llamada no está autenticada.

1.2.2.4. Rest API de Twitter

La API REST o API pública es la utilizada en el proyecto para la extracción de información de Twitter. Ésta proporciona acceso de lectura sobre los datos públicos en Twitter, y por lo tanto no requiere disponer datos de ingreso de las cuentas a las que se pretende acceder, por lo que se pueden extraer datos como los tweets de un usuario, información de perfiles usuario y datos de los seguidores entre otras opciones de una forma muy sencilla y para cualquier cuenta.

Para configurar la API, primero hay que registrarse en Twitter y crear una App en Twitter Apps, luego generar las credenciales o el OAuth con el cual la API identifica las peticiones y así crear las respuestas JSON.

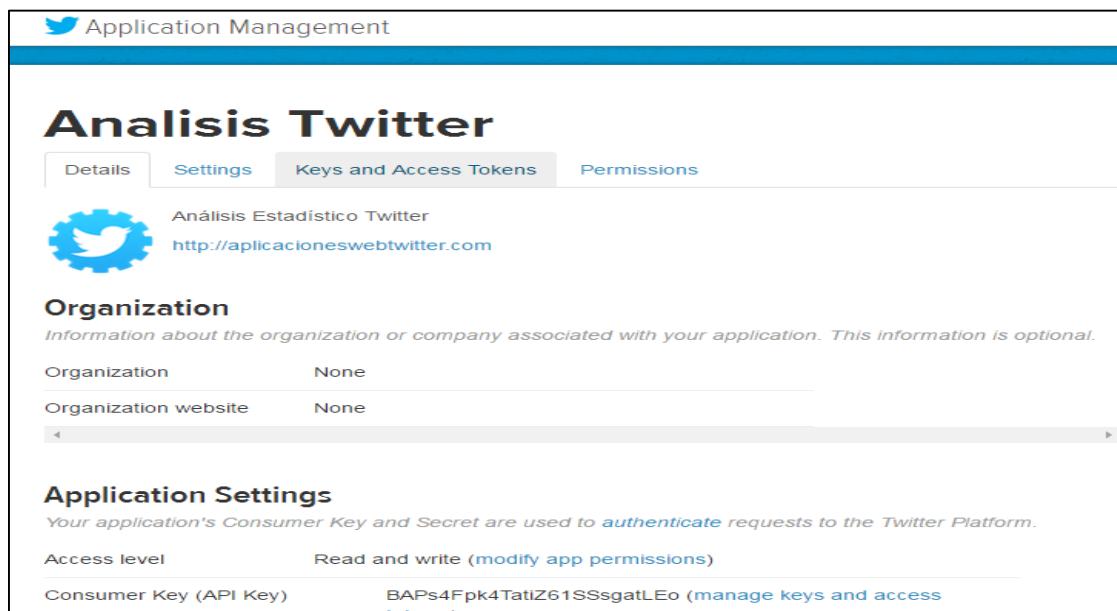


Figura 4. Ejemplo de Aplicación Developer en Twitter

Algunas de las peticiones o métodos más comunes que se pueden realizar mediante la API son *GET followers/ids*, *GET friends/ids*, *GET followers/list*, *GET users/show*, *GET users/lookup* o *GET statuses/user_timeline*, aunque hay muchos más. En el uso de las diferentes peticiones existentes hay que tener en cuenta algunos factores que limitan las posibilidades de estas. Los dos factores más importantes son:

- Para aquellos métodos que devuelven una línea de tiempo de tweets, hay que tener en cuenta que la paginación de la información puede generar problemas debido a la naturaleza de tiempo real de los propios tweets. Por lo que lo más común será utilizar el parámetro `max_id` a partir del cual se cargarán los tweets, evitando así duplicidades en caso de haber actualizaciones entre petición y petición.
- Otro factor a tener en cuenta es el límite en la tasa de peticiones. La tasa límite de la API se define en intervalos (o ventanas) de 15 minutos en las que según el método utilizado se pueden realizar un número determinado de peticiones u otro.

Algunas de las funciones que ofrece el API Rest de Twitter son:

- **GET statuses/mentions_timeline:** Devuelve las 20 menciones más recientes (tweets que contengan `@screen_name` de un usuario) para el usuario autenticado. Este método sólo puede devolver hasta 800 tweets.
- **GET statuses/user_timeline:** Devuelve una colección de los más recientes tweets escritos por el usuario indicado por los parámetros `screen_name` o `user_id`. La línea de tiempo devuelto es el equivalente a la que se ve cuando se ve el perfil de un usuario en Twitter. Este método sólo se puede devolver hasta 3.200 Tweets más recientes de un usuario. Retweets nativos de otros estados por el usuario que se incluyen en este total.
- **GET statuses/home_timeline:** Devuelve una colección de los más recientes tweets y retweets publicados por el usuario y los usuarios que siguen. Hasta 800 tweets se pueden obtener en esta línea de tiempo. Es más volátil para los usuarios que siguen muchos usuarios o siguen a los usuarios que twittean con frecuencia.
- **GET statuses/retweets_of_me:** Devuelve los tweets más recientes publicados por el usuario autenticado que se han retweeted por otros. Esta linea de tiempo es un subconjunto del GET estados / user_timeline.
- **GET statuses/retweets/: id:** Devuelve una colección de los 100 retweets más recientes del tweet especificado por el id de parámetro.

- **GET friends/list:** Devuelve una colección de objetos de usuario también conocido como los seguidores de un usuario.
- **GET followers/list:** Devuelve una colección de objetos de usuario (usuarios a los que estas siguiendo).
- **GET account/verify_credentials:** Devuelve un código de respuesta 200 OK HTTP y una representación del usuario que solicita si la autenticación se ha realizado correctamente; devuelve un código de estado 401 y un mensaje de error en caso contrario. Este método se utiliza para probar si las credenciales de usuario suministrados son válidas.
- **GET favorites/list:** Devuelve los 20 tweets más recientes del agrado del usuario especificado.
- **POST statuses/update:** Actualiza el estado actual del usuario autenticado, también conocido como tweet. Para cada intento de actualización el texto se compara con los últimos tweets de los usuarios de autenticación. Se bloqueará cualquier intento que daría lugar a la duplicación, lo que resulta en un error 403. Por lo tanto, un usuario no puede enviar el mismo estado dos veces en una fila.

1.2.3. Herramientas de análisis de datos de Twitter

1.2.4. Twitonomy

Twitonomy es una aplicación web que te permite analizar una cuenta de Twitter, a través de diferentes métodos. Este servicio permite generar analíticas visuales y detalladas de tweets, retweets, respuestas, menciones y hashtags. También se puede buscar y filtrar data de las personas que seguimos y de las que nos siguen. Otra de las características importantes de este servicio es que se puede monitorear las interacciones entre amigos, desde menciones, hasta favoritos, muestra la lista de followers a los cuales no se sigue y también permite gestionar las listas de Twitter. [13]

Dos características interesantes que ofrece Twitonomy son la exportación y las utilidades gráficas. En cuanto a la primera se puede exportar a un Excel los tweets que hemos realizado y en cuanto a los gráficos, podemos obtenerlos de distintos tipos tanto con

información de nuestra actividad como de los usuarios que más interactúan con nuestro contenido.

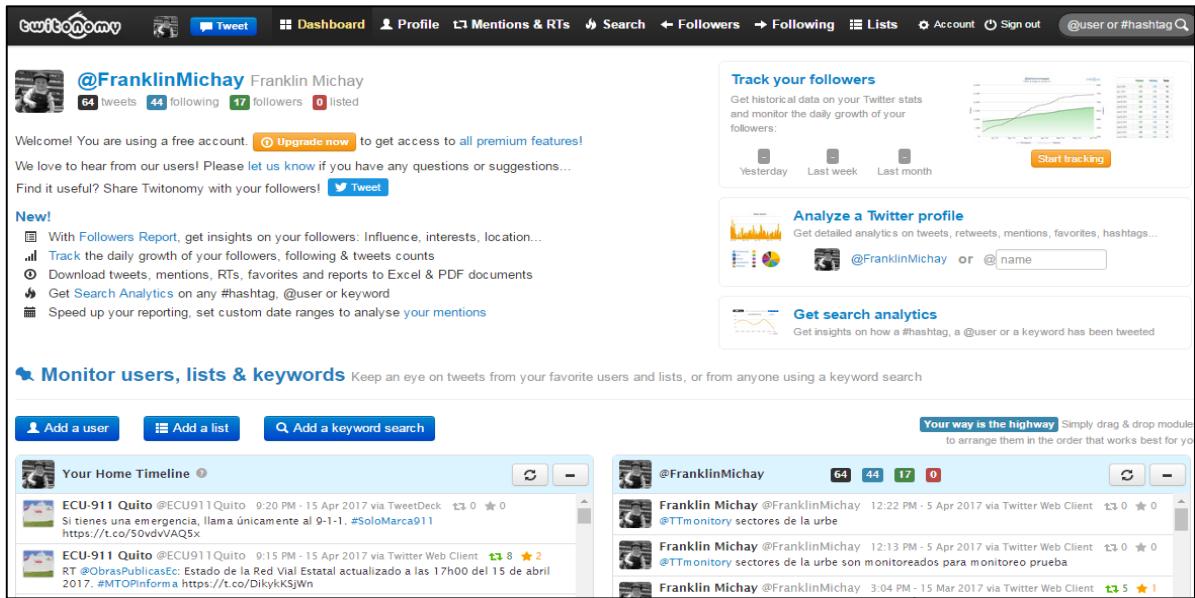


Figura 5. Aplicación Twitonomy

1.2.5. Trendinalia

Es un servicio web que proporciona una monitorización de los diferentes hashtags que se convierten en trending topic en algún momento del día. Dicha aplicación proporciona información diversa sobre el trending topic en cuestión, su duración, ubicación y gráficas para comparar unas tendencias con otras. En esta aplicación se puede filtrar por día y lugar ofreciendo una visión más local de la actualidad. Esta aplicación mantiene un registro sobre los trending topics a lo largo del tiempo. [14]

En la figura 5 se presenta una captura de la aplicación donde se puede ver la lista de trending topics y la duración de los mismos como tendencia en Twitter. A la izquierda de la imagen se puede ver que la aplicación permite filtrar resultados por país, fecha e incluso la ciudad a la que pertenecen. A la derecha de la misma se puede apreciar un conjunto de gráficas que hacen más explicativos los resultados mostrados.

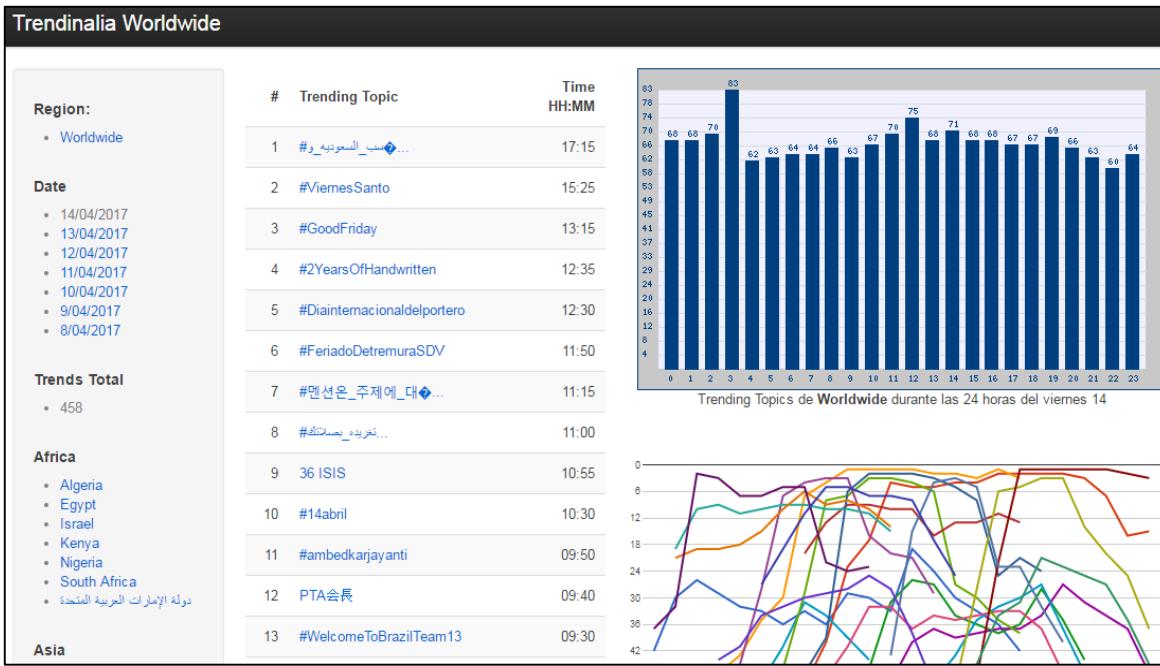


Figura 6. Aplicación Trendinalia

1.3. Sistema Integrado de Seguridad ECU911 Loja

Es un servicio de respuesta inmediata e integral a una determinada emergencia. Coordina la atención de los organismos de respuesta articulados en la institución para casos de accidentes, desastres y emergencias movilizando recursos disponibles para brindar atención rápida a la ciudadanía.

Policía Nacional, Fuerzas Armadas, Cuerpo de Bomberos, Comisión Nacional de Tránsito, Ministerio de Salud Pública, Instituto Ecuatoriano de Seguridad Social, Secretaría de Gestión de Riesgos, Cruz Roja Ecuatoriana y otros organismos locales encargados de la atención de emergencias, han unido esfuerzos para brindar la mejor atención a través de un número único: 911

El ECU 911, a través de una moderna plataforma tecnológica y con base a políticas, normativas y procesos, articula sus servicios de video vigilancia, botones de auxilio, alarmas comunitarias, recepción y despachos de atención a emergencias a través de

llamadas con la coordinación de instituciones públicas, mediante dependencias o entes a su cargo que dan respuestas a la ciudadanía en situaciones de emergencia.

Así se reemplaza a todos los números de emergencia que fragmentaban la atención en distintas instituciones de respuesta, logrando la cooperación articulada para soluciones integrales.

La llamada al ECU 911 puede realizarse desde cualquier teléfono fijo o móvil, sin costo alguno, las 24 horas del día, los 365 días del año. [15]

1.3.1. Objetivos estratégicos

- Incrementar la eficiencia y eficacia en la calidad de servicios de emergencia enfocados a la ciudadanía.
- Incrementar las capacidades institucionales en la prestación de servicios de emergencia en beneficio de la ciudadanía.
- Incrementar la eficiencia operacional del ECU 911.
- Incrementar el desarrollo del talento humano del ECU 911.
- Incrementar el uso eficiente del presupuesto del ECU 911.

1.3.2. Entidades coordinadas

Las entidades articuladas al Servicio Integrado de Seguridad ECU 911 son: [16]

- Cruz Roja Ecuatoriana. [17]
- Secretaría Nacional de Riesgos. [18]
- Ministerio de Salud Pública. [19]
- Instituto Ecuatoriano de Seguridad Social. [20]
- Policía Nacional. [21]
- Cuerpos de Bomberos Ecuador.
- Ministerio de Defensa. [22]

1.3.3. Estadísticas

Las estadísticas son una de las principales formas de medir la eficacia y funcionalidad del ECU 911. Esta herramienta ayuda a conocer la real magnitud de un incidente y enriquecer la toma de decisiones, evaluar el sistema y mejorar acciones para eventos futuros.

CENTRO	AÑO	MES	# INCIDENTE SEGURIDAD CIUDADANA (TOTAL)	# INCIDENTE GETIÓN SANITARIA (TOTAL)	# INCIDENTE TRÁNSITO Y MOVILIDAD (TOTAL)	# INCIDENTE SERVICIOS MUNICIPALES (TOTAL)
LOJA	2014	AGOSTO	4.831	1.289	656	156
LOJA	2014	SEPTIEMBRE	4.495	1.269	575	160
LOJA	2014	OCTUBRE	4.626	1.187	547	216
LOJA	2014	NOVIEMBRE	4.834	1.302	562	191
LOJA	2014	DICIEMBRE	5.069	1.511	608	227
TOTAL			23.855	6.558	2.948	950

Figura 7. Incidentes atendidos por el ECU911 Loja desde agosto a diciembre de 2014

CENTRO	AÑO	MES	# INCIDENTE SEGURIDAD CIUDADANA (TOTAL)	# INCIDENTE GETIÓN SANITARIA (TOTAL)	# INCIDENTE TRÁNSITO Y MOVILIDAD (TOTAL)	# INCIDENTE SERVICIOS MUNICIPALES (TOTAL)
LOJA	2015	ENERO	4.049	1.613	535	211
LOJA	2015	FEBRERO	3.859	1.502	640	230
LOJA	2015	MARZO	4.005	1.563	738	472
LOJA	2015	ABRIL	3.934	1.448	697	344
LOJA	2015	MAYO	4.944	1.526	983	307
LOJA	2015	JUNIO	3.827	1.447	979	430
LOJA	2015	JULIO	4.100	1.500	915	358
LOJA	2015	AGOSTO	4.306	1.637	960	363
LOJA	2015	SEPTIEMBRE	4.111	1.502	888	260
LOJA	2015	OCTUBRE	4.797	1.559	913	266
LOJA	2015	NOVIEMBRE	4.673	1.566	924	213
LOJA	2015	DICIEMBRE	4.867	1.674	994	205
TOTAL			51.472	18.537	10.166	3.659

Figura 8. Incidentes atendidos por el ECU911 Loja desde enero a diciembre de 2015

1.3.4. Servicios que presta ECU 911 para una Seguridad Ciudadana Integral

- **Video vigilancia**

El servicio de video vigilancia de ECU 911 utiliza lo más avanzado de la tecnología para monitorear las actividades que puedan generar situaciones de riesgo.

- **Línea única de emergencias**

Gracias a la coordinación inter-institucional del Servicio Integrado ECU 911, usted ya no necesita acudir a varios números telefónicos. En una emergencia solo marque 911.

- **Vinculación comunitaria**

Charlas y capacitaciones a niños, jóvenes y comunidades enfocadas en el buen uso del servicio ECU 911 y la importancia de la colaboración ciudadana en la seguridad integral

- **Coordinación institucional**

EL trabajo en conjunto de todas las instituciones de respuesta a emergencias permite que una misma situación sea atendida desde una perspectiva completa permitiendo una respuesta eficaz e integral. Reduce los tiempos de respuesta y logra movilizar unidades especializadas para urgencias puntuales.

Con el objetivo de incrementar la efectividad de la labor que realiza el SIS ECU 911, ponemos a disposición de la ciudadanía una variedad de servicios que ofrecen una asistencia única y permanente:

- Consultas de líneas telefónicas suspendidas. [23]
- Aplicación para celulares. [24]
- Localizador Móvil. [25]
- Estadio Seguro. [26]
- Feriado Seguro. [27]

1.3.5. Geolocalización

La geolocalización o denominada también georreferenciación es una solución tecnológica que permite determinar la ubicación o posicionamiento geográfico de un dispositivo.

La plataforma tecnológica de atención de emergencias del Servicio Integrado de Seguridad ECU 911 permite visualizar de manera automática en el sistema de información geográfica (GIS) el posicionamiento aproximado de la llamada realizada por una persona a la línea única de emergencias 9-1-1 desde un teléfono móvil o fijo a nivel nacional.

La entrega de la información de localización aproximada relacionado a geolocalización de un terminal por parte de las operadoras de telefonía, así como, los niveles de precisión de dicha información, está regulada por la Agencia de Regulación y Control de las Telecomunicaciones “ARCOTEL”.

Las prestadoras de la telefonía realizan el cálculo de la localización aproximada del teléfono móvil o fijo de acuerdo a las siguientes metodologías:

- Se toman en cuenta la cantidad o densidad de radio bases celulares existentes en el área geográfica donde se encuentra el teléfono móvil. Mientras más radio bases o antenas existan, mejor será la precisión de la información de ubicación del teléfono. Aquí el teléfono móvil intercambia parámetros con la radio base para poder calcular su ubicación.
- Se realiza el posicionamiento del teléfono móvil en base a la información del sistema de posicionamiento global (GPS) interno, dicha funcionalidad se encuentra activa de manera predeterminada en la mayoría de teléfonos inteligentes. Con este método, existe mayor precisión en la información de localización.
- La información de direcciones y coordenadas provistas por las operadoras de telefonía fija que permiten posicionar la información del abonado que realiza la llamada. [28]

1.3.6. Aplicación Smartphone ECU911

Uno de los objetivos del ECU 911 es trabajar continuamente en la innovación tecnológica para responder efectivamente en casos de emergencias, por ello ofrece a la ciudadanía la aplicación smartphone ECU 911.

La herramienta que fue creada por ecuatorianos puede ser utilizada incluso por personas con alguna discapacidad auditiva o física

Los usuarios de teléfonos inteligentes, con sistemas operativos Android, iOS y Blackberry pueden descargarla de forma para solicitar asistencia en incidentes o emergencias.

La app ECU 911 es gratuita y para descargarla el usuario debe registrar la información que le pide el sistema de manera veraz, pues esta servirá para ser atendido de acuerdo a

su emergencia. Cabe señalar que la información proporcionada es totalmente confidencial. [29]

1.3.7. Feriado seguro

El Servicio Integrado de Seguridad ECU 911 en coordinación con las entidades articuladas, realiza continuamente planes de contingencia para ofrecer a la ciudadanía información durante los días feriados. El objetivo de estas campañas consiste en brindar protección a los viajeros poniendo énfasis en la prevención y atención de accidentes de tránsito, información sobre el estado vial, consejos de seguridad a fin de que turistas nacionales y extranjeros disfruten de los días festivos sin riesgos. [30]

1.3.8. Estadio Seguro:

Con el fin de garantizar la integridad física de los aficionados y deportistas, el Ministerio Coordinador de Seguridad junto al ECU 911, implementaron el proyecto “Estadio Seguro”, que consiste en ubicar cámaras de video vigilancia en los diferentes escenarios deportivos para que la ciudadanía pueda disfrutar de manera segura los partidos de fútbol.

La prioridad es la prevención de la violencia mediante el monitoreo constante del desenvolvimiento y comportamiento de los asistentes dentro y fuera de los diferentes estadios a escala nacional. [31]

CAPÍTULO II

2.1. Herramientas para el desarrollo de Aplicaciones Web

2.1.1. Herramientas para el almacenamiento de datos

2.1.1.1. Mysql

MySQL, es el servidor de bases de datos relacionales más comúnmente utilizado, fue desarrollado por la empresa MySQL AB, que cedió las licencias correspondientes al proyecto OpenSource, por lo que su rápido desarrollo es causa del empeño de millones de programadores de todo el mundo.

Al ser un servidor de bases de datos relacionales, MySQL se convierte en una herramienta veloz en la accesibilidad a los datos introducidos en las distintas tablas independientes que forman las bases de datos de este lenguaje. MySQL es actualmente el sistema de bases de datos más popular de la red. [32]

- **Características de MySQL**

- ✓ Un amplio subconjunto de ANSI SQL 99, y varias extensiones.
- ✓ Soporte a multiplataforma
- ✓ Procedimientos almacenados
- ✓ Disparadores (*triggers*)
- ✓ Cursores
- ✓ Vistas actualizables
- ✓ Soporte a VARCHAR
- ✓ INFORMATION_SCHEMA

- **Ventajas**

- ✓ Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- ✓ Mejores utilidades de administración (backup, recuperación de errores, etc).
- ✓ Aunque se cuelgue, no suele perder información ni corromper los datos.
- ✓ Mejor integración con PHP.
- ✓ No hay límites en el tamaño de los registros.
- ✓ Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- ✓ MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla.

- **Inconvenientes**

- ✓ No soporta transacciones, "roll-backs" ni subselects.
- ✓ No considera las claves ajenas. Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

2.1.1.2. PostgreSQL

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad.

Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos del SQL 2008, incluyendo INTEGER, numérico, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C / C + +, Java, Net, Perl, Python, Ruby, Tcl, ODBC, entre otros. [33]

▪ Ventajas

- ✓ Por su arquitectura de diseño, escala muy bien al aumentar el número de CPUs y la cantidad de RAM.
- ✓ Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial).
- ✓ Tiene mejor soporte para triggers y procedimientos en el servidor.
- ✓ Soporta un subconjunto de SQL92 MAYOR que el que soporta MySQL. [34]

▪ Inconvenientes

- ✓ Consume bastante más recursos y carga más el sistema.
- ✓ Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- ✓ Menos funciones en PHP.

2.1.2. Comparativa entre herramientas de Bases de Datos para el desarrollo de Aplicaciones Web

De acuerdo a las características de cada una de las herramientas descritas anteriormente se realizó un cuadro comparativo descrito en la tabla 1.

TABLA I. CUADRO COMPARATIVO DE LAS HERRAMIENTAS DE BASES DE DATOS

HERRAMIENTAS	RECURSOS	RENDIMIENTO	LÍMITE DE REGISTROS	INTEGRACIÓN CON PHP
MYSQL	Mejores utilidades de administración (backup, recuperación de errores, etc).	Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.	No hay límites en el tamaño de los registros	Mejor integración con PHP.
POSTGRESQL	Consumo bastantes más recursos y carga más el sistema.	Es de 2 a 3 veces más lenta que MySQL.	Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k, pero con un coste añadido en el rendimiento).	Menos funciones en PHP.

En base a la comparativa realizada se ha seleccionado MYSQL como el motor de base de datos indicado para el almacenamiento de los datos de la aplicación ya que muestra algunas ventajas como es el uso de pocos recursos del sistema con relación a POSTGRESQL, nos proporciona un mayor rendimiento y velocidad al realizar consultas a la base de datos, proporciona muy buena integración con PHP punto importante puesto que es el lenguaje en el que vamos a desarrollar nuestra aplicación.

2.2. Lenguaje de programación

2.2.1. Java

Java es un lenguaje de programación desarrollado por Sun Microsystems, fue presentado en la segunda mitad del año 1995. Java es un lenguaje muy valorado porque los programas Java se pueden ejecutar en diversas plataformas con sistemas operativos como Windows, Mac OS, Linux o Solaris. James Gosling, el director del equipo de trabajo encargado de desarrollar Java, hizo realidad la promesa de un lenguaje independiente de la plataforma. Se buscaba diseñar un lenguaje que permitiera programar una aplicación una sola vez que luego pudiera ejecutarse en distintas máquinas y sistemas operativos. Para conseguir la portabilidad de los programas Java se utiliza un entorno de ejecución para los programas compilados. Este entorno se denomina Java Runtime Environment (JRE). Es gratuito y está disponible para los principales sistemas operativos. Esto asegura que el mismo programa Java pueda ejecutarse en Windows, Mac OS, Linux o Solaris.

Java es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana.

- Es un lenguaje que es compilado, generando ficheros de clases compilados, pero estas clases compiladas son en realidad interpretadas por la máquina virtual java. Siendo la máquina virtual de java la que mantiene el control sobre las clases que se estén ejecutando.
- Es un lenguaje multiplataforma: El mismo código java que funciona en un sistema operativo funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual java.
- Es un lenguaje seguro: La máquina virtual al ejecutar el código java realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.
- Gracias al API de java podemos ampliar el lenguaje para que sea capaz de, por ejemplo, comunicarse con equipos mediante red, acceder a bases de datos, crear páginas HTML dinámicas, crear aplicaciones visuales al estilo Windows.

Para poder trabajar con java es necesario emplear un software que permita desarrollar en java, existen varias alternativas comerciales en el mercado: JBuilder, Visual Age, Visual Café, y un conjunto de herramientas shareware e incluso freeware que permiten trabajar

con java. Pero todas estas herramientas en realidad se basan en el uso de una herramienta proporcionada por Sun, el creador de java, que es el Java Development Kit (JDK).

2.2.2. PHP

Originalmente diseñado por el programador danés-canadiense Rasmus Lerdorf, en el año 1994 en base a la escritura de un grupo de CGI binarios escritos en el lenguaje C.

PHP es un lenguaje gratuito y multiplataforma, brinda la posibilidad de acceso a muchos tipos de bases de datos, también se puede crear páginas dinámicas, así como la posibilidad de separar el diseño del contenido de una web.

PHP comenzó siendo unas simples funciones que permitían crear páginas web rápidamente (tratar los formularios) y sin la necesidad de realizar código en C/C++, que poco a poco ha venido evolucionando en un lenguaje completo, orientado totalmente a la web. Para realizar una página web, PHP es mucho más fácil de aprender que Java, dado que este último, es orientado totalmente a objeto, por lo que lo hace menos flexible que PHP. [35]

- **Características**

- ✓ **Velocidad:** No requiere grandes recursos del sistema, se integra muy bien junto a otras aplicaciones, especialmente bajo ambientes Unix.
- ✓ **Estabilidad:** Utiliza su propio sistema de administración de recursos y posee de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- ✓ **Seguridad:** Maneja distintos niveles de seguridad, estos pueden ser configurados desde el archivo.ini
- ✓ **Simplicidad:** Dispone de una amplia gama de librerías, y permite la posibilidad de agregarle extensiones.

- **Ventajas**

- ✓ PHP corre en (casi) cualquier plataforma utilizando el mismo código fuente,
- ✓ PHP es completamente expandible y modificable. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.
- ✓ Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen yTHTTPD. Otra alternativa es configurarlo como módulo CGI.
- ✓ Permite la interacción con gran cantidad de motores de bases de datos tales como MySQL, MS SQL, Oracle, Informix, PostgreSQL, etc.
- ✓ PHP es Open Source, no requiere de licencias. [36]

2.3. Comparativa entre lenguajes de programación para el desarrollo de aplicaciones web.

TABLA II. CUADRO COMPARATIVO ENTRE LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE APLICACIONES WEB

METODOLOGÍA	CARACTERÍSTICAS	LIBRERÍAS DISPONIBLES	HOSPEDAJE EN EL SERVIDOR
JAVA	<ul style="list-style-type: none"> ✓ Pensado para aplicaciones web para uso particular. ✓ Java es tipado (hay que declarar tipos de variables) ✓ 100% orientado a objetos ✓ Rendimiento muy pesado. 	<ul style="list-style-type: none"> ✓ Librerías no orientadas a aplicaciones web 	<ul style="list-style-type: none"> ✓ No se encuentran servidores Java que ofrezcan JSP/Servlets gratuitos
PHP	<ul style="list-style-type: none"> ✓ Pensado para aplicaciones web para la gente en general. ✓ No hay tipado ✓ Permite mezclar clases con funciones de programación estructurada. ✓ Rendimiento con mayor rapidez. 	<ul style="list-style-type: none"> ✓ Librerías orientadas a aplicaciones web 	<ul style="list-style-type: none"> ✓ es más fácil encontrar servidor PHP con base de datos gratuitos

Luego de haber analizado los lineamientos anteriores se consideró al lenguaje de programación PHP como la mejor opción para el desarrollo de la aplicación puesto que se trata de una aplicación de menor tamaño con un tiempo de desarrollo corto, así mismo hay un sinnúmero de librerías desarrolladas en este lenguaje, muchas de estas permiten conectarnos con el API de Twitter lo cual es fundamental para cumplir con nuestro objetivo.

2.4. Librerías para el desarrollo de la aplicación web

2.4.1. Twconnect

Esta biblioteca encapsula TwitterOAuth biblioteca desarrollada por Abraham Williams para el acceso a los métodos del API de Twitter. Esta biblioteca simplifica el registro de Twitter y el acceso envolviendo métodos TwitterOAuth en su propia clase.

2.4.2. Twitter Bootstrap

Es una colección de herramientas de software libre para la creación de sitios y aplicaciones web dinámicas y con altos estándares de calidad. Contiene plantillas de diseño basadas en HTML5 y CSS 3 con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript. La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (responsive design) apoyándonos en un framework potente con numerosos componentes webs que permitirán ahorrar mucho tiempo y esfuerzo. [37]

▪ Características

- ✓ Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- ✓ Se integra perfectamente con las principales librerías JAVASCRIPT.
- ✓ Ofrece un diseño sólido usando estándares como CSS3/HTML5.
- ✓ Funciona con todos los navegadores, incluido Internet Explorer usando HTML para que reconozca los tags HTML5
- ✓ Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.
- ✓ Convierte los componentes de una web en full responsive.

2.4.3. HighCharts

HighCharts es una librería escrita en Javascript que permite la creación de gráficas. La librería ofrece un método fácil e interactivo para insertar graficas en su sitio web o aplicación web.

- **Características**

- ✓ La librería es compatible con todos los navegadores modernos incluyendo iPhone/iPad e Internet Explorer desde su versión 6.
- ✓ No es comercial, no se necesita el permiso de los autores para su implementación en sitios web personales o sin fines de lucro.
- ✓ Es abierto, todas las características pueden ser personalizadas permitiendo una gran flexibilidad además HighCharts está escrito solamente con código Javascript, sólo se requiere incluir el archivo highcharts.js y cualquiera de los tres frameworks más populares de Javascript (jQuery, MooTools o Prototype). [38]

CAPÍTULO III

3.1. Metodologías para el Desarrollo de Aplicaciones Web

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros.

A continuación, se presenta resumidamente el contexto en el que surgen las metodologías ágiles, sus valores, principios y comparación entre ellas:

3.1.1. Metodología Scrum

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Scrum es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es:

- Ligero
- Fácil de entender
- Extremadamente difícil de llegar a dominar

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar.

Scrum es un framework para trabajar en equipo en una serie de interacciones. Las fases en las que se divide y define un proceso de SCRUM son las siguientes:

- El **¿Quién? y el ¿Qué?**: identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.
- El **¿Dónde? y el ¿Cuándo?**: que representan el Sprint
- El **¿Por qué? y el ¿Cómo?**: representan las herramientas que utilizan los miembros de Scrum

3.1.1.1. Roles en Scrum - ¿Quién? y ¿Qué?

El equipo de Scrum consiste en tres diferentes roles: [39]

- El Product Owner/Dueño del producto es la “voz del cliente” y el responsable de desarrollar, mantener y priorizar las tareas en el backlog.
- El Scrum Master es responsable de asegurarse que el trabajo del equipo vaya bien siguiendo las bases de Scrum. Además, se encarga de remover cualquier obstáculo que pueda encontrar el equipo de desarrollo.
- Los Development Team Members/Miembros del Equipo de desarrollo son los encargados de escribir y probar el código.



Figura 9. Roles de la metodología Scrum

3.1.1.2. El Sprint - ¿Dónde? ¿Cuándo?

El Sprint es la unidad básica de trabajo para un equipo Scrum. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil. Es una simple iteración llevada a cabo por los miembros del equipo. Un equipo puede completar varios sprints durante el desarrollo del proyecto. Un Sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración de un entregable. El tiempo mínimo para un Sprint es de una semana y el máximo es de 4 semanas. Dentro del desarrollo de un Sprint se llevan a cabo ciertos eventos, estos reciben el nombre de Scrum Events o Eventos Scrum. Estos son: [39]

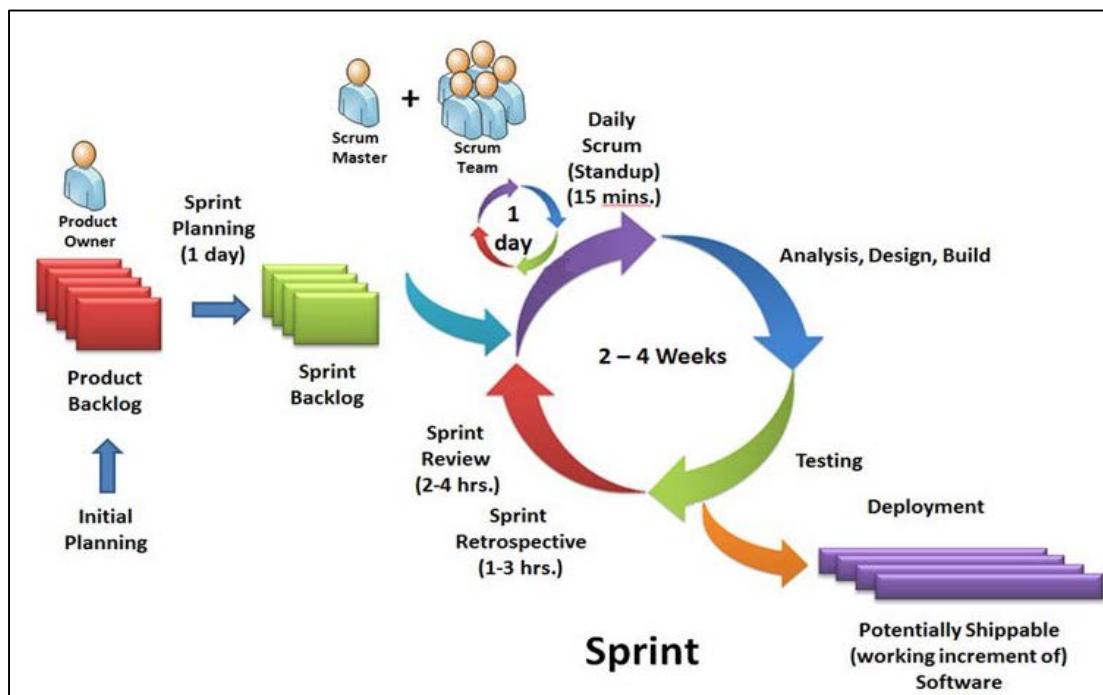


Figura 10. Sprint de la metodología Scrum

▪ Planeamiento del Sprint/Sprint Planning

Todos los involucrados en el equipo se reúnen para planificar el Sprint. Durante este evento se decide qué requerimientos o tareas se le asignará a cada uno de los elementos del equipo. Cada integrante deberá asignar el tiempo que crea prudente para llevar a cabo sus requerimientos. De esta manera se define el tiempo de duración del Sprint.

- **Reunion de Equipo de Scrum/Scrum team meeting**

Estas reuniones se deben realizar diariamente con un máximo de 15 minutos. Siempre en el mismo horario y lugar. En ellas, cada miembro del equipo deberá responder tres simples preguntas:

- ¿Qué hiciste ayer?
- ¿Qué tienes planeado hacer hoy?
- ¿Qué obstáculos encontraste en el camino?

Estas reuniones sirven para que todos los miembros del equipo se apoyen entre ellos. Si alguno de ellos tiene algún inconveniente que tome más tiempo del asignado en resolverse; este debe tratarse más a fondo en una reunión enfocada en buscar la mejor solución para ello.

- **Refinamiento del Backlog/Backlog Refinement**

El Product Owner revisa cada uno de los elementos dentro del Product Backlog con el fin de esclarecer cualquier duda que pueda surgir por parte del equipo de desarrolladores. También sirve para volver a estimar el tiempo y esfuerzo dedicado a cada uno de los requerimientos.

- **Revision del Sprint/Sprint Review**

Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Se hace una demostración de todos los requerimientos finalizados dentro del Sprint. En este punto no es necesario que todos los miembros del equipo hablen. Pueden estar presentes pero la presentación está a cargo del Scrum Master y el Product Owner.

- **Retrospectiva del Sprint/Retrospective**

En este evento, el Product Owner se reúne con todo su equipo de trabajo y su Scrum Master para hablar sobre lo ocurrido durante el Sprint. Los puntos principales a tratar en esta reunión son:

- Qué se hizo mal durante el Sprint para poder mejorar el próximo
- Qué se hizo bien para seguir en la misma senda del éxito

- Qué inconvenientes se encontraron y no permitieron poder avanzar como se tenía planificado

3.1.1.3. Herramientas Scrum - ¿Por qué? ¿Cómo?

Para poder definir las respuestas a estas preguntas, se hace uso de ciertas herramientas que Scrum nos provee. Estas son:

- **Backlog de Producto/Product Backlog**

Esto puede referirse a todo elemento que sea parte del proyecto. Puede ser un bug, una referencia o parte de un requerimiento. Brindan información muy general del proyecto y muchas veces no son tomados como requerimientos oficiales.

- **Historias de Usuario /User Stories**

Es un elemento especial del product Backlog. Son llamados Historias porque en ellos se proporciona información sobre cómo debe ser el comportamiento del requerimiento que se está trabajando. De igual manera, proporciona información directa del cliente en caso de existir algún cambio. Generalmente estos sí son tomados como requerimientos oficiales.

- **Backlog del Sprint/Sprint Backlog**

Es el conjunto de elementos tomados del Product Backlog que fueron priorizados, medidos y aceptados en las reuniones de Sprint Planning. Estos, en conjunto con sus respectivos User Stories, forman oficialmente los requerimientos a elaborar en cada uno de los Sprints que tendrá el proyecto.

- **El panel de Tareas/The Taskboard**

El panel de tareas muestra todas y cada una de las tareas que tienen asignadas cada uno de los miembros del equipo. Esta tabla se divide en tres columnas que representan el estado de la tarea:

- Por hacer
- Haciendo
- Terminado

Al inicio del Sprint todas están en la primera columna. Al momento de pasar una tarea a la columna número dos, indicará al Scrum Master y al Product Owner qué está haciendo cada miembro del equipo y cuánto tiempo lleva trabajando en dicha tarea. Al finalizar la tarea, esta debe cambiarse a la última columna. Esto quiere decir que está listo para que QA haga las pruebas necesarias.

- **Definición de “Listo” /Definition of Done**

Todo equipo eficaz y ágil tiene ciertos acuerdos que deben cumplirse antes de dar por finalizado un Proyecto. Estos son:

- Todas las tareas están completas
- Revisión de Código / Code Reviewed
- Pruebas realizadas a cada elemento desarrollado
- Revisión por parte de los clientes (que cumpla sus necesidades)
- La revisión de las condiciones de aceptación por parte del Product Owner

Estas herramientas son útiles no sólo durante un Sprint; sino que ayudan a lo largo del proyecto, ya que ayudan al equipo a entender por qué hacen lo que están haciendo. Son visibles para cada uno de los miembros del equipo y para las personas que están fuera también. Scrum no es más que una metodología que puede ser aplicable a cualquier tipo de proyecto. Aplicarlo requiere de un cambio de cultura laboral por parte de cada uno de los miembros que compondrán dicho equipo. Pero cuando el resultado sea hacer bien los proyectos en el menor tiempo posible y al menor costo, todo el sacrificio habrá valido la pena.

3.1.2. Metodología RUP

El Proceso Unificado (RUP) fue desarrollado por Philippe Kruchten, Ivar Jacobson y otros de la Rational como el proceso complementario al UML. El RUP es un armazón de proceso y como tal puede acomodar una gran variedad de procesos.

Como resultado de esta mentalidad de armazón de procesos, el RUP puede usarse en un estilo muy tradicional de cascada o de una manera ágil. Como resultado usted puede usar el RUP como un proceso ágil, o como un proceso pesado - todo depende de cómo lo adapte a su ambiente.

El proceso de ciclo de vida de RUP se divide en cuatro fases bien conocidas llamadas **Incepción, Elaboración, Construcción y Transición**. Esas fases se dividen en iteraciones, cada una de las cuales produce una pieza de software demostrable. La duración de cada iteración puede extenderse desde dos semanas hasta seis meses. Las fases son: [40]

- **Incepción.** Significa “comienzo”, se especifican los objetivos del ciclo de vida del proyecto y las necesidades de cada participante. Esto entraña establecer el alcance y las condiciones de límite y los criterios de aceptabilidad. Se identifican los casos de uso que orientarán la funcionalidad. Se diseñan las arquitecturas candidatas y se estima la agenda y el presupuesto de todo el proyecto, en particular para la siguiente fase de elaboración. Típicamente es una fase breve que puede durar unos pocos días o unas pocas semanas.
- **Elaboración.** Se analiza el dominio del problema y se define el plan del proyecto. RUP presupone que la fase de elaboración brinda una arquitectura suficientemente sólida junto con requerimientos y planes bastante estables. Se describen en detalle la infraestructura y el ambiente de desarrollo, así como el soporte de herramientas de automatización. Al cabo de esta fase, debe estar identificada la mayoría de los casos de uso y los actores, debe quedar descripta la arquitectura de software y se debe crear un prototipo de ella. Al final de la fase se realiza un análisis para determinar los riesgos y se evalúan los gastos hechos contra los originalmente planeados.
- **Construcción.** Se desarrollan, integran y verifican todos los componentes y rasgos de la aplicación. RUP considera que esta fase es un proceso de manufactura, en el que se debe poner énfasis en la administración de los recursos y el control de costos, agenda y calidad. Los resultados de esta fase (las versiones alfa, beta y otras versiones de prueba) se crean tan rápido como sea posible. Se debe compilar también una versión de entrega. Es la fase más prolongada de todas.
- **Transición.** Comienza cuando el producto está suficientemente maduro para ser entregado. Se corrigen los últimos errores y se agregan los rasgos pospuestos. La fase consiste en prueba beta, piloto, entrenamiento a usuarios y despacho del producto a mercadeo, distribución y ventas. Se produce también la

documentación. Se llama transición porque se transfiere a las manos del usuario, pasando del entorno de desarrollo al de producción.

A través de las fases se desarrollan en paralelo nueve disciplinas: Modelado de Negocios, Requerimientos, Análisis & Diseño, Implementación, Prueba, Gestión de Configuración & Cambio, Gestión del Proyecto y Entorno. Además de estas disciplinas, RUP define algunas prácticas comunes: [40]

- **Desarrollo iterativo de software.** Las iteraciones deben ser breves y proceder por incrementos pequeños. Esto permite identificar riesgos y problemas tempranamente y reaccionar frente a ellos en consecuencia.
- **Administración de requerimientos.** Identifica requerimientos cambiantes y postula una estrategia disciplinada para administrarlos.
- **Uso de arquitecturas basadas en componentes.** La reutilización de componentes permite asimismo ahorros sustanciales en tiempo, recursos y esfuerzo.
- **Modelado visual del software.** Se deben construir modelos visuales, porque los sistemas complejos no podrían comprenderse de otra manera. Utilizando una herramienta como UML, la arquitectura y el diseño se pueden especificar sin ambigüedad y comunicar a todas las partes involucradas.
- **Prueba de calidad del software.** RUP pone bastante énfasis en la calidad del producto entregado.
- **Control de cambios y trazabilidad.** La madurez del software se puede medir por la frecuencia y tipos de cambios realizados.

3.1.3. Metodología XP

La metodología Extreme Programming (XP) representa un conjunto de buenas prácticas para un desarrollo de software exitoso. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación entre todos los participantes de un proyecto, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes. [41]

3.1.3.1. Roles XP

Existen diferentes roles (actores) y responsabilidades en XP para diferentes tareas y propósitos durante el proceso: [41]

TABLA III. ROLES DE LA METODOLOGÍA XP

Programador (Programmer)	<ul style="list-style-type: none"> ✓ Pieza básica en desarrollos XP. ✓ Responsable sobre la generación de código fuente. ✓ Responsable sobre el diseño y maquetado de la aplicación. ✓ Responsable de la administración de Bases de Datos. ✓ Responsable sobre la integridad del sistema (pruebas). ✓ Sin distinción entre analistas, diseñadores o codificadores; en XP, los programadores diseñan, programan y realizan las pruebas.
Cliente (Customer)	<ul style="list-style-type: none"> ✓ Pieza básica en desarrollos XP. ✓ Define especificaciones. ✓ Influye sin controlar. ✓ Confía en el grupo de desarrollo. ✓ Define pruebas funcionales.
Entrenador (Coach)	<ul style="list-style-type: none"> ✓ El líder del equipo - toma las decisiones importantes. ✓ Principal responsable del proceso. ✓ Tiende a estar en un segundo plano a medida que el equipo madura.
Rastreador (Tracker)	<ul style="list-style-type: none"> ✓ Observa sin molestar. ✓ Recoge, analiza y publica información sobre la marcha del proyecto sin afectar demasiado el proceso. ✓ Supervisa el cumplimiento de las estimaciones en cada iteración. ✓ Informa sobre la marcha de la iteración en curso. ✓ Controla la marcha de las pruebas funcionales, de los errores reportados, de las responsabilidades aceptadas y de la prueba añadida por los errores.
Probador (Tester)	<ul style="list-style-type: none"> ✓ Ayuda al cliente con las pruebas funcionales. ✓ Se asegura de que los test funcionales se ejecuten

3.1.3.2. Valores XP

- ✓ **Comunicación:** Todos son parte del equipo y la comunicación es esencial, desde los requerimientos hasta la programación. En los métodos tradicionales de desarrollo de software, la comunicación de los requerimientos a los desarrolladores se realiza a través de la documentación.
- ✓ **Simplicidad:** Nos dirigiremos a nuestro objetivo a pasos simples y pequeños, mitigando las fallas a medida que ocurran. Crearemos algo de lo cual podamos sentirnos orgullos y que pueda mantenerse en el largo plazo a costos razonables. Asimismo, un diseño y programación simple mejora la calidad de las comunicaciones, pues es más fácil de implementar y entender por todos en el equipo.
- ✓ **Retroalimentación (Feedback):** Se tomará seriamente los compromisos con el usuario establecidos en todas las iteraciones, entregando software en funcionamiento en cada una. Mostraremos al usuario nuestro software frecuentemente y de forma temprana, escuchando cuidadosamente sus observaciones y realizando los cambios que sean necesarios. Adaptaremos nuestros procesos al proyecto y no al contrario.
 - **Retroalimentación del sistema:** Por medio de la ejecución de pruebas unitarias y de integración, los programadores reciben retroalimentación directa del estado del sistema.
 - **Retroalimentación del cliente (usuario):** Las pruebas de aceptación, son diseñadas conjuntamente por el cliente y los analistas de pruebas, obteniendo en conjunto retroalimentación del estado actual del sistema. Esta revisión puede hacerse cada 1 o 2 semanas, permitiendo así que el cliente sea quien guíe el desarrollo del software.
 - **Retroalimentación del equipo:** Cuando el cliente trae nuevos requerimientos, el equipo puede directamente proporcionar la estimación del tiempo que tomará implementarlos.

- ✓ **Coraje:** Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave.
- ✓ **Respeto:** Todos en el equipo dan y reciben el respeto que merecen como integrantes del equipo y los aportes de cada integrante, son valorados por todos. Todos contribuyen, así sea simplemente con entusiasmo. Los desarrolladores respetan la experticia de los clientes y viceversa. La Gerencia respeta el derecho del equipo de asumir responsabilidad y tener autoridad sobre su trabajo. Respeto es tanto por el trabajo de los demás como por el trabajo de uno mismo. [41]

3.1.3.3. Actividades de XP

- **Codificar**

Es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.

- **Hacer pruebas**

Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.

- **Escuchar**

Si vamos a hacer pruebas tenemos que preguntar si lo obtenido es lo deseado, y tenemos que preguntar a quién necesita la información. Tenemos que escuchar a nuestros clientes cuáles son los problemas de su negocio, debemos de tener una escucha activa explicando lo que es fácil y difícil de obtener, y la realimentación entre ambos nos ayudan a todos a entender los problemas.

- **Diseñar**

El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

3.1.3.4. Fases de la Metodología XP

El ciclo de vida de XP se resalta en el carácter interactivo e incremental del desarrollo de software. Las iteraciones son cortas, puesto que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se va a obtener y esto va a representar una mejor calidad del producto a largo plazo. Existe una fase de análisis inicial orientada a programar las iteraciones de desarrollo y cada iteración incluye diseño, codificación y pruebas, fases superpuestas de tal manera que no se separan en el tiempo [41].

En la siguiente figura se muestra las fases en las que se subdivide el ciclo de vida XP: [41]

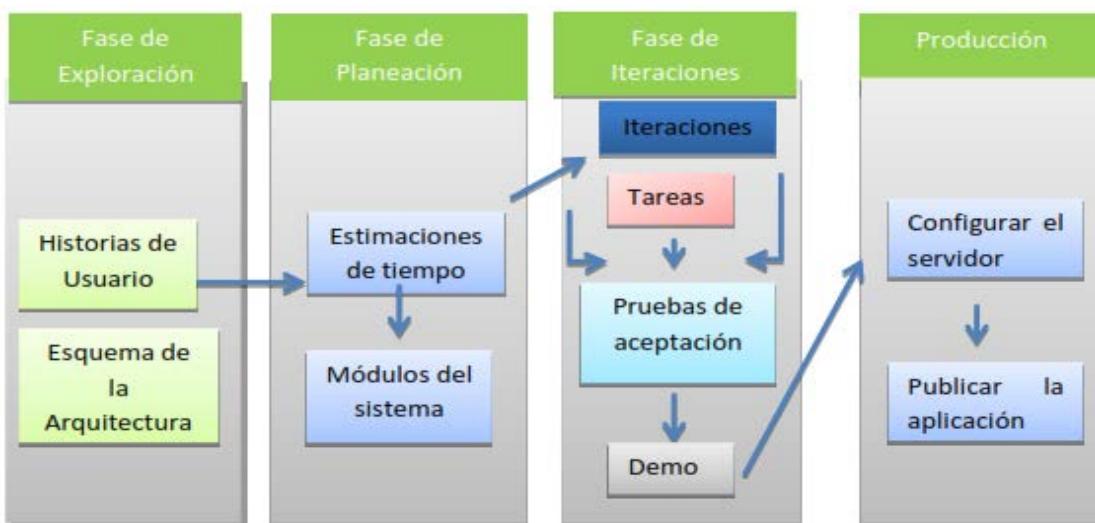


Figura 11. Fases de la metodología XP

- **Fase de la exploración**

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- **Fase del planeamiento**

Se priorizan las historias de usuario y se acuerda el alcance del release. Los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma. La duración del cronograma del primer release no excede normalmente dos meses. La fase de planeamiento toma un par de días. Se deben incluir varias iteraciones para lograr un release. El cronograma fijado en la etapa de planeamiento se realiza a un número de iteraciones, cada una toma de una a cuatro semanas en ejecución. La primera iteración crea un sistema con la arquitectura del sistema completo. Esto es alcanzado seleccionando las historias que harán cumplir la construcción de la estructura para el sistema completo. El cliente decide las historias que se seleccionarán para cada iteración. Las pruebas funcionales creadas por el cliente se ejecutan al final de cada iteración. Al final de la última iteración el sistema está listo para producción.

- **Fase de iteraciones**

Requiere prueba y comprobación extra del funcionamiento del sistema antes de que éste se pueda liberar al cliente. En esta fase, los nuevos cambios pueden todavía ser encontrados y debe tomarse la decisión de si se incluyen o no en el release actual. Durante esta fase, las iteraciones pueden ser aceleradas de una a tres semanas. Las ideas y las sugerencias propuestas se documentan para una puesta en práctica posterior por ejemplo en la fase de mantenimiento. Después de que se realice el primer release productivo para uso del cliente, el proyecto de XP debe mantener el funcionamiento del sistema mientras realiza nuevas iteraciones.

- **Fase de producción**

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno real. También se toma decisiones sobre las características del servidor de aplicaciones a implementar.

3.1.3.5. Ventajas y Desventajas de la Metodología XP

- **Ventajas**

- ✓ Programación organizada.
- ✓ Implementa una forma de trabajo donde se adapta fácilmente a las circunstancias.
- ✓ Fomenta la comunicación entre los clientes y los desarrolladores.
- ✓ Puede ser aplicada a cualquier lenguaje de programación.
- ✓ El cliente tiene el control sobre las prioridades.
- ✓ Se hacen pruebas continuas durante el proyecto.

- **Desventajas**

- ✓ Se recomienda emplearlo en proyectos a corto plazo.
- ✓ Altas comisiones en caso de fallar.
- ✓ Imposible prever todo antes de programar.

3.1.4. Comparativa de metodologías para el desarrollo de aplicaciones web.

TABLA IV. COMPARATIVA DE METODOLOGÍAS PARA EL DESARROLLO DE APLICACIONES WEB

METODOLOGÍA	CARACTERÍSTICAS	VENTAJAS	DESVENTAJAS
XP	<ul style="list-style-type: none"> ✓ Desarrollo iterativo e incremental. ✓ Pruebas unitarias contínuas frecuentemente repetidas. ✓ Las iteraciones de entrega son en 3 semanas ✓ Las tareas que se van terminando en las diferentes entregas al cliente son susceptibles a modificaciones ✓ Se centra más en la programación o creación del producto. 	<ul style="list-style-type: none"> ✓ Comunicación. ✓ Simplicidad. ✓ Alta calidad en mínimo tiempo. 	<ul style="list-style-type: none"> ✓ Se usa en proyectos pequeños.
SCRUM	<ul style="list-style-type: none"> ✓ Las iteraciones de entrega son en 4 semanas. ✓ Lo que se termina, funciona y está bien, se aparta y ya no se toca. ✓ Se basa más en la administración del proyecto. 	<ul style="list-style-type: none"> ✓ Los miembros en un proyecto XP trabajan en parejas. 	<ul style="list-style-type: none"> ✓ Cada integrante del equipo de Scrum trabaja de forma individual.
RUP	<ul style="list-style-type: none"> ✓ Programación por equipos. ✓ Interacción con usuarios estratégicos. ✓ Desarrollo interno en etapas iterativas. 	<ul style="list-style-type: none"> ✓ Mayor documentación. ✓ Configuración y control de cambios. ✓ Es centrado en arquitectura guiado por riesgos. 	<ul style="list-style-type: none"> ✓ Los cambios son en una fase

En base a la comparativa realizada se ha seleccionado XP como metodología de desarrollo de la aplicación puesto que muestra algunas ventajas como es el trabajo en conjunto con el usuario lo que garantiza un mejor acercamiento a los requerimientos que el cliente desea.

e. Materiales y Métodos

Para el desarrollo del presente trabajo de titulación se ha visto como conveniente utilizar los materiales y métodos descritos a continuación.

1. Materiales

Para el desarrollo del presente trabajo de titulación los materiales utilizados son: Codeigniter como framework para el desarrollo de la aplicación, así mismo Sublime Text como editor de texto, Enterprise Architect para el modelado, pencil Project para el prototipado de pantallas, Gantt Project para la realización del cronograma.

Otra herramienta utilizada es Xampp, herramienta de desarrollo mediante la cual se pudo probar la aplicación localmente sin necesidad de tener que alojarla a un host en internet.

2. Métodos

Para el desarrollo del presente trabajo de titulación se utilizaron diferentes métodos y técnicas que ayudaron a cumplir con los objetivos planteados.

Método Deductivo: Este método se aplicó para obtener la información necesaria sobre el uso de la red social Twitter para determinar posibles emergencias y determinar la veracidad de éstas.

Método Inductivo: A través de este método se plantearon las principales dificultades al determinar posibles emergencias mediante el uso de Twitter y determinar si la solución planteada los resolvió.

3. Técnicas

Encuesta: Con esta técnica se obtuvo información relevante y necesaria referente al tema de la investigación, la cual ayudara a sustentarlo y justificarlo.

Técnica de la investigación Bibliográfica: Con esta técnica se sustentó la base teórica de la investigación, mediante consultas a: fuentes bibliográficas confiables, libros, revistas indexadas, artículos científicos, base de datos científicas entre otras.

4. Metodología

Para la elaboración del presente proyecto se implementó la metodología de desarrollo XP, la cual cuenta con 4 fases: exploración, planeamiento, iteraciones y producción mediante las cuales se pudo dar cumplimiento con el proyecto.

f. Resultados

Como una de las primeras pretensiones de la investigación, se definió la hipótesis a evidenciar con el trabajo de investigación. Así mismo, se plantearon cuatro fases que ayuden en la realización de la investigación.

✓ **HIPÓTESIS**

Hipótesis: ¿La monitorización de una cuenta de Twitter a través de una aplicación web permite que las emergencias publicadas mediante esta red social sean atendidas?

En cada una de las fases se han realizado diferentes tareas con el fin de comprobar la interrogante planteada, las cuales se detallan a continuación.

1. PRIMERA FASE: Exploración

En esta fase de realiza el análisis de la problemática y se obtienen los requerimientos del proyecto a desarrollar.

Para poder identificar los requerimientos se llevó a cabo reuniones con el personal del departamento de Comunicación del Sistema Integrado de Seguridad ECU911 de la ciudad de Loja. Se determinaron requerimientos de usuario.

1.1. Historias de Usuario

Se pretende realizar una aplicación web que permita realizar el análisis estadístico y monitoreo de una cuenta de Twitter. Con esta base se ha recolectado información que permita conocer las necesidades de los usuarios al hacer uso de una cuenta de Twitter.

1. Control y acceso al sistema

TABLA V. HISTORIA DE USUARIO CONTROL Y ACCESO AL SISTEMA

HISTORIA DE USUARIO	
Número: 01	Usuario: Usuario
Nombre de la Historia: Control y acceso al sistema	
Prioridad en el negocio: Alta	Riesgos en el Desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 1
Programador Responsable: Franklin Michay	
Descripción: El usuario tendrá acceso a un formulario LOGIN, el cual comprobará, si los datos introducidos corresponden al usuario real; previo ingreso hacia el sistema.	
La pantalla tendrá las siguientes características: <ul style="list-style-type: none">▪ Un formulario LOGIN.▪ Campos de texto con una breve descripción, de la información a ingresarse.▪ Un campo CONTRASEÑA que mostrará asteriscos en lugar de caracteres.	
Eventos al presionar el botón Ingresar. <ul style="list-style-type: none">▪ Se verificará que todos los campos de texto estén llenos, caso contrario el sistema mostrará un mensaje de alerta, correspondiente al campo requerido, sin afectar los campos llenos.	
Observaciones:	

2. Control y acceso una cuenta de Twitter

TABLA VI. HISTORIA DE USUARIO CONTROL Y ACCESO A UNA CUENTA DE TWITTER

HISTORIA DE USUARIO	
Número: 02	Usuario: Usuario
Nombre de la Historia: Control y acceso a una cuenta de Twitter	
Prioridad en el negocio: Alta	Riesgos en el Desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador Responsable: Franklin Michay	
Descripción: El usuario tendrá acceso a un formulario LOGIN que se obtendrá de la autenticación con el API de Twitter, el cual comprobará, si los datos introducidos corresponden al usuario real de una cuenta de Twitter; previo ingreso hacia el sistema.	
La pantalla tendrá las siguientes características: <ul style="list-style-type: none">▪ Un formulario LOGIN.▪ Campos de texto con una breve descripción, de la información a ingresarse.▪ Un campo CONTRASEÑA que mostrará asteriscos en lugar de caracteres.	
Eventos al presionar el botón Ingresar. <ul style="list-style-type: none">▪ Se verificará que todos los campos de texto estén llenos, caso contrario el sistema mostrará un mensaje de alerta, correspondiente al campo requerido, sin afectar los campos llenos.	
Observaciones: Los usuarios deben poseer una cuenta de Twitter activa con la que podrán acceder al sistema haciendo uso de su usuario y contraseña.	

3. Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter

TABLA VII. HISTORIA DE USUARIO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER

HISTORIA DE USUARIO	
Número: 03	Usuario: Usuario
Nombre de la Historia: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter	
Prioridad en el negocio: Alta	Riesgos en el Desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 1
Programador Responsable: Franklin Michay	
Descripción: El usuario podrá visualizar la estadística e información general de su cuenta de Twitter. La información que se mostrará es: número de seguidores, número siguiendo, numero de tweets, listas y una estadística de los seguidores, siguiendo, nuevos seguidores y menciones por día.	
Características:	
La pantalla Perfil de usuario mostrará: <ul style="list-style-type: none">▪ Información acerca de los seguidores, siguiendo, número de tweets, número de listas.▪ El Time-line de sus tweets.	
La pantalla General mostrará: <ul style="list-style-type: none">▪ Información acerca de nuevos seguidores, menciones por día, tweets por día.▪ Un diagrama de barras con estadística de la ubicación de seguidores.▪ Un diagrama de barras con estadística de la ubicación de siguiendo.	
La pantalla Seguidores mostrará: <ul style="list-style-type: none">▪ Una tabla con información general de seguidores.	
La pantalla Siguiendo mostrará: <ul style="list-style-type: none">▪ Una tabla con información general de las personas a las que está siguiendo.	
Observaciones:	

4. Mostrar lista de tweets (Favoritos, de usuario, de la comunidad, de Alertas) de una cuenta de Twitter.

TABLA VIII. HISTORIA DE USUARIO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER

HISTORIA DE USUARIO	
Número: 04	Usuario: Usuario
Nombre de la Historia: Mostrar lista de tweets (Favoritos, de usuario, de la comunidad, de alertas) de una cuenta de Twitter.	
Prioridad en el negocio: Media	Riesgos en el Desarrollo: Media
Puntos estimados: 4	Iteración asignada: 1
Programador Responsable: Franklin Michay	
Descripción: El usuario podrá visualizar la información de tweets que se vinculan a su cuenta de Twitter. La información que se mostrará es: Lista de tweets Favoritos, Lista de tweets de su comunidad, Lista de tweets publicados por el usuario, y Lista de tweets que pertenecen a una Alerta.	
Características:	
<p>La pantalla [Tweets de tu Comunidad] mostrará:</p> <ul style="list-style-type: none"> ▪ Los tweets que son publicados por su comunidad (seguidores y siguiendo). <p>La pantalla [Tus Tweets] mostrará:</p> <ul style="list-style-type: none"> ▪ Los tweets que ha publicado. <p>La pantalla [Tweets de tus Alertas] mostrará:</p> <ul style="list-style-type: none"> ▪ Los tweets que pertenecen a una alerta. <p>La pantalla [Tus tweets Favoritos] mostrará:</p> <ul style="list-style-type: none"> ▪ Los tweets a los que ha pinchado me gusta. 	
Observaciones: Para mostrar los tweets de la pantalla [Tweets de tus Alertas], el usuario tendrá que haber ejecutado previamente una Alerta.	

5. Mostrar estadística de la actividad de una cuenta de Twitter.

TABLA IX. HISTORIA DE USUARIO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER

HISTORIA DE USUARIO	
Número: 05	Usuario: Administrador, Usuario
Nombre de la Historia: Mostrar estadística de la actividad de una cuenta de Twitter	
Prioridad en el negocio: Media	Riesgos en el Desarrollo: Alta
Puntos estimados: 5	Iteración asignada: 2
Programador Responsable: Franklin Michay	
Descripción: El usuario podrá visualizar la estadística de la actividad su cuenta de Twitter respecto a su comunidad. La información que se mostrará es: Usuarios que más retuitean, usuarios que más mencionan y los tweets más retuiteados. El usuario podrá generar la estadística de cada alerta creada.	
Características:	
La pantalla [Estadística de tu Comunidad] mostrará: <ul style="list-style-type: none">▪ La lista de usuarios que más nos retuitean.▪ La lista de usuarios que más nos mencionan▪ Un diagrama tipo pastel donde se muestra el porcentaje de retweets de cada usuario.▪ Un diagrama tipo pastel donde se muestra el porcentaje de menciones de cada usuario.▪ Un diagrama de barras donde se muestra los tweets más retuiteados con el número de retweets y Favoritos.	
La pantalla [Estadística de tus Alertas] mostrará: <ul style="list-style-type: none">▪ La lista de alertas.▪ Un diagrama de barras donde se muestra los tweets más retuiteados de cada alerta.	
Eventos al presionar el botón Estadística. Se verifica que cada alerta tenga información para generar la estadística, caso contrario el sistema mostrará un mensaje de alerta.	
La pantalla [Estadística de tus Tweets] mostrará:	

- Un diagrama de barras donde se muestra la estadística del número tweets, RT y menciones que se generan por día durante los últimos siete días.
- Un diagrama de barras donde se muestra la estadística del número tweets, RT y menciones que se generan por día durante el último mes.
- Un diagrama de barras donde se muestra la estadística del número tweets, RT y menciones que se generan por día al seleccionar un rango de fechas.

Eventos al presionar el botón Consultar.

Se verificará que el usuario seleccione un rango de fechas, caso contrario el sistema mostrará un mensaje de alerta.

Observaciones:

6. Publicar y programar tweets

TABLA X. HISTORIA DE USUARIO PUBLICAR Y PROGRAMAR TWEETS

HISTORIA DE USUARIO	
Número: 06	Usuario: Administrador, Usuario
1. Nombre de la Historia: Publicar y programar tweets	
Prioridad en el negocio: Alta	Riesgos en el Desarrollo: Medio
Días estimados: 6	Iteración asignada: 2
Programador Responsable: Franklin Michay	
Descripción: El usuario podrá publicar y/o programar tweets	
Características:	
La pantalla [General] mostrará: <ul style="list-style-type: none"> ▪ Un botón que desplegará una pantalla emergente para publicar tweets. ▪ Un botón que desplegará una pantalla emergente para programar tweets. 	
Eventos al presionar el botón [Publicar Tweet]: Se despliega una pantalla emergente donde se podrá escribir el contenido del tweet el cual deberá ser de 144 caracteres y un campo para cargar imagen si así lo decide el usuario.	
Eventos al presionar el botón [Programar Tweet]: Se despliega una pantalla emergente donde se podrá escribir el contenido del	

tweet el cual deberá ser de 144 caracteres, un campo para cargar imagen si así lo decide el usuario, un calendario donde se podrá elegir la fecha y un menú desplegable para seleccionar la hora de programación del tweet.

Observaciones:

7. Administrar Alertas

TABLA XI. HISTORIA DE USUARIO ADMINISTRAR ALERTAS

HISTORIA DE USUARIO	
Número: 07	Usuario: Administrador, Usuario
Nombre de la Historia: Administrar Alertas	
Prioridad en el negocio: Medio	Riesgos en el Desarrollo: Medio
Puntos estimados: 7	Iteración asignada: 2
Programador Responsable: Franklin Michay	
Descripción: El usuario podrá crear, modificar, eliminar y ejecutar alerta. Los campos a ingresar son: nombre y parámetros.	
Características:	
La pantalla [Alertas] mostrará: <ul style="list-style-type: none"> ▪ Los botones Crear Alerta, Editar, Eliminar y Ejecutar Alerta. 	
Eventos al presionar el botón Crear Alerta. <ul style="list-style-type: none"> ▪ Se mostrará la pantalla [Crear Alerta] donde se verificará que todos los campos de texto estén llenos, caso contrario el sistema mostrará un mensaje de alerta, correspondiente al campo requerido, sin afectar los campos llenos. 	
Observaciones: Tras ejecutar una alerta, el sistema verifica si existen tweets que coincidan con la alerta creada (esto se determinará de acuerdo al texto del tweet y los campos al crear una alerta) y notificará al correo electrónico la existencia del mismo caso contrario mostrará un mensaje de alerta.	

8. Administrar Usuarios

HISTORIA DE USUARIO	
Número: 08	Usuario: Administrador
Nombre de la Historia: Administrar Usuarios	
Prioridad en el negocio: Alto	Riesgos en el Desarrollo: Medio
Puntos estimados: 8	Iteración asignada: 3
Programador Responsable: Franklin Michay	
Descripción: El usuario administrador podrá crear, modificar, eliminar y ver estadísticas de un usuario. Los campos a ingresar son: nombres, apellidos, correo, usuario, contraseña, tipo de usuario y estado (activo e inactivo).	
Características: La pantalla [Administrar Usuarios] mostrará: <ul style="list-style-type: none">▪ Los botones Crear Nuevo Usuario, Modificar, Ver Alertas, Ver Tweets Programados. Eventos al presionar el botón Crear Alerta. <ul style="list-style-type: none">▪ Se mostrará la pantalla [Crear Alerta] donde se verificará que todos los campos de texto estén llenos, caso contrario el sistema mostrará un mensaje de alerta, correspondiente al campo requerido, sin afectar los campos llenos. Observaciones: Tras presionar el botón [Ver Alertas], el sistema verifica si existen alertas creadas por ese usuario y las mostrará en la pantalla [Alertas Creadas]. De la misma manera al presionar el botón [Ver Tweets programados], el sistema verificará si existen tweets programados por ese usuario y los mostrará en la pantalla [Tweets Programados].	

2. SEGUNDA FASE: Planeación

2.1. Cronograma de Actividades

En base a las historias de usuario, de acuerdo a ello se las ordenó y se obtuvo los módulos del sistema. Así mismo, se estableció un cronograma de actividades; el cual debe realizarse en dos iteraciones.

TABLA XII. CRONOGRAMA DE ACTIVIDADES DE LA APLICACIÓN

MÓDULO	N.-	Nombre Historia	Esfuerzo de Desarrollo		
			Semanas Ideales	Días Ideales	Horas Ideales
1	01	Control y acceso al sistema	1	5	40
	02	Control y acceso a una cuenta de Twitter	1	5	40
	03	Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter	3	15	120
	04	Mostrar Time-line de tweets (Favoritos, Time-line de usuario, Home-time-line, Time-line de Alertas) de una cuenta de Twitter	4	20	160
2	05	Mostrar estadística de la actividad de una cuenta de Twitter	6	30	240
	06	Publicar y programar tweet	1.5	8	64
	07	Administrar Alertas	3	15	120
	08	Administrar Usuarios	3	15	120

2.2. Módulos del Sistema

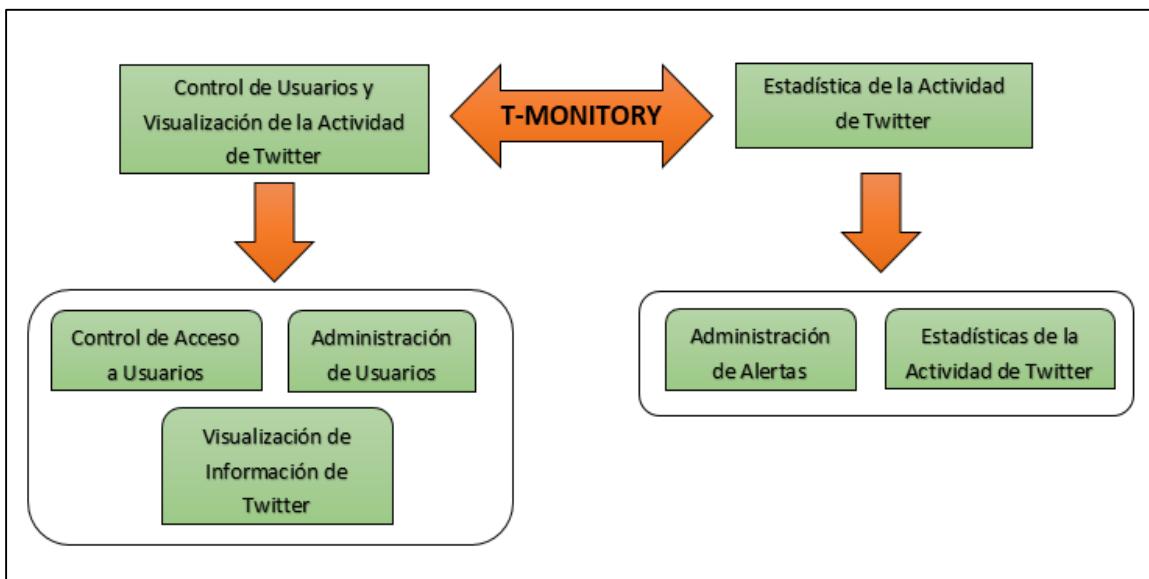


Figura 12. Módulos de la aplicación

✓ Control Usuarios y Visualización de información de Twitter

Los usuarios que necesiten usar la aplicación web deberán autenticarse en el sistema utilizando un usuario con los permisos necesarios para poder realizar las operaciones deseadas. La aplicación brindará acceso a una pantalla inicial donde el usuario deberá acceder con su usuario y contraseña previamente generados en la base de datos; una vez dentro del sistema el usuario tendrá que loguearse con el usuario y contraseña de una cuenta de Twitter activa para poder acceder totalmente al sistema.

Una vez dentro de la aplicación el usuario podrá visualizar toda la información general de su cuenta de twitter; así mismo, se encargará de guardar en base de datos toda esta información para luego mostrarla mediante diagramas estadísticos.

✓ Estadística de actividad de Twitter

Será el encargado de mostrar la estadística de la actividad de una cuenta de Twitter mediante diagramas de barras y diagramas de pastel. Así mismo se podrá administrar alertas las cuales nos permitirán monitorear una cuenta de Twitter.

2.3. Arquitectura de la aplicación

La arquitectura general del sistema está compuesta por un servidor principal de Twitter, un servidor de base de datos y el usuario.

En el servidor de base de datos se encuentra la base de datos administrada bajo el gestor de base de datos Mysql quien proporciona toda la información de una cuenta de Twitter.

El servidor de la aplicación web es el encargado de enviar las respuestas a las consultas hechas por el cliente acerca de la información de los registros de Twitter. (Ver figura 6)

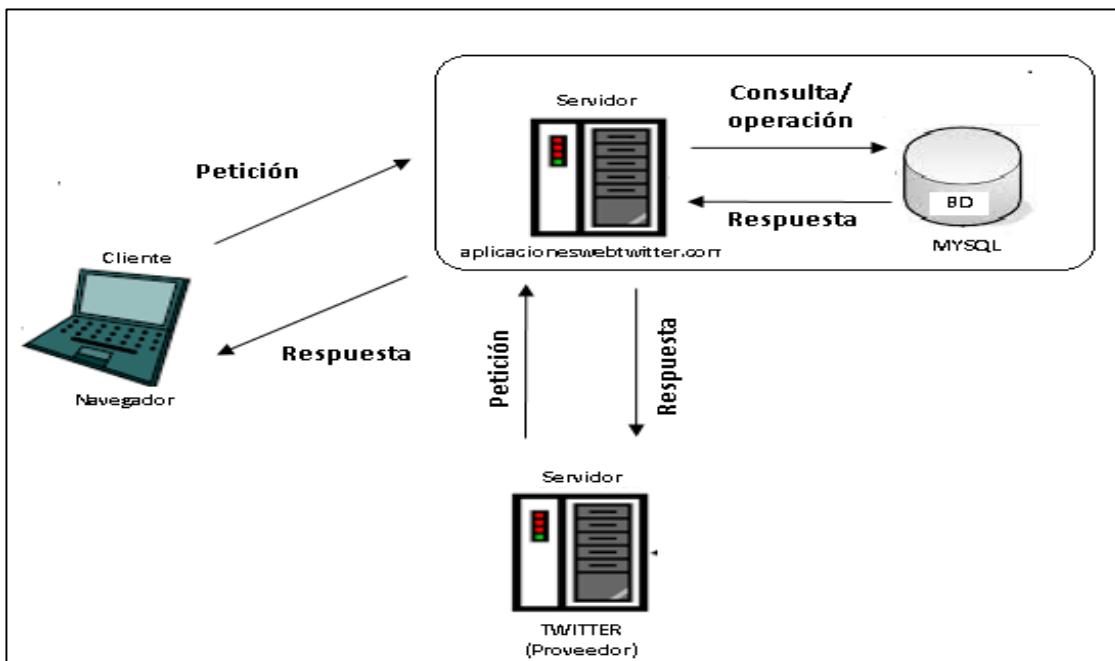


Figura 13. Arquitectura de la aplicación

3. TERCERA FASE: Iteraciones

3.1. ITERACIÓN I

3.1.1. Cronograma de Actividades

TABLA XIII. CRONOGRAMA DE ACTIVIDADES: ITERACIÓN I

ITERACIÓN	N.-	Nombre Historia	Duración Semanas	Duración Horas
1	01	Control y acceso al sistema	1	40
1	02	Control y acceso a una cuenta de Twitter	1	40
1	03	Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter	3	120
1	04	Mostrar lista de tweets (Favoritos, de usuario, de la comunidad, de Alertas) de una cuenta de Twitter	4	160
				Total=360

3.1.2. Tareas por historia: Iteración I

✓ Historia 1: Control y acceso al sistema

**TABLA XIV. CRONOGRAMA DE ACTIVIDADEDES DEL CASO DE USO
CONTROL Y ACCESO AL SISTEMA**

Historia:1	Control y acceso al sistema	
Número	Nombre	Tiempo Estimado
1	Configuración del entorno de desarrollo Web	8 horas
2	Creación de scripts utilizando SQL	8 horas
3	Diseño de Interfaz gráfica login	8 horas
4	Mapeo Objeto Relacional- Entidad Usuario	8 horas
5	Creación de funciones PHP	8 horas
		40 Horas

2. Configuración del entorno de desarrollo Web

TABLA XV. TAREA 1: CONFIGURACIÓN DEL ENTORNO DE DESARROLLO WEB DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA

N.- Tarea: 1	N.- Historia: 1
Nombre de la Tarea: Configuración del entorno de desarrollo Web	
Tipo de Tarea: General	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de instalación y configuración de un entorno de desarrollo web, utilizando software libre.	
Componentes a implementar:	
Base de datos: Mysql 5.0, PhpMyAdmin 4.6.5.2	
Framework: Codeigniter	
Lenguaje de programación: PHP 5.6.30	
Servidor de Páginas Web: Apache 2.4.25	
Componentes: Twitter Bootstrap	
Editores de código: SublimeText 3	

3. Creación de scripts utilizando SQL

TABLA XVI. TAREA 2: CREACIÓN DE SCRIPTS UTILIZANDO SQL
DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA

N.- Tarea: 2	N.- Historia: 1
Nombre de la Tarea: Creación de scripts utilizando SQL	
Tipo de Tarea: Administrador de Base de Datos	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación de script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> . create database tweett: Permite crear la base de datos “tweett” create table login: Permite crear la tabla “login”, con sus respectivas claves primarias y tipos de datos. insert into login: Permite ingresar datos de los usuarios manualmente, pues en el momento no se dispone de formularios para administrar usuarios.	

4. Diseño de Interfaz gráfica login

TABLA XVII.TAREA 3: DISEÑO DE INTERFÁZ GRÁFICA LOGIN DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA

N.- Tarea: 3	N.- Historia: 1
Nombre de la Tarea: Diseño de Interfaz gráfica login	
Tipo de Tarea: Diseño y maquetado	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: La tarea describe el proceso de diseño, para el formulario login, mediante la utilización de hojas de estilo CSS y archivos JS, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para mejorar el diseño de nuestros componentes <i>HTML</i> .	

5. Mapeo Objeto Relacional- Entidad Usuario

TABLA XVIII. TAREA 4: MAPEO OBJETO RELACIONAL ENTIDAD USUARIO DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA

N.- Tarea: 4	N.- Historia: 1
Nombre de la Tarea: Mapeo Objeto Relacional- Entidad Usuario	
Tipo de Tarea: Programación	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de mapeo de la entidad usuario, aplicando el paradigma de la programación orientada a objetos, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. <ul style="list-style-type: none">▪ Implementación del driver “mysqli”, para gestionar las conexiones hacia nuestra base de datos de manera rápida y segura.▪ Conversión de nuestra tabla “login” en un objeto, utilizando una clase <i>PHP</i>, para poder instanciarlo y acceder a sus respectivos atributos y métodos.	

6. Creación de archivos PHP

TABLA XIX. TAREA 5: CREACIÓN DE ARCHIVOS PHP DEL CASO DE USO CONTROL Y ACCESO AL SISTEMA

N.- Tarea: 5	N.- Historia: 1
Nombre de la Tarea: Creación de funciones PHP	
Tipo de Tarea: Programación	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones PHP, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.	
function comprobar: Verifica si el correo y contraseña de un usuario está registrado en la base de datos, para proceder a crear las variables de tipo sesión.	

✓ **Historia 2: Control y acceso a una cuenta de Twitter**

TABLA XX. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
CONTROL Y ACCESO A UNA CUENTA DE TWITTER

Historia:2	Control y acceso a una cuenta de Twitter	
Número	Nombre	Tiempo Estimado
1	Creación de scripts utilizando SQL	8 horas
2	Creación de una aplicación de desarrollo para acceder a los datos de una cuenta de Twitter mediante su API.	8 horas
3	Configuración de credenciales para el acceso al API de Twitter.	8 horas
4	Creación de funciones PHP para la verificación de datos de usuario de una cuenta de Twitter.	8 horas
5	Creación de funciones PHP para guardar datos del usuario autenticado.	8 horas
		40 Horas

1. Creación de scripts utilizando SQL

TABLA XXI. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL
DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE
TWITTER

N.- Tarea: 1	N.- Historia: 2
Nombre de la Tarea: Creación de scripts utilizando SQL	
Tipo de Tarea: Administrador de Base de Datos	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación del script <i>SQL</i> , para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> . create table usuario: Permite crear la tabla “usuario”, con sus respectivas claves primarias y tipos de datos.	

2. Creación de una aplicación de desarrollo para acceder a los datos de una cuenta de Twitter mediante su API.

TABLA XXII. TAREA 2: CREACIÓN DE UNA APLICACIÓN DE DESARROLLO PARA ACCEDER A LOS DATOS DE UNA CUENTA DE TWITTER DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER

N.- Tarea: 2	N.- Historia: 2
Nombre de la Tarea: Creación de una aplicación de desarrollo para acceder a los datos de una cuenta de Twitter mediante su API.	
Tipo de Tarea: General	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación de una aplicación que nos permita obtener las credenciales de acceso al API de Twitter, mediante las cuales se podrá acceder a toda la información de una cuenta implementando las funciones que nos proporciona en su documentación.	

3. Configuración de credenciales para el acceso al API de Twitter.

**TABLA XXIII. TAREA 3: CONFIGURACIÓN DE CREDENCIALES
PARA EL ACCESO AL API DE TWITTER DEL CASO DE USO
CONTROL Y ACCESO A UNA CUENTA DE TWITTER**

N.- Tarea: 3	N.- Historia: 2
Nombre de la Tarea: Configuración de credenciales para el acceso al API de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: La tarea describe el proceso de utilización de las credenciales (tokens) obtenidos en la tarea 2. Los tokens o claves de autenticación (“consumer_key” y “consumer_secret”) permiten acceder y verificar toda la información de un usuario al acceder con su nombre de usuario/ screen_name y su contraseña.	

4. Creación de funciones PHP para la verificación de datos de usuario de una cuenta de Twitter.

TABLA XXIV. TAREA 4: CREACIÓN DE FUNCIONES PHP PARA LA VERIFICACIÓN DE DATOS DE USUARIO DE UNA CUENTA DE TWITTER DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER

N.- Tarea: 4	N.- Historia: 2
Nombre de la Tarea: Creación de funciones PHP para la verificación y extracción de datos de usuario de una cuenta de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones para la verificación y extracción de datos de usuario de una cuenta de Twitter mediante la utilización de la librería Twconnect, librería en donde se utilizarán todas las funciones del API de Twitter. function twaccount_verify_credentials(): Permite extraer información general (nombre, screen_name, ubicación, descripción, fecha de registro) de una cuenta de Twitter.	

5. Creación de funciones PHP para guardar datos del usuario autenticado.

TABLA XXV. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR DATOS DEL USUARIO AUTENTICADO DEL CASO DE USO CONTROL Y ACCESO A UNA CUENTA DE TWITTER

N.- Tarea: 5	N.- Historia: 2
Nombre de la Tarea: Creación de funciones PHP para guardar datos del usuario autenticado.	
Tipo de Tarea: Programación	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones <i>PHP</i> , que nos permitirá guardar la información de autenticación de un usuario de Twitter. function guardar_datos_usuario(): Verifica si los datos a guardar ya existen en la base de datos, esto lo realiza mediante el id del usuario autenticado. Si no hay datos relacionados con ese id se guardarán los datos, caso contrario no realizará dicha acción.	

- ✓ **Historia 3:** Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter.

**TABLA XXVI.CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA
CUENTA DE TWITTER**

Historia :3	Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter	
Número	Nombre	Tiempo Estimado
1	Creación de scripts utilizando SQL.	8
2	Creación de funciones PHP para la extracción de datos del perfil y la comunidad de un usuario de Twitter.	40
3	Diseño de interfaces gráficas Perfil de Usuario y General.	16
4	Diseño de interfaces gráficas Seguidores y Siguiendo.	16
5	Creación de funciones PHP para guardar datos del perfil y la comunidad de un usuario de Twitter.	40
		120 Horas

1. Creación de scripts utilizando SQL

TABLA XXVII. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL
DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN
GENERAL DE UNA CUENTA DE TWITTER

N.- Tarea: 1	N.- Historia: 3
Nombre de la Tarea: Creación de scripts utilizando SQL	
Tipo de Tarea: Administrador de Base de Datos	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación del script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas PhpMyAdmin. create table follower_list: Permite crear la tabla “follower_list”, con sus respectivas claves primarias y tipos de datos. create table friend_list: Permite crear la tabla “friend_list”, con sus respectivas claves primarias y tipos de datos.	

2. Creación de funciones PHP para la extracción de datos del perfil y la comunidad de un usuario de Twitter.

TABLA XXVIII. TAREA 2: CREACIÓN DE FUNCIONES PHP PARA LA EXTRACCIÓN DE DATOS DEL PERFIL Y LA COMUNIDAD DE UN USUARIO DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER

N.- Tarea: 2	N.- Historia: 3
Nombre de la Tarea: Creación de funciones PHP para la extracción de datos del perfil y la comunidad de un usuario de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación de las funciones necesarias para la extracción de datos de las tablas usuario, “follower_list” y “friend_list” creadas anteriormente.	
function tw_friends_list(): Permite extraer el JSON con los datos de los seguidores de una cuenta de Twitter.	
function tw_follower_list(): Permite extraer el JSON con los datos de los usuarios a quien se está siguiendo en Twitter.	

3. Diseño de interfaces gráficas Perfil de Usuario y Dashboard.

TABLA XXIX. TAREA 3: DISEÑO DE INTERFACES GRÁFICAS
PERFIL DE USUARIO Y GENERAL DEL CASO DE USO MOSTRAR
ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE
TWITTER

N.- Tarea: 3	N.- Historia: 3
Nombre de la Tarea: Diseño de interfaces gráficas Perfil de Usuario y General.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 16 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas frm_app y frm_perfil_usuario mediante la utilización de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> . frm_app.php: Es la pantalla de inicio donde se muestra información general de una cuenta de Twitter como nuevos seguidores, menciones y tweets por día, estadísticas de la ubicación de los seguidores y siguiendo mediante la utilización de diagramas de barras. frm_perfil_usuario.php: En esta pantalla se muestra toda la información del perfil de una cuenta de Twitter como número de seguidores, siguiendo, tweets, listas; nombre de usuario, screen_name, ubicación fecha de creación y un time_line con los tweets que ha publicado.	

4. Diseño de interfaces gráficas Seguidores y Siguiendo.

TABLA XXX. TAREA 4: DISEÑO DE INTERFACES GRÁFICAS
SEGUIDORES Y SIGUIENDO DEL CASO DE USO MOSTRAR
ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE
TWITTER

N.- Tarea: 4	N.- Historia: 3
Nombre de la Tarea: Diseño de interfaces gráficas Seguidores y Siguiendo.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 16 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas “frm_amigos” y “frm_seguidores” mediante la utilización de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> . frm_amigos.php: En esta pantalla se muestra toda la información de los seguidores de una cuenta de Twitter. frm_seguidores.php: En esta pantalla se muestra toda la información de los usuarios a los que estas siguiendo en Twitter.	

5. Creación de funciones PHP para guardar, extraer y mostrar datos del perfil y la comunidad de un usuario de Twitter.

TABLA XXXI. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR, EXTRAER Y MOSTRAR DATOS DEL PERFIL Y LA COMUNIDAD DE UN USUARIO DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER

N.- Tarea: 5	N.- Historia: 2
Nombre de la Tarea: Creación de funciones PHP para guardar, extraer y mostrar datos del perfil y la comunidad de un usuario de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
<p>Descripción:</p> <p>La presente tarea describe el proceso de creación de funciones <i>PHP</i>, que nos permitirá guardar la información de datos del perfil y la comunidad (seguidores y siguiendo) de un usuario de Twitter.</p> <p>Todos los métodos de guardar datos verifican si los datos a guardar ya existen en la base de datos, esto lo realiza mediante el id del usuario autenticado. Si no hay datos relacionados con ese id se guardarán los datos, caso contrario no realizará dicha acción.</p> <p><i>function user_tw():</i> Función para guardar User_Time_line o tweets que un usuario a publicado en Twitter.</p> <p><i>Function extrae_utl():</i> extrae los datos del User_Time_line mediante la utilización de sesiones rápidas (set_flashdata).</p> <p><i>mostrar_utl():</i> muestra los tweets del User_Time_Line en pantalla.</p> <p><i>function guarda_friend_list():</i> Función para guardar los seguidores de una cuenta de Twitter.</p> <p><i>Function mostrar_amigos():</i> muestra lista de amigos en la pantalla “frm_amigos.php”.</p> <p><i>function guarda_follower_list():</i> Función para guardar los usuarios que estas siguiendo en Twitter.</p> <p><i>Function mostrar_seguidores():</i> muestra lista de siguiendo en la pantalla</p>	

“frm_seguidores”.

Function extrae_nuevos_seguidores(): extrae los seguidores que se han agregado recientemente a la base de datos.

Function mostrar_nuevos_seg(): muestra lista de nuevos amigos en la pantalla “frm_app.php”.

Function estadistica_tw_dia(): muestra el número de tweets publicados por día en la pantalla “frm_app.php”.

Function estadistica_seg_dia(): muestra el número de seguidores por día en la pantalla “frm_app.php”.

Function extrae_zone_friend(): muestra estadística de la zona en donde se localizan los seguidores de un usuario de Twitter.

Function extrae_zone_follower(): muestra estadística de la zona en donde se localizan los usuarios siguiendo de un usuario de Twitter.

- ✓ **Historia 4:** Mostrar tweets (Favoritos, de usuario, de su comunidad, de Alertas) de una cuenta de Twitter

**TABLA XXXII. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
MOSTRAR TWEETS DE UNA CUENTA DE TWITTER**

Historia:4	Mostrar Lista de tweets (Favoritos, Comunidad, Usuario, Alertas) de una cuenta de Twitter	
Número	Nombre	Tiempo Estimado
1	Creación de scripts utilizando SQL.	8
2	Creación de funciones PHP para la extracción de datos de tweets favoritos, de usuario y de la comunidad de un usuario de Twitter.	60
3	Diseño de interfaces gráficas Tus tweets Favoritos y Tus tweets	16
4	Diseño de interfaces gráficas Tweets de tu comunidad y Tweets de tus Alertas	16
5	Creación de funciones PHP para guardar, extraer y mostrar datos de tweets de tu comunidad y tus tweets Favoritos.	60
		160 Horas

1. Creación de scripts utilizando SQL

TABLA XXXIII. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER

N.- Tarea: 1	N.- Historia: 4
Nombre de la Tarea: Creación de scripts utilizando SQL	
Tipo de Tarea: Administrador de Base de Datos	Tiempo Estimado: 8 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación del script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> . create table tweet: Permite crear la tabla tweet, con sus respectivas claves primarias y tipos de datos. create table user_time_line: Permite crear la tabla “user_time_line”, con sus respectivas claves primarias y tipos de datos. create table favorito: Permite crear la tabla “favorito”, con sus respectivas claves primarias y tipos de datos.	

1. Creación de funciones PHP para la extracción de datos de favoritos, tweets de usuario y tweets de la comunidad de un usuario de Twitter.

TABLA XXXIV. TAREA 2: CREACIÓN DE FUNCIONES PHP PARA LA EXTRACCIÓN DE DATOS DE TWEETS FAVORITOS, DE USUARIO Y DE LA COMUNIDAD DE UN USUARO DE TWITTER DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER

N.- Tarea: 2	N.- Historia: 4
Nombre de la Tarea: Creación de funciones PHP para la extracción de datos de Tweets favoritos, de usuario y de la comunidad de un usuario de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 60 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación de las funciones necesarias para la extracción de datos de las tablas usuario, follower_list y friend_list creadas anteriormente. function user_time_line(): Permite extraer el JSON con la data de los tweets publicados por un usuario de Twitter. function home_time_line(): Permite extraer el JSON con la data de los tweets publicados por los seguidores y siguiendo de un usuario de Twitter. function tw_favoritos(): Permite extraer el JSON con la data de los tweets a los que un usuario de Twitter ha seleccionado como favoritos.	

2. Diseño de interfaces gráficas Tus Tweets Favoritos y Tus Tweets.

TABLA XXXV. TAREA 3: DISEÑO DE INTERFACES GRÁFICAS TUS TWEETS FAVORITOS Y TUS TWEETS DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER

N.- Tarea: 3	N.- Historia: 4
Nombre de la Tarea: Diseño de interfaces gráficas Tus Tweets Favoritos y Tus tweets.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 16 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas frm_time_line_favoritos y frm_user_time_line mediante la utilización de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> . frm_time_line_favoritos.php: En esta pantalla se muestra los tweets seleccionados por un usuario de Twitter como favoritos. frm_user_time_line.php: En esta pantalla se muestra los tweets publicados por un usuario de Twitter.	

3. Diseño de interfaces gráficas Tweets de tu comunidad y Tweets de tus Alertas.

TABLA XXXVI. TAREA 4: DISEÑO DE INTERFACES GRÁFICAS
TWEETS DE TU COMUNIDAD Y TWEETS DE TUS ALERTAS DEL
CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER

N.- Tarea: 4	N.- Historia: 4
Nombre de la Tarea: Diseño de interfaces gráficas Tweets de tu comunidad y Tweets de tus Alertas.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 16 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas para mostrar el Home-Time-Line y los tweets que pertenecen a una alerta mediante la utilización de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> . frm_home_time_line.php: En esta pantalla se muestra los tweets de la comunidad de una cuenta de Twitter. frm_tweets_alerta.php: En esta pantalla se muestra los tweets relacionados al id de una Alerta.	

4. Creación de funciones PHP para guardar, extraer y mostrar datos de Tweets de tu comunidad y Tus tweets Favoritos.

TABLA XXXVII. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR, EXTRAER Y MOSTRAR DATOS DE TWEETS DE LA COMUNIDAD Y FAVORITOS DEL CASO DE USO MOSTRAR TWEETS DE UNA CUENTA DE TWITTER

N.- Tarea: 5	N.- Historia: 4
Nombre de la Tarea: Creación de funciones PHP para guardar, extraer y mostrar datos de la Comunidad y Tus Tweets Favoritos de una cuenta de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 60 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones <i>PHP</i> , que nos permitirá guardar los tweets de la comunidad (seguidores y siguiendo) y tweets favoritos de un usuario de Twitter. Todos los métodos de guardar datos verifican si los datos a guardar ya existen en la base de datos, esto lo realiza mediante el id del usuario autenticado. Si no hay datos relacionados con ese id se guardarán los datos, caso contrario no realizará dicha acción. function guardar_tw(): guarda Home_Time_line o tweets que un usuario a publicado en Twitter. function extrae_htl(): extrae los datos de Home_Time_line mediante la utilización de sesiones rápidas (set_flashdata). function mostrar_htl(): permite mostrar los tweets del Home_Time_Line en pantalla. function guardar_favorites_list(): guarda los tweets marcados por un usuario de Twitter como favoritos. function extrae_favorito(): extrae los datos del Home_Time_line de favoritos mediante	

la utilización de sesiones rápidas (set_flashdata).

Function mostrar_favorito(): permite mostrar los tweets del Home_Time_Line de favoritos en pantalla.

3.1.3. Pruebas Iteración I

1) Especificación de prueba: Control y acceso al sistema

Historia de Usuario: 1

Versión <1.0>

Historial de Revisiones

TABLA XXXVIII. HISTORIAL DE REVISIONES ITERACIÓN I
VERSIÓN 1.0

Versión	Descripción	Autor
1.0	Revisión	Franklin Michay
1.1	Modificación	Franklin Michay

En esta historia se verifica que los ingresos de los datos hacia el sistema cumplan con las especificaciones establecidas en la historia 1, previo al acceso hacia las pantallas de login y análisis estadístico de una cuenta de Twitter. Si los datos ingresados no coinciden con los de la base de datos se imprimen alertas informando la situación actual de la operación. Para poder loguearse con Twitter un usuario deberá tener una cuenta activa.

Registro de información correcta

✓ Descripción:

Los valores introducidos por el usuario deben cumplir con las reglas establecidas, para acceder hacia el sistema, por el momento no dispondrá de pantallas de administración.

✓ Condiciones de ejecución:

El usuario deberá estar previamente creado, para poder registrarse y a acceder hacia el sistema.

✓ Entrada:

El usuario introducirá su usuario y contraseña

✓ Resultado esperado:

Tras la introducción de datos, si el usuario procesado ha ingresado sus datos correctos, accede hacia el sistema.

✓ Evaluación de la prueba:

Prueba satisfactoria.

Registro de información incorrecta

✓ Descripción:

Si los datos ingresados por el usuario creado no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados. Al intentar ingresar sin haber introducido el usuario y contraseña se nos generará una alerta.

✓ Condiciones de ejecución:

El usuario deberá estar previamente creado, para poder registrarse y a acceder hacia el sistema.

✓ Entrada:

El usuario introducirá su usuario y contraseña
<p>✓ Resultado esperado: El formulario <i>login</i>, imprime las alertas respectivas en caso de que el usuario ingrese, campos que no concuerden.</p>
<p>✓ Evaluación de la prueba: Prueba satisfactoria.</p>

2) Especificación de prueba: Control y acceso a una cuenta de Twitter

Historia de Usuario: 2

Versión <1.1>

Historial de Revisiones

TABLA XXXIX. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN
1.1

Versión	Descripción	Autor
1.1	Revisión	Franklin Michay
1.2	Modificación	Franklin Michay

En esta historia se verifica que los ingresos de los datos hacia el API de Twitter cumplan con las especificaciones establecidas en la historia 2. Los usuarios deben disponer de una cuenta en Twitter para poder loguearse. El usuario deberá ingresar su usuario y contraseña para acceder y obtener su información. Si los datos ingresados no coinciden con los de la cuenta de Twitter, su API lanzará una alerta informando la situación de la operación.

Registro de información correcta

✓ **Descripción:**

Los valores introducidos por el usuario deben cumplir con las reglas establecidas, para acceder a una cuenta de Twitter.

✓ **Condiciones de ejecución:**

El usuario deberá disponer de una cuenta de Twitter activa.

✓ **Entrada:**

El usuario introducirá su *usuario* y *contraseña* de la cuenta de Twitter

✓ **Resultado esperado:**

Tras la introducción de datos, si el usuario procesado ha ingresado sus datos correctos, accede hacia el sistema.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

Registro de información incorrecta

✓ **Descripción:**

Si los datos ingresados por el usuario creado no concuerdan, se imprimirán alertas definidas por el API de Twitter, que indiquen el estado de los campos ingresados.

✓ **Condiciones de ejecución:**

El usuario deberá haber creado previamente una cuenta de Twitter para poder registrarse y a acceder hacia el sistema.

✓ **Entrada:**

El usuario introducirá su usuario y contraseña.

✓ **Resultado esperado:**

El formulario *login* de Twitter, imprime las alertas respectivas en caso de que el usuario ingrese, campos que no concuerden.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

3) Especificación de prueba: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de twitter

Historia de Usuario: 3

Versión <1.2>

Historial de Revisiones

TABLA XL. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN 1.2

Versión	Descripción	Autor
1.2	Revisión	Franklin Michay
1.3	Modificación	Franklin Michay

En esta historia se verifica que los datos y estadísticas que se muestran en las pantallas cumplan con las especificaciones establecidas en la historia 3. Los usuarios podrán seguir la actividad de una cuenta de Twitter mediante la visualización de estadísticas. Se mostrará información del perfil, seguidores y siguiendo de una cuenta de Twitter.

Registro de información correcta

✓ Descripción:

Los datos mostrados deben describir la actividad de una cuenta de Twitter.

✓ Condiciones de ejecución:

El usuario deberá ingresar a la pantalla Dashboard para visualizar estadísticas de la comunidad de una cuenta Twitter. Para mostrar información general de una cuenta deberá ingresar a la pantalla Perfil de usuario y la visualización de los seguidores y siguiendo con la información general de cada usuario se mostrará en las pantallas Seguidores y Siguiendo respectivamente.

✓ Entrada:

El usuario iniciará sesión con un usuario normal y posteriormente con un usuario de una cuenta de Twitter.

✓ Resultado esperado:

Tras la navegación por las pantallas de la aplicación se mostrarán los datos y la estadística de la cuenta de Twitter.

✓ Evaluación de la prueba:

Prueba satisfactoria.

Registro de información incorrecta

✓ Descripción:

Si no se muestran datos al acceder a las pantallas el usuario deberá volver a iniciar sesión en la aplicación

✓ Condiciones de ejecución:

El usuario deberá iniciar sesión con un usuario normal y un usuario de una cuenta de Twitter.

✓ **Entrada:**

El usuario iniciará sesión con un usuario normal y posteriormente con un usuario de una cuenta de Twitter.

✓ **Resultado esperado:**

La navegación por las pantallas permitirá la visualización de los datos y la estadística de una cuenta de Twitter, en caso contrario se imprime las alertas respectivas.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

4) Especificación de prueba: Mostrar Time-line de tweets (Favoritos, Time-line de usuario, Home-time-line, Time-line de Alertas) de una cuenta de Twitter

Historia de Usuario: 4

Versión <1.3>

Historial de Revisiones

TABLA XLI. HISTORIAL DE REVISIONES ITERACIÓN I VERSIÓN
1.3

Versión	Descripción	Autor
1.3	Revisión	Franklin Michay
1.4	Modificación	Franklin Michay

En esta historia se verifica que los datos que se muestran en las pantallas cumplan con las especificaciones establecidas en la historia 4. Los usuarios podrán visualizar los tweets pertenecientes al home-time-line, time-line de favoritos, user-time-line y time-line de alertas.

Registro de información correcta

✓ Descripción:

En los datos mostrados se deben visualizar los tweets publicados y vinculados a una cuenta de Twitter.

✓ Condiciones de ejecución:

El usuario deberá ingresar a las pantallas Home-Time-Line, Time-Line-Favoritos, User-Time-Line y Time-Line-Alertas.

✓ Entrada:

El usuario iniciará sesión con un usuario normal y posteriormente con un usuario de una cuenta de Twitter.

✓ Resultado esperado:

Tras la navegación por las pantallas de la aplicación se mostrarán los tweets de la cuenta de Twitter.

✓ Evaluación de la prueba:

Prueba satisfactoria.

Registro de información incorrecta

✓ Descripción:

Si no se muestran datos al acceder a las pantallas el usuario deberá volver a iniciar sesión en la aplicación o deberá interactuar más con una cuenta de Twitter.

✓ Condiciones de ejecución:

El usuario deberá iniciar sesión con un usuario normal y un usuario de una cuenta de Twitter.

✓ Entrada:

El usuario iniciará sesión con un usuario normal y posteriormente con un usuario de

una cuenta de Twitter.

✓ **Resultado esperado:**

La navegación por las pantallas permitirá la visualización de los datos y la estadística de una cuenta de Twitter, en caso contrario se imprime las alertas respectivas.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

3.1.4. Resultados de la Iteración I

Se tuvieron inconvenientes al momento de conectar la aplicación a una cuenta de Twitter los cuales fueron solucionados a su debido tiempo, para así, no retrasar tareas posteriores.

✓ **Plan de entrega inicial**

Historia 1: Control y acceso al sistema

Historia 2: Control y acceso a una cuenta de Twitter

Historia 3: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter

Historia 4: Mostrar lista de tweets (Favoritos, de usuario, de la comunidad, de Alertas) de una cuenta de Twitter

Versión del sistema 1.4

✓ **Modificación de requerimientos**

2. Control y acceso al sistema (Se mantiene)
3. Control y acceso a una cuenta de Twitter (Se mantiene)
4. Mostrar estadística e Información general de una cuenta de Twitter (movida a la siguiente iteración)
5. Mostrar lista de tweets de una cuenta de Twitter (Se mantiene)

3.1.5. Demo de la Versión: Iteración I

Historia 1: Control y acceso al sistema (5 días)

Login: El usuario debe ingresar un usuario y contraseña válidos, después seleccionar el botón Ingresar para acceder al sistema.



Figura 14. Ingreso al sistema

Historia 2: Control y acceso a una cuenta de Twitter (5 días)

Login con Twitter: El usuario debe seleccionar el botón (Sign in with Twitter), luego deberá ingresar el usuario y contraseña válidos de una cuenta de Twitter, después seleccionar el botón Ingresar para acceder al sistema.

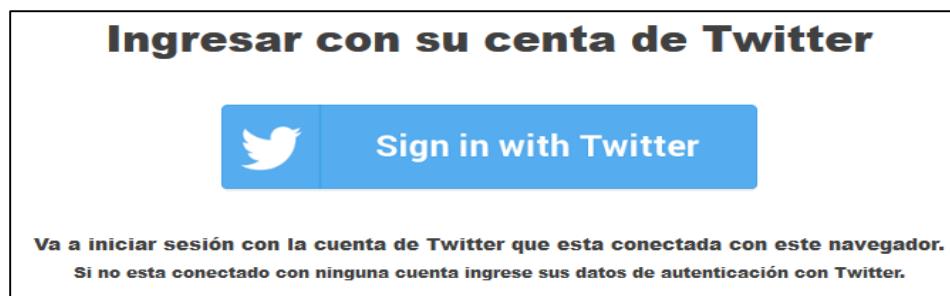


Figura 15. Autenticación con Twitter

¿Autorizas a Análisis Twitter para utilizar tu cuenta?



Análisis Twitter
aplicacioneswebtwitter.com/
Análisis Estadístico Twitter

Usuario o correo electrónico
Contraseña
 Recordar mis datos - [¿Olvidaste tu contraseña?](#)

Iniciar sesión **Cancelar**

Esta aplicación podrá:

- Leer Tweets de tu cronología.
- Ver a quién sigues y seguir a nuevas personas.
- Actualizar tu perfil.
- Publicar Tweets por ti.

No podrá:

- Acceder a tus mensajes directos.
- Ver tu contraseña de Twitter.

Figura 16. Autorización de la aplicación

Historia 3: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter (15 días)

Visualizar Dashboard: El usuario debe seleccionar la pantalla Dashboard para poder visualizar la estadística de una cuenta de Twitter.



Figura 17. Actividad de una cuenta de Twitter

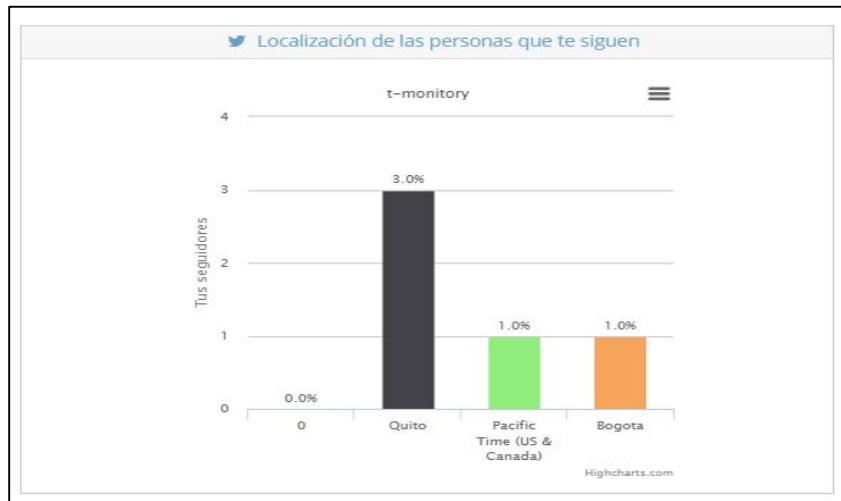


Figura 18. Estadística de seguidores de una cuenta de Twitter

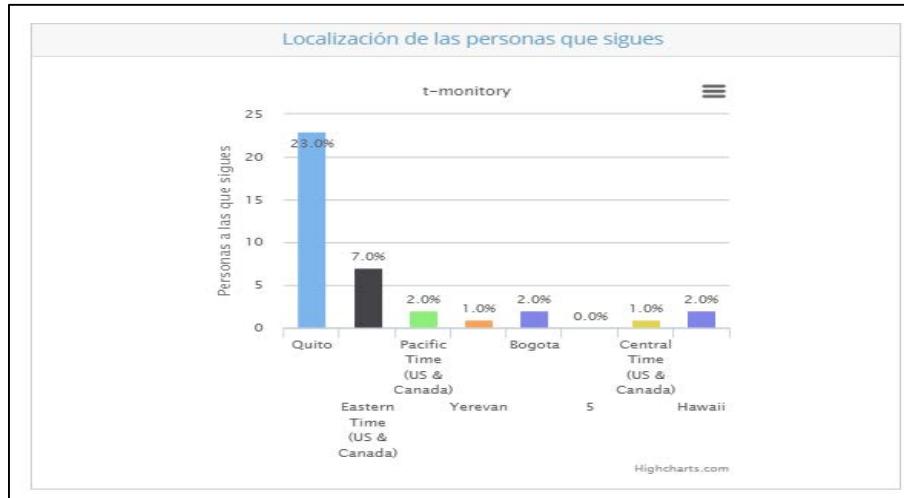


Figura 19. Estadística de siguiendo de una cuenta de Twitter

Visualizar Perfil de Usuario: El usuario debe seleccionar la pantalla Perfil de Usuario del menú que se despliega al seleccionar el botón de Bienvenida del usuario para poder visualizar información general de su cuenta.

The screenshot shows the 'Perfil de Usuario' (User Profile) section of the t-monitory platform. On the left, there is a sidebar menu with options: General, Tu Comunidad, Analítica, Tweetts, and Interacción. The 'General' option is selected. In the main area, there is a profile picture of a child named Franklin Michay. To the right of the picture, there are four colored boxes displaying statistics: 63 Tweets (blue), 44 Seguidores (yellow), 17 Siguiendo (pink), and 22 Favoritos (green). Below these stats is a table with the following information:

Nombre	Franklin Michay
Nombre de Usuario	@FranklinMichay
Ubicación	Loja, Ecuador
Creación	Tue Mar 03 20:10:06 +0000 2015
Descripción	UNL

Figura 20. Información del perfil de una cuenta de Twitter

Visualizar Seguidores: El usuario debe seleccionar la pantalla Seguidores del menú Tu Comunidad para poder visualizar información de cada usuario.

The screenshot shows the 'TU COMUNIDAD' (Your Community) section of the t-monitory platform. On the left, there is a sidebar menu with options: Tu Comunidad, Seguidores, Siguiendo, Analítica, Tweetts, and Interacción. The 'Tu Comunidad' option is selected. In the main area, it says 't-monitory te siguen 6 personas' (t-monitory is followed by 6 people). Below this, there is a table titled 'Seguidores' (Followers) with the following data:

Usuario	Screen Name	Ubicación	Seguidores	Siguiendo	Tweets	Fecha Registro
@angel12	angel1210356892		17	76	0	2016-08-11
Franklin Michay	FranklinMichay	Loja, Ecuador	12	38	54	2016-08-11

Figura 21. Seguidores de una cuenta de Twitter

Visualizar Siguiendo: El usuario debe seleccionar la pantalla Siguiendo del menú Tu Comunidad para poder visualizar información de cada usuario.

Usuario	Screen Name	Ubicación	Seguidores	Siguiendo	Tweets
Paul McCartney	PaulMcCartney	England	2819361	16	3597
Turismo Ecuador	TurismoEc	Ecuador	206719	859	35479

Figura 22. Amigos de una cuenta de Twitter

Historia 4: Mostrar Lista de tweets (Favoritos, de usuario, de la comunidad, Alertas) de una cuenta de twitter (20 días)

Visualizar Tweets de tu Comunidad: El usuario debe seleccionar la pantalla **Tweets de tu Comunidad** menú Tweets para poder visualizar los tweets de su comunidad de Twitter.

Screen Name	Tweet	RT	Favoritos
ECU911_	Descanse tranquilo. Recuerde que trabajamos 24/7 por su seguridad y la de sus seres queridos. #ECU911Conmigo https://t.co/t0cF1YuHbQ	0	0
ActualidadRT	Indocumentada que pagó impuestos en EE.UU. devuelve el golpe https://t.co/6LgkqbSuy2 https://t.co/V56YgyRX01	6	8
ECU911Esmeralda	Buenas noches. Descanse tranquill@ el #ECU911 trabaja por su seguridad. #ECU911Conmigo. https://t.co/sx2MTIm9Xj	0	1
ECU911_	[BOLETÍN] #ECU911 #Ambato coordinó recursos para atención a heridos por accidente de avioneta en #Pastaza... https://t.co/kdl0HD9j5w	4	4

Figura 23. Tweets de tu Comunidad de una cuenta de Twitter

Visualizar Tus Tweets: El usuario debe seleccionar la pantalla Tus Tweets del menú Tweetts para poder visualizar sus tweets.

The screenshot shows a user interface for managing tweets. On the left, there is a sidebar with a tree view under the 'Tweets' category. The 'Tus Tweets' node is selected. The main area is titled 'Lista de tus Tweets'. It includes a search bar and a table with the following columns: Screen Name, Tweet, fecha (date), RT (Retweets), and Favoritos (Favorites). There are four entries in the table:

Screen Name	Tweet	fecha	RT	Favoritos
FranklinMichay	RT @lodelmomentoloj: [TRABAJO] Se requiere contratar asesores comerciales para empresa, los interesados comunicarse al 0992686558. https://...	2017-04-18	2	0
FranklinMichay	prueba programaciòn tweet https://t.co/Fd6PnTgwWF	2017-04-18	0	0
FranklinMichay	prueba publicaciòn tweet https://t.co/b0gnQAkKKN	2017-04-17	0	0
FranklinMichay	Prueba @TTmonitory tweets	2017-04-17	0	0

Figura 24. Tus Tweets de Twitter

Visualizar Tweets de tus Alertas: El usuario debe seleccionar la pantalla Tweets de tus Alertas del menú Tweetts para poder visualizar los tweets que pertenecen a una alerta de Twitter.

The screenshot shows a user interface for managing tweets. On the left, there is a sidebar with a tree view under the 'Tweets' category. The 'Tus Alertas' node is selected. The main area is titled 'Lista de Tweets de tu Alerta: Accidente en Loja 19-01-2017 ecu911Loja'. It includes a search bar and a table with the following columns: Screen Name, Tweet, RT, Favoritos, and Fecha. There are two entries in the table:

Screen Name	Tweet	RT	Favoritos	Fecha
Ecu911Macas	#ECU911Reporta: Huamboya, 200 metros antes del río Shangaime, se coordinó atención en accidente de tránsito con @Salud_CZ6 @PoliciaEcuador	1	0	2017-05-03
ECU911NuevaLoja	#ECU911Reporta Accidente de tránsito en Cascales #ECU911 coordina atención con @PoliciaEcuador y @Salud_Ec https://t.co/QEONaBB9IA	1	1	2017-05-03

At the bottom, it says 'Mostrando registros del 1 al 2 de un total de 2 registros' and has navigation buttons for 'Anterior' and 'Siguiente'.

Figura 25. Tweets de tus Alertas de una cuenta de Twitter

Visualizar Tus Tweets Favoritos: El usuario debe seleccionar la pantalla Tus tweets Favoritos para poder visualizar los tweets que ha seleccionado como favorito.

	Screen Name	Tweet	RT	Favoritos
	ECU911Loja	[ENTREVISTA] Se informa sobre la operatividad del #ECU911 y de las instituciones articuladas en el feriado de Agosto https://t.co/9MaWRKqX8D	3	1913
	diariopinion	Antonio Valencia jugó su partido 250 y es candidato a mejor jugador con el Manchester United https://t.co/WCPRo9KbK4 https://t.co/M7qD4xeLOf	0	6307
	Franklin_Michay	@TTmonitory probando your app	2	23
	Franklin_Michay	@TTmonitory haciendo pruebas	1	23

Figura 26. Tus tweets favoritos de una cuenta de Twitter

3.1.6. Pruebas de Aceptación

Historia 1: Control y acceso al sistema

- ✓ Acceso hacia el sistema seguro, puesto que cuenta con variables de tipo sesión.
- ✓ El formulario Login cuenta con la validación de los campos respectiva.
- ✓ Un usuario luego de haber ingresado su usuario y contraseña, procede hacia el sistema.
- ✓ Si un usuario no está registrado en la base de datos, no tendrá acceso al sistema.

Historia 2: Control y acceso a una cuenta de Twitter

- ✓ Pantalla para ingresar al sistema con una cuenta de Twitter, si permite el acceso.
- ✓ Formulario para ingresar al sistema haciendo uso de una cuenta de Twitter si da acceso al usuario.

Historia 3: Mostrar estadística e información general (perfil, seguidores y siguiendo) de una cuenta de Twitter

- ✓ Pantalla General que permite la visualización de la estadística de la comunidad de una cuenta de Twitter, si muestra los datos.
- ✓ Pantallas Perfil de usuario, Seguidores y Siguiendo que permite la visualización de información del perfil de usuario, seguidores y siguiendo, si muestran datos.

Historia 4: Mostrar Lista de tweets (Favoritos, de usuario, de usuario, de Alertas) de una cuenta de Twitter

- ✓ Pantallas Tweets de tu Comunidad, Tus Tweets, Tweets de tus Alertas y Tus tweets Favoritos que permite la visualización de los tweets de una cuenta de Twitter, si muestra datos.

3.1.7. Incidencias

- ✓ Se rediseñaron las tablas usuario, favorito, follower_list, friend_list, tweet, user_time_line, puesto que inicialmente contaban con pocos campos.
- ✓ Las pantallas cambiaron de aspecto frecuentemente, ya que fueron apareciendo nuevos diseños.
- ✓ La historia 3 ha tenido que ser movida a la siguiente iteración debido a nuevos requerimientos por parte del cliente.

3.2. ITERACIÓN II

3.2.1. Cronograma de Actividades

TABLA XLII. CRONOGRAMA DE ACTIVIDADES ITERACIÓN II

ITERACIÓN	N.-	Nombre Historia	Duración Semanas	Duración Horas
2	05	Mostrar estadística de la actividad de una cuenta de Twitter	6	240
2	03	Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter (modificación)	1	40
				Total=280

3.2.2. Tareas por historia: Iteración II

- ✓ **Historia 5:** Mostrar estadística de la actividad de una cuenta de Twitter

TABLA XLIII. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE
TWITTER

Historia:5	Mostrar estadística de la actividad de una cuenta de Twitter.	
Número	Nombre	Tiempo Estimado
1	Creación de scripts utilizando SQL.	10
2	Creación de funciones PHP para la generación de estadísticas (Alertas, Menciones, RT y Tweets) de una cuenta de Twitter.	70
3	Diseño de interfaces gráficas Estadísticas de tu comunidad y Menciones y RT.	30
4	Diseño de interfaces gráficas Estadísticas de tus alertas y Estadísticas de tus Tweets.	30
5	Creación de funciones PHP para guardar, extraer y mostrar datos estadísticos de una cuenta de Twitter.	100
		240 Horas

1. Creación de scripts utilizando SQL

TABLA XLIV. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL
DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE
UNA CUENTA DE TWITTER

N.- Tarea: 1	N.- Historia: 5
Nombre de la Tarea: Creación de scripts utilizando SQL	
Tipo de Tarea: Administrador de Base de Datos	Tiempo Estimado: 10 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación del script <i>SQL</i> , para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> . create table tweet_rt: Permite crear la tabla “ <i>tweet_rt</i> ” en la cual se va almacenar todos los retuits de un tweet. create table user_retweet: Permite crear la tabla “ <i>user_retweet</i> ” en donde se va a almacenar los ids de los usuarios que retuitean un tweet.	

1. **Creación de funciones PHP para la generación de estadísticas (Alertas, Menciones, RT y Tweets) de una cuenta de Twitter.**

TABLA XLV. TAREA 2: CREACIÓN DE FUNCIONES PHP PARA LA GENERACIÓN DE ESTADÍSTICAS DE UNA CUENTA DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER

N.- Tarea: 2	N.- Historia: 5
Nombre de la Tarea: Creación de funciones PHP para la generación de estadísticas (Alertas, Menciones, RT y Tweets) de una cuenta de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 70 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación de las funciones necesarias para la extracción de datos de las tablas tweet_rt y user_retweet creadas anteriormente. Además, se crearán los métodos necesarios para generar las estadísticas. function busca_statuses_retweets_of_me(): Permite extraer el JSON con la data de los tweets de un usuario que han sido retuiteados por otros. function user_tw_rt(): Permite extraer el JSON con la data de los tweets retuiteados pasándole como parámetro el id del tweet. function estadistica_mt_dia(): Genera estadísticas de las menciones de un usuario de Twitter. function estadistica_rt_dia(): Genera estadísticas por día de los RT de un usuario de Twitter. function estadistica_tw_dia(): Genera estadísticas de tweets de un usuario de Twitter por día. function estadistica_semana(): Genera estadísticas de tweets de un usuario de	

Twitter por semana.

function estadistica_mes(): Genera estadísticas de tweets de un usuario de Twitter por mes.

2. Diseño de interfaces gráficas Estadísticas de tu comunidad y Menciones y RT.

TABLA XLVI. TAREA 3: DISEÑO DE INTERFACES GRÁFICAS ESTADÍSTICAS DE TU COMUNIDAD Y MENCIONES Y RT DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER

N.- Tarea: 3	N.- Historia: 5
Nombre de la Tarea: Diseño de interfaces gráficas Estadísticas de tu comunidad y Menciones y RT.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 30 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas "frm_estadisticas" y "frm_menciones_y_rt" utilizando hojas de estilo CSS y archivos JAVASCRIPT, también se utilizó la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> y la librería Highcharts para visualizar las estadísticas.	
frm_estadisticas.php: En esta pantalla se muestra la lista de usuarios que más retuitean, mencionan y los tweets más retuiteados. Las estadísticas se visualizan mediante diagramas de pasteles y diagramas de barras.	
frm_menciones_y_rt.php: En esta pantalla se muestra las menciones de un usuario de Twitter junto a los botones de contestar mención y seguir o dejar de seguir un usuario.	

3. Diseño de interfaces gráficas Estadísticas de tus alertas y Estadísticas de tus Tweets.

TABLA XLVII. TAREA 4: DISEÑO DE INTERFACES GRÁFICAS ESTADÍSTICAS DE TUS ALERTAS Y TWEETS DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER

N.- Tarea: 4	N.- Historia: 5
Nombre de la Tarea: Diseño de interfaces gráficas Estadísticas de tus alertas y Estadísticas de tus Tweets.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 30 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas para mostrar las estadísticas de los tweets mediante la utilización de hojas de estilo CSS, archivos JAVASCRIPT, la librería <i>Highcharts</i> para la visualización de las estadísticas mediante pasteles y diagramas de barras y la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> . frm_analitica_tweets.php: En esta pantalla se muestra las estadísticas de tweets de acuerdo al día, semana y mes. Además, se incluye estadísticas por rango de fechas.	

4. Creación de funciones PHP para guardar, extraer y mostrar datos estadísticos de una cuenta de Twitter.

TABLA XLVIII. TAREA 5: CREACIÓN DE FUNCIONES PHP PARA GUARDAR, EXTRAER Y MOSTRAR DATOS ESTADÍSTICOS DE TWITTER DEL CASO DE USO MOSTRAR ESTADÍSTICA DE LA ACTIVIDAD DE UNA CUENTA DE TWITTER

N.- Tarea: 5	N.- Historia: 5
Nombre de la Tarea: Creación de funciones PHP para guardar, extraer y mostrar datos estadísticos de una cuenta de Twitter.	
Tipo de Tarea: Programación	Tiempo Estimado: 60 Horas
Programador Responsable: Franklin Michay	
<p>Descripción:</p> <p>La presente tarea describe el proceso de creación de funciones <i>PHP</i>, que nos permitirá guardar los tweets de la comunidad (seguidores y siguiendo) y tweets favoritos de un usuario de Twitter.</p> <p>Todos los métodos de guardar datos verifican si los datos a guardar ya existen en la base de datos, esto lo realiza mediante el id del usuario autenticado. Si no hay datos relacionados con ese id se guardarán los datos, caso contrario no realizará dicha acción.</p> <p>function tw_rangoo_mt(): Consulta en base de datos las menciones de un usuario de acuerdo a la fecha en que se publicaron en Twitter.</p> <p>function tw_rangoo_seg(): Consulta en base de datos los seguidores de un usuario de acuerdo a la fecha en que se generaron en Twitter.</p> <p>function tw_rangoo_rt(): Consulta en base de datos los RT de un usuario de acuerdo a la fecha en que se publicaron en Twitter.</p> <p>function tw_rangoo_tw(): Consulta en base de datos los tweets de un usuario de acuerdo a la fecha en que se publicaron en Twitter.</p> <p>function tw_rango (): Consulta en base de datos los tweets de un usuario de acuerdo a un rango de fechas en que se publicaron en Twitter.</p> <p>Function tw_rango_retweet (): Consulta en base de datos los ids de tweets de un usuario de acuerdo a un rango de fechas en que se publicaron en Twitter.</p>	

- ✓ **Historia 3:** Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter.

**TABLA XLIX. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA
CUENTA DE TWITTER**

Historia:3	Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter	
Número	Nombre	Tiempo Estimado
1	Rediseño de interfaz gráfica General.	16
2	Diseño de interfaces gráficas Publicar y Programar tweet.	24
3	Creación de funciones PHP para guardar datos de tweets programados.	40
		80 Horas

1. Rediseño de interfaz gráfica General

TABLA L. TAREA 1: REDISEÑO DE LA INTERFÁZ GRÁFICA
GENERAL DEL CASO DE USO MOSTRAR ESTADÍSTICA E
INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER

N.- Tarea: 1	N.- Historia: 3
Nombre de la Tarea: Rediseño de interfaz gráfica General	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 16 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de rediseño de la pantalla frm_app mediante la utilización de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> . frm_app.php: Es la pantalla de inicio donde se muestra información general de una cuenta de Twitter como nuevos seguidores, menciones y tweets por día y estadísticas de la ubicación de los seguidores y siguiendo mediante la utilización de diagramas de barras. Además de agregarán dos formularios para publicar y programar un tweet, así mismo, se mostrarán el TimeLine de Favoritos y de usuario.	

2. Diseño de formularios Publicar y Programar tweets

TABLA LI. TAREA 2: DISEÑO DE FORMULARIOS PUBLICAR Y PROGRAMAR TWEETS DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER

N.- Tarea: 2	N.- Historia: 3
Nombre de la Tarea: Diseño de formularios Publicar y Programar tweet.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 24 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño de creación de los formularios para la publicar y programar un tweet mediante la utilización de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> .	

3. Creación de funciones PHP para guardar datos de tweets programados.

TABLA LII. TAREA 3: CREACIÓN DE FUNCIONES PHP PARA GUARDAR DATOS DE TWEETS PROGRAMADOS DEL CASO DE USO MOSTRAR ESTADÍSTICA E INFORMACIÓN GENERAL DE UNA CUENTA DE TWITTER

N.- Tarea: 3	N.- Historia: 3
Nombre de la Tarea: Creación de funciones PHP para guardar datos de tweets programados.	
Tipo de Tarea: Programación	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones <i>PHP</i> , que nos permitirá guardar los tweets programados por un usuario de Twitter function programer_tweet (): Función para programar los tweets de acuerdo a una fecha y hora. Function upload_file(): convierte la imagen seleccionada por el usuario a un identificador reconocible por el API de Twitter.	

3.2.3. Pruebas Iteración II

1) Especificación de prueba: Mostrar estadística de la actividad de una cuenta de Twitter

Historia de Usuario: 5

Versión <1.4>

Historial de Revisiones

TABLA LIII. HISTORIAL DE REVISIONES ITERACIÓN II VERSIÓN 1.4

Versión	Descripción	Autor
1.4	Revisión	Franklin Michay
1.5	Modificación	Franklin Michay

En esta historia se verifica que las estadísticas que se muestran en las pantallas cumplan con las especificaciones establecidas en la historia 5. Los usuarios podrán seguir la actividad de los usuarios una cuenta de Twitter mediante la visualización de estadísticas. Se mostrará información de los usuarios que retuitean, mencionan y los tweets más relevantes de una cuenta de Twitter.

Registro de información correcta

✓ Descripción:

Los valores mostrados en las estadísticas deberán reflejar datos reales de la cuenta monitoreada.

✓ Condiciones de ejecución:

El usuario deberá estar previamente creado, para poder registrarse y acceder hacia el sistema.

✓ **Entrada:**

Seleccionar un rango de fechas.

✓ **Resultado esperado:**

Tras la selección de las fechas, si existen datos dentro de ese rango se mostrarán las estadísticas.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

Registro de información incorrecta

✓ **Descripción:**

Si no existen datos en el rango de fechas seleccionados no se mostrarán las estadísticas.

✓ **Condiciones de ejecución:**

El usuario deberá estar previamente creado, para poder registrarse y acceder hacia el sistema.

✓ **Entrada:**

Seleccionar un rango de fechas.

✓ **Resultado esperado:**

No se mostrarán estadísticas en caso de que no existan datos en el rango de fechas seleccionado.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

2) Especificación de prueba: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter.

Historia de Usuario: 3

Versión <1.5>

Historial de Revisiones

TABLA LIV. HISTORIAL DE REVISIONES ITERACIÓN II VERSIÓN
1.5

Versión	Descripción	Autor
1.5	Revisión	Franklin Michay
1.6	Modificación	Franklin Michay

En esta historia se verifica que los datos, estadísticas y formularios para programar y publicar tweets que se muestran en las pantallas cumplan con las especificaciones establecidas en las historias 3 y 6. Los usuarios podrán seguir la actividad de una cuenta de Twitter mediante la visualización de estadísticas. Se mostrará información del perfil, Lista de tweets (favoritos y de usuario), seguidores y siguiendo de una cuenta de Twitter.

Registro de información correcta

✓ **Descripción:**

Los datos mostrados deben describir la actividad de una cuenta de Twitter.

✓ **Condiciones de ejecución:**

El usuario deberá ingresar a la pantalla General para visualizar estadísticas de la comunidad de una cuenta Twitter. Para mostrar información general de una cuenta deberá ingresar a la pantalla Perfil de usuario y la visualización de los seguidores y siguiendo con la información general de cada usuario se mostrará en las pantallas Seguidores y Siguiendo respectivamente.

✓ en la pantalla General el usuario deberá seleccionar los botones publicar tweet y/o

programar tweet los cuales mostrarán una pantalla emergente donde se tendrá que ingresar la información requerida.

✓ **Entrada:**

El usuario iniciará sesión con un usuario normal y posteriormente con un usuario de una cuenta de Twitter.

✓ **Resultado esperado:**

Tras la navegación por las pantallas de la aplicación se mostrarán los datos y la estadística de la cuenta de Twitter. Además se publicarán y se programarán los tweets.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

Registro de información incorrecta

✓ **Descripción:**

Si no se muestran datos al acceder a las pantallas el usuario deberá volver a iniciar sesión en la aplicación

✓ **Condiciones de ejecución:**

El usuario deberá iniciar sesión con un usuario normal y un usuario de una cuenta de Twitter.

✓ **Entrada:**

El usuario deberá ingresar su usuario y contraseña y posteriormente tras la redirección del API de Twitter deberá loguearse con un usuario de Twitter.

✓ **Resultado esperado:**

La navegación por las pantallas permitirá la visualización de los datos y la estadística de una cuenta de Twitter, en caso contrario se imprime las alertas respectivas.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

3.2.4. Resultados de la Iteración II

- ✓ **Plan de entrega inicial**

Historia 5: Mostrar estadística de la actividad de una cuenta de Twitter.

Historia 3: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter (modificada).

Versión del sistema 1.6

- ✓ **Modificación de requerimientos**

1. Mostrar estadística de la actividad de una cuenta de Twitter (Se mantiene)
2. Mostrar estadística e Información general de una cuenta de twitter (Se mantiene)

3.2.5. Demo de la Versión: Iteración II

Historia 5: Mostrar estadística de la actividad de una cuenta de twitter (30 días)

Estadísticas de tu comunidad: El usuario debe ingresar al submenú **Estadísticas de tu Comunidad** de la pantalla **Analítica**, en esta pantalla se mostrarán estadísticas de los usuarios que más retuitean y mencionan, así mismo se mostrarán estadísticas de los tweets más relevantes.

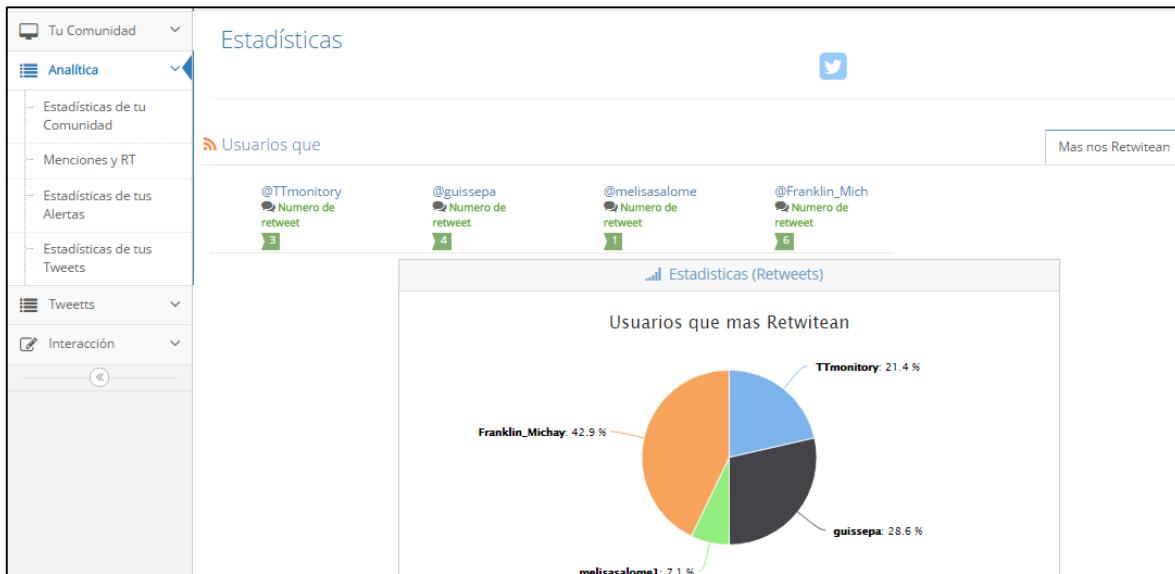


Figura 27. Estadísticas de usuarios que más retuitean una cuenta de Twitter

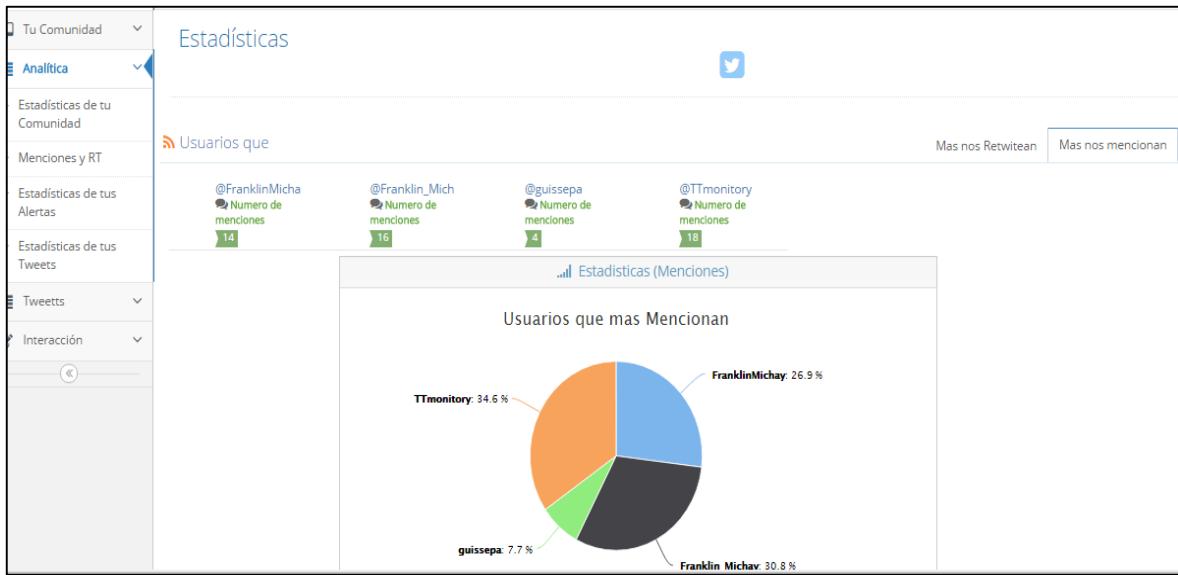


Figura 28. Estadística de menciones de una cuenta de Twitter

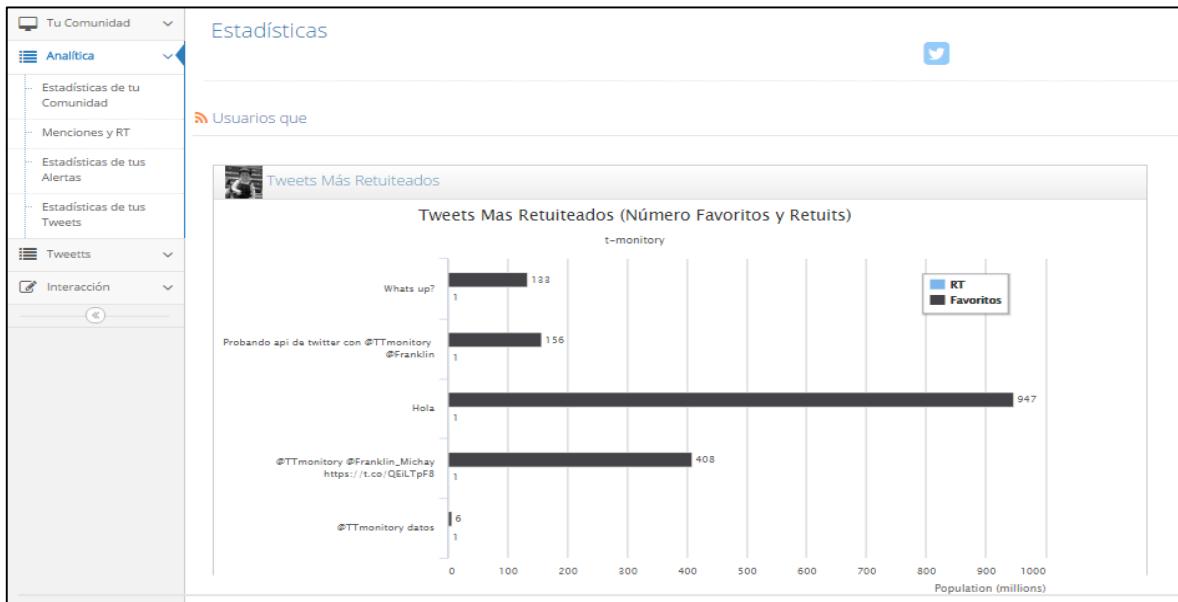


Figura 29. Estadísticas de tweets más retuiteados de una cuenta de Twitter

Menciones y RT: El usuario debe ingresar al submenú **Menciones y RT** de la pantalla **Analítica**, en esta pantalla se mostrarán las menciones y RT de una cuenta de Twitter.

The screenshot shows the 'Menciones y RT' (Mentions and RT) section of the Twitter Analytics dashboard. On the left, there's a sidebar with navigation links like Dashboard, Tu Comunidad, Analítica (which is currently selected), Estadísticas de tu Comunidad, Menciones y RT (selected), Estadísticas de tus Alertas, Estadísticas de tus Tweets, Tweets, and Interacción. The main area has a header 'TUS MENCIONES' with a Twitter logo. Below it, a message says 'Puedes responder o seguir a los usuarios que te hayan mencionado'. A table lists three mentions:

Screen Name	Tweet	RT	Favorites	Fecha de creación	Acciones
FranklinMichay	@FranklinMichay probando audience contestame	0	9	2015-03-03	Tweet to @FranklinMichay Follow @FranklinMichay
Franklin_Michay	@FranklinMichay mencioname	0	4	2011-03-10	Tweet to @Franklin_Michay Follow @Franklin_Michay
guissepa	Twiter @FranklinMichay	0	428	2015-01-04	Tweet to @guissepa Follow @guissepa

Figura 30. Mencionar y/o seguir una cuenta de Twitter

Estadísticas de tus Alertas: El usuario debe ingresar al submenú **Estadísticas de tus Alertas** de la pantalla **Analítica**, en esta pantalla se mostrarán estadísticas de las alertas creadas por un usuario.

The screenshot shows the 'Analítica' (Analytics) section of the Twitter dashboard. Under 'Alertas' (Alerts), it displays a list of alerts with their tweet counts:

Nombre Alerta	Número Tweets	Acciones
Accidente	125	Estadística
Transito	90	Estadística
Zamora	425	Estadística
asdasd	6	Estadística
Bolivia vs Ecuador	411	Estadística

Figura 31. Alertas de una cuenta de Twitter

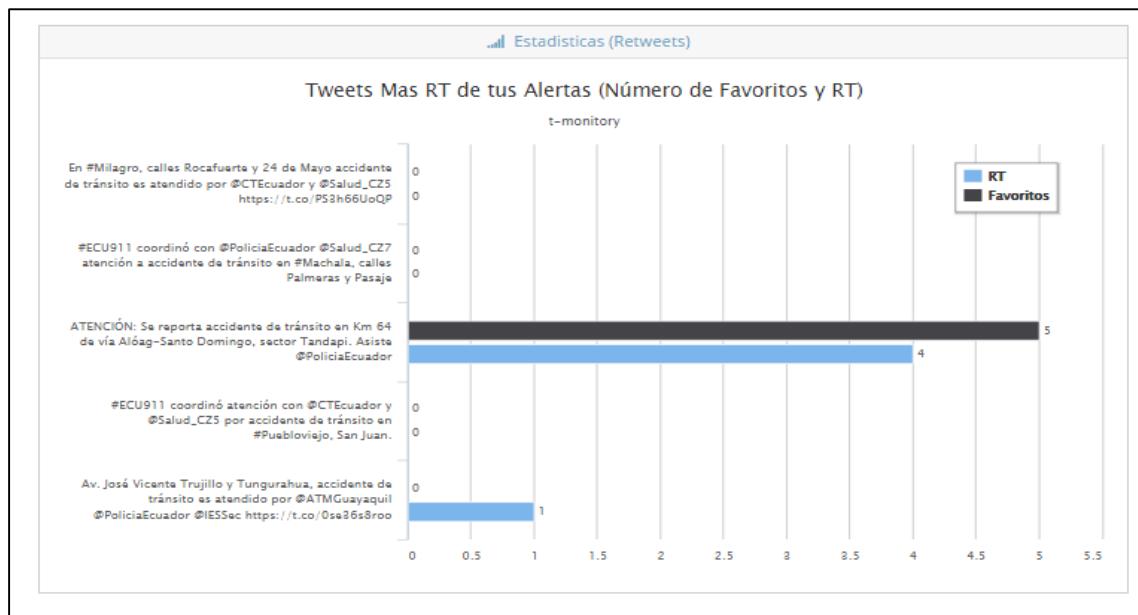


Figura 32.Tweets más retuiteados de alertas de una cuenta de Twitter

Estadísticas de tus Tweets: El usuario debe ingresar al submenú **Estadísticas de tus Tweets** de la pantalla **Analítica**, en esta pantalla se mostrarán estadísticas de los tweets de una cuenta de Twitter.

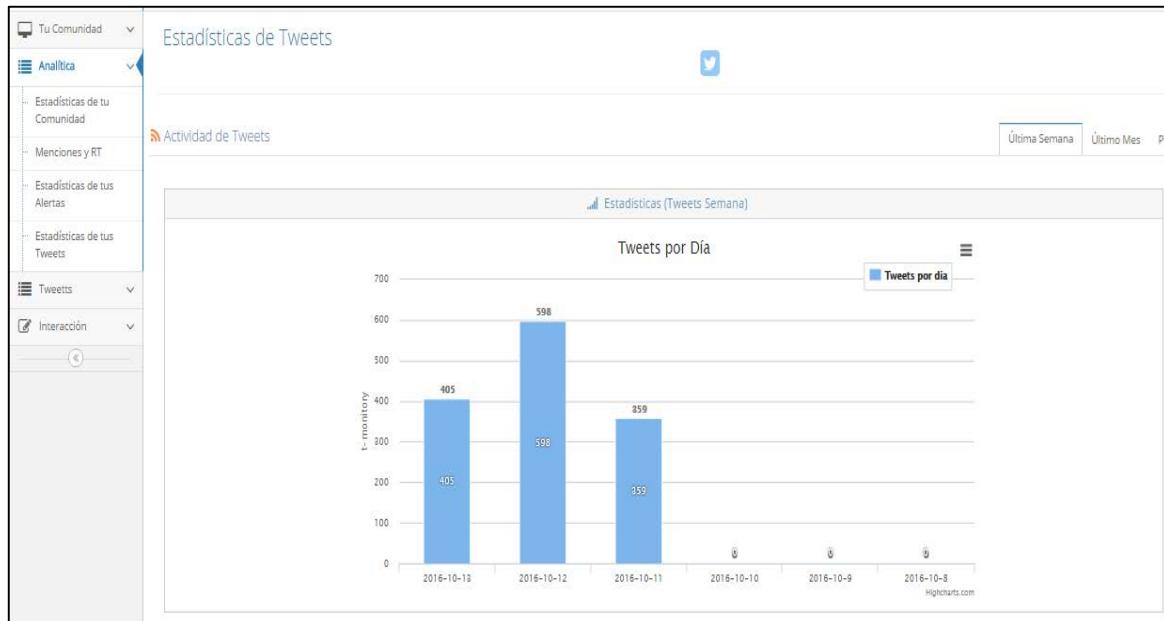


Figura 33. Estadística de tweets de una cuenta de Twitter (por día)

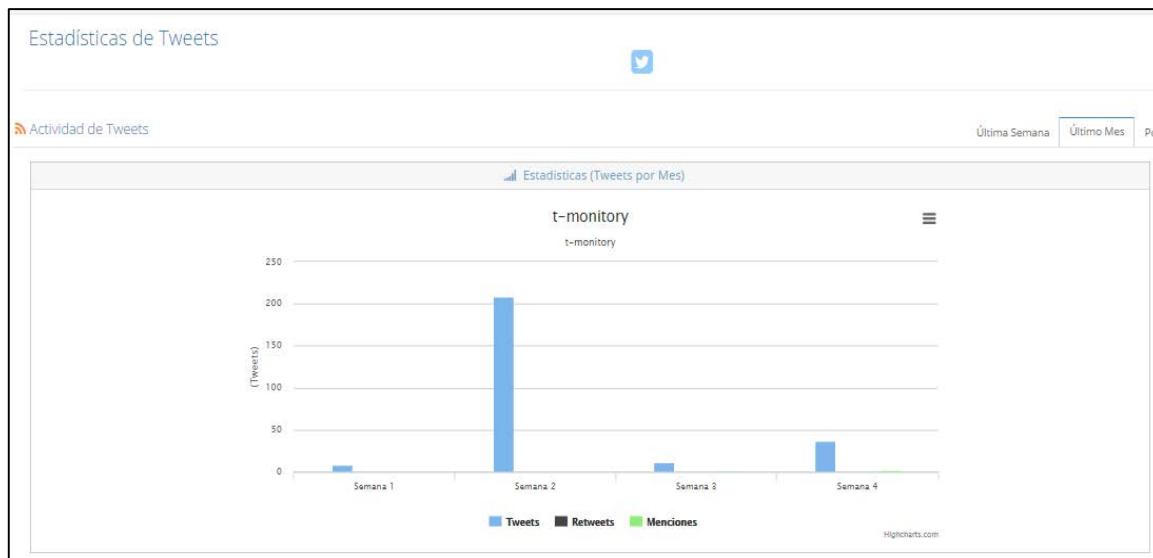


Figura 34. Estadística de tweets de una cuenta de Twitter (por mes)

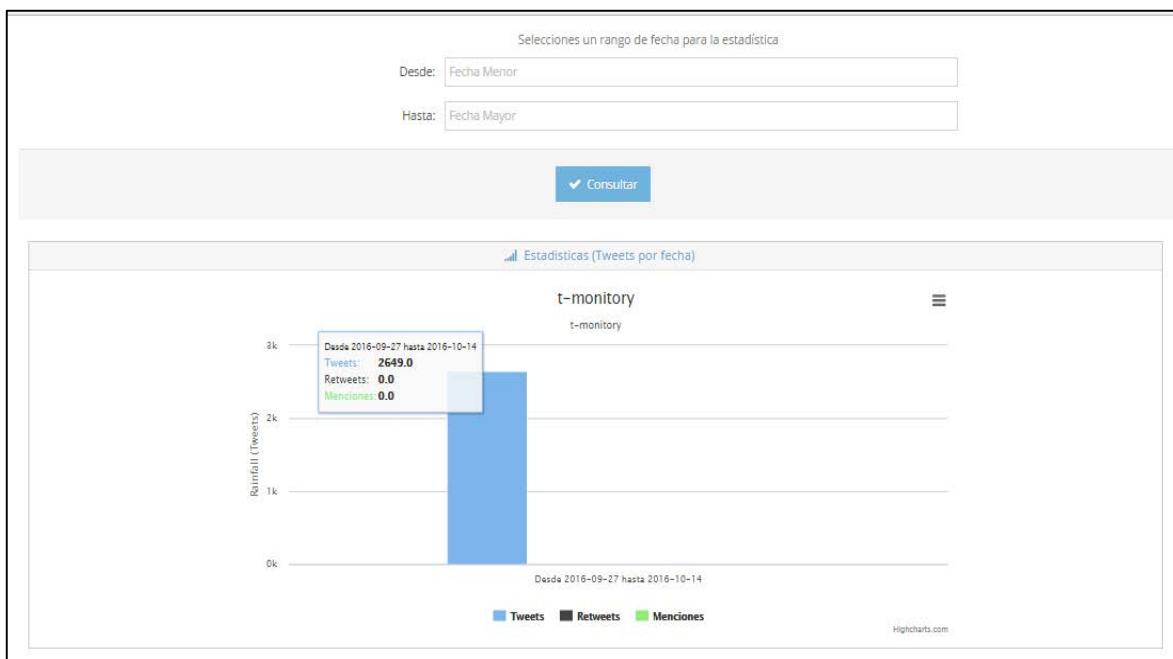


Figura 35. Estadística de tweets de una cuenta de Twitter (por selección de rango de fechas)

Historia 3: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter (modificada) (5 días)

Visualizar Dashboard: El usuario debe seleccionar la pantalla Dashboard para poder visualizar la estadística de una cuenta de Twitter.

Además, podrá publicar y programar tweets mediante la selección de los botones publicar y programar tweet que se muestran en la pantalla principal.



Figura 36. Actividad de una cuenta de Twitter (rediseñada)

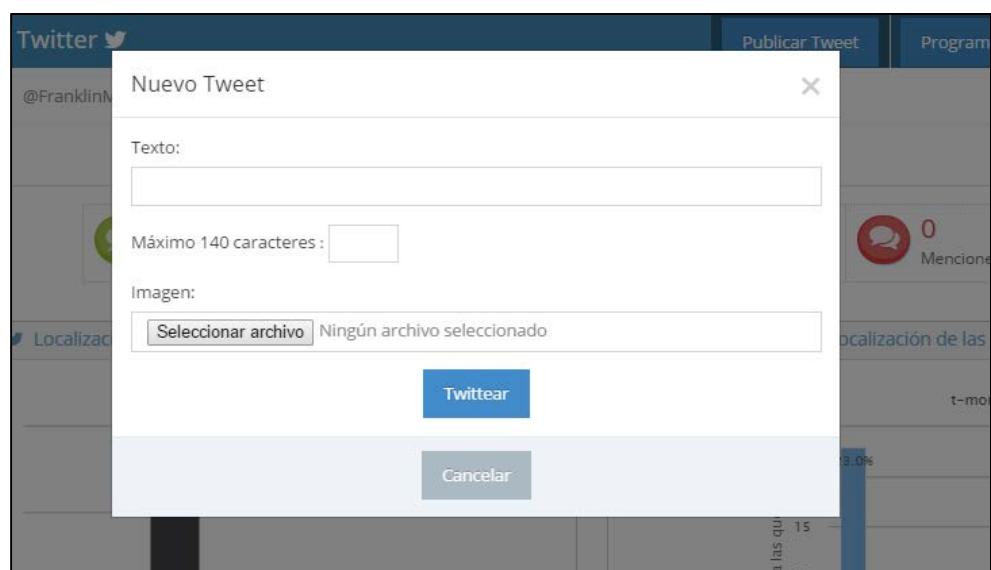


Figura 37. Publicación de tweet

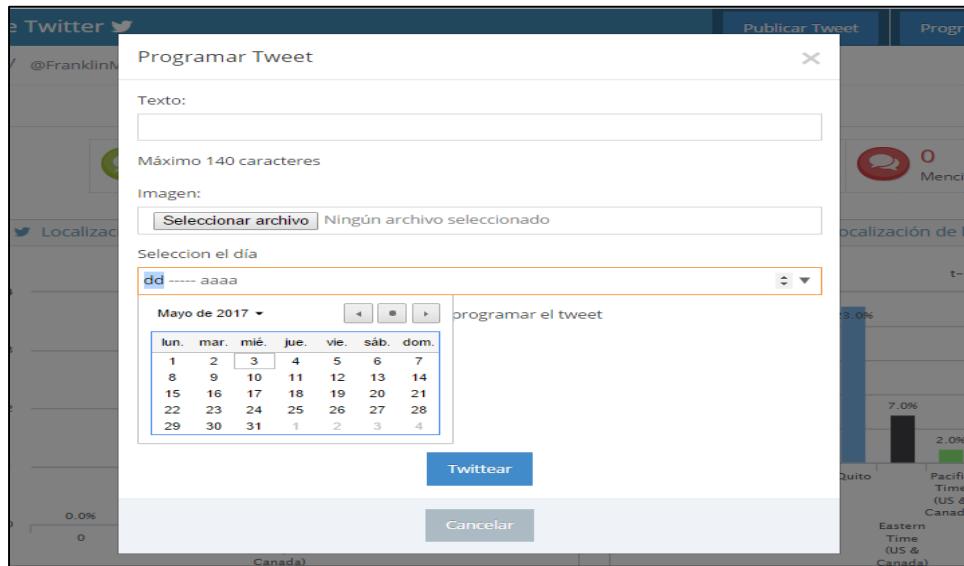


Figura 38. Programación de tweet

3.2.6. Pruebas de Aceptación

Historia 5: Mostrar estadística de la actividad de una cuenta de twitter

- ✓ Se muestran las estadísticas
- ✓ Se muestran las menciones.
- ✓ Se puede mencionar y/o seguir un usuario de Twitter.

Historia 3: Mostrar estadística e Información general (Perfil, Seguidores y Siguiendo) de una cuenta de Twitter (modificada)

- ✓ La publicación y programación tweets funciona.
- ✓ Se visualiza las estadísticas de la cuenta de Twitter.

3.2.7. Incidencias

- ✓ Se aumentó nuevas funcionalidades a las vistas puesto que se aumentaron nuevos requerimientos.

3.3. ITERACIÓN III

3.3.1. Cronograma de Actividades

TABLA LV. CRONOGRAMA DE ACTIVIDADES ITERACIÓN III

ITERAC IÓN	N.-	Nombre Historia	Duración Semanas	Duración Horas
3	06	Publicar y programar tweets	3	64
3	07	Administrar Alertas	3	60
3	08	Administrar Usuarios	3	60
				Total=184

3.3.2. Tareas por historia: Iteración III

- ✓ **Historia 6: Publicar y programar tweets**

**TABLA LVI. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
PUBLICAR Y PROGRAMAR TWEETS**

Historia:6	Publicar y programar tweets	
Número	Nombre	Tiempo Estimado
1	Creación de scripts utilizando SQL.	10
2	Diseño de interfaces gráficas para publicar y programar un tweet.	20
3	Creación de funciones PHP para crear, mostrar, eliminar y programar tweet.	34
		64 Horas

1. Creación de scripts utilizando SQL

TABLA LVII. TAREA 1: CREACIÓN DE SCRIPTS UTILIZANDO SQL
DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS

N.- Tarea: 1	N.- Historia: 6
Nombre de la Tarea: Creación de scripts utilizando SQL	
Tipo de Tarea: Administrador de Base de Datos	Tiempo Estimado: 10 Horas
Programador Responsable: Franklin Michay	
Descripción: Esta tarea describe el proceso de creación del script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> .	

2. Diseño de interfaces gráficas para publicar y programar tweets.

TABLA LVIII. TAREA 2: DISEÑO DE INTERFACES GRÁFICAS PARA PUBLICAR Y PROGRAMAR TWEETS DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS

N.- Tarea: 2	N.- Historia: 6
Nombre de la Tarea: Diseño de interfaces gráficas para publicar y programar un tweet.	
Tipo de Tarea: Diseño y Maquetado	Tiempo Estimado: 20 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas para publicar y programar un tweet mediante la utilización de hojas de estilo CSS, archivos JAVASCRIPT y la librería <i>TWITTER BOOTSTRAP MASTER</i> , para el diseño de los componentes <i>HTML</i> .	
 Formulario Publicar Tweet: Permite ingresar el texto e imagen para la publicación del tweet. Formulario Programar Tweet: Permite ingresar el texto, imagen, fecha y hora de publicación del tweet.	

3. Creación de funciones PHP para crear, mostrar, eliminar y programar Tweet.

TABLA LIX. TAREA 3: CREACIÓN DE FUNCIONES PHP PARA CREAR, MOSTRAR, ELIMINAR Y PROGRAMAR TWEETS DEL CASO DE USO PUBLICAR Y PROGRAMAR TWEETS

N.- Tarea: 3	N.- Historia: 6
Nombre de la Tarea: Creación de funciones PHP para crear, mostrar, eliminar y programar Tweet.	
Tipo de Tarea: Programación	Tiempo Estimado: 34 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones <i>PHP</i> , que nos permitirá programar y publicar un tweet. function programer_tweet () : Función para programar los tweets de acuerdo a una fecha y hora. Function upload_file() : convierte la imagen seleccionada por el usuario a un identificador reconocible por el API de Twitter.	

✓ **Historia 7: Administrar Alertas**

TABLA LX. CRONOGRAMA DE ACTIVIDADES DEL CASO DE USO
ADMINISTRAR ALERTAS

Historia:7	Administrar Alertas	
Número	Nombre	Tiempo Estimado
1	Diseño de bases de datos	15
2	Diseño de interfaces gráficas para alerta	40
3	Mapeo Objeto-Relacional entidad alerta.	25
4	Creación de funciones PHP para administrar alerta	40
		120 Horas

1. Diseño de bases de datos

TABLA LXI. TAREA 1: DISEÑO DE BASE DE DATOS DEL CASO DE USO ADMINISTRAR ALERTAS

N.- Tarea: 1	N.- Historia: 7
Nombre de la Tarea: Diseño de bases de datos	
Tipo de Tarea: Administración de base de datos	Tiempo Estimado: 15 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none">• Diseño diagrama entidad relación entre entidades: Alerta y Usuario.• Un usuario puede crear varias alertas por lo que se incorporó la clave primaria de alerta hacia la tabla del usuario. También se describe el proceso de creación del script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> .	
create table alert: Permite crear la tabla “alert” en la cual se va almacenar las alertas creadas por cada usuario.	

2. Diseño de interfaces gráficas para Alerta

TABLA LXII. TAREA 2: DISEÑO DE INTERFACES GRÁFICAS PARA ALERTAS DEL CASO DE USO ADMINISTRAR ALERTAS

N.- Tarea: 2	N.- Historia: 7
Nombre de la Tarea: Diseño de interfaces gráficas para Alerta	
Tipo de Tarea: Diseño y maquetado	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas para mostrar alertas mediante la utilización de hojas de estilo CSS, archivos JAVASCRIPT y la librería <i>TWITTER BOOTSTRAP</i> , para el diseño de los componentes <i>HTML</i> .	
Formulario Crear Alerta: Permite ingresar los campos de texto para la creación de alertas.	
Formulario Editar Alerta: Permite editar los campos de texto seleccionados.	
Formulario Alertas: Permite mostrar la lista de alertas creadas por cada usuario.	

3. Mapeo objeto relacional entidad alerta.

TABLA LXIII. TAREA 3: MAPEO OBJETO RELACIONAL ENTIDAD ALERTA DEL CASO DE USO ADMINISTRAR ALERTAS

N.- Tarea: 3	N.- Historia: 7
Nombre de la Tarea: Mapeo objeto relacional entidad alerta	
Tipo de Tarea: Programación	Tiempo Estimado: 25 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de mapeo de la entidad alerta, aplicando el paradigma de la <i>programación orientada a objetos</i> , para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente.	
Métodos implementados: guarda_alarma(): Permite ingresar alerta a la base de datos, utilizando el comando <i>INSERT</i> . modifica(): Permite editar los campos de la tabla “alert” utilizando en comando <i>UPDATE</i> . elimina_alerta(): Recibe como parámetro el id de la entidad “alerta” y la elimina de la base de datos, utilizando el comando <i>DELETE</i> .	

4. Creación de funciones PHP para administrar alerta

TABLA LXIV. TAREA 4: CREACIÓN DE FUNCIONES PHP PARA
ADMINISTRAR ALERTA DEL CASO DE USO ADMINISTRAR
ALERTAS

N.- Tarea: 4	N.- Historia: 7
Nombre de la Tarea: Creación de funciones PHP para administrar alerta	
Tipo de Tarea: Programación	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones <i>PHP</i> , que nos permitirá administrar las alertas. Function ejecuta_alerta(): Verifica en base de datos si existen tweets que coinciden con cada alerta creada. Function extrae_alerta(): Muestra la lista de alertas en la pantalla “Alertas”. Function listar_tw_alerta(): Muestra la lista de tweets que pertenecen a una alerta.	

✓ **Historia 8: Administrar Usuarios**

TABLA LXV. CRONOGRAMA DE ACTIVIDADES DEL
CASO DE USO ADMINISTRAR USUARIOS

Historia: 8 Administrar Usuarios		
Número	Nombre	Tiempo Estimado
1	Diseño de bases de datos	15
2	Diseño de interfaces gráficas para usuario	40
3	Mapeo Objeto-Relacional entidad usuario	25
4	Creación de funciones PHP para administrar usuario	40
		140 Horas

1. Diseño de bases de datos

TABLA LXVI. TAREA 1: DISEÑO DE BASE DE DATOS DEL CASO DE USO ADMINISTRAR USUARIO

N.- Tarea: 1	N.- Historia: 8
Nombre de la Tarea: Diseño de bases de datos	
Tipo de Tarea: Administración de base de datos	Tiempo Estimado: 15 Horas
Programador Responsable: Franklin Michay	
Descripción: Se describe el proceso de creación del script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PhpMyAdmin</i> . create table login: Permite crear la tabla “login” en la cual se van a almacenar los usuarios creados. create table admin: Permite crear la tabla “admin” en la cual se van a almacenar los usuarios administradores de la aplicación.	

2. Diseño de interfaces gráficas para usuario

TABLA LXVII. TAREA 2: DISEÑO DE INTERFACES GRÁFICAS PARA USUARIO DEL CASO DE USO ADMINISTRAR USUARIOS

N.- Tarea: 2	N.- Historia: 8
Nombre de la Tarea: Diseño de interfaces gráficas para Usuario	
Tipo de Tarea: Diseño y maquetado	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de diseño, para la creación de las pantallas para mostrar alertas mediante la utilización de hojas de estilo CSS, archivos JAVASCRIPT y la librería <i>TWITTER BOOTSTRAP</i> , para el diseño de los componentes <i>HTML</i> .	
Formulario Crear Usuario: Permite ingresar los campos de texto para la creación de usuarios.	
Formulario Editar Usuario: Permite editar los campos de texto seleccionados.	
Formulario Administrar Usuario: Permite mostrar la lista de usuarios creados.	
Formulario Alertas Creadas: Permite mostrar la lista de alertas creadas por un usuario.	
Formulario Tweets Programados: Permite mostrar la lista de tweets programados por un usuario.	

3. Mapeo objeto relacional entidad alerta.

TABLA LXVIII. TAREA 3: MAPEO OBJETO RELACIONAL ENTIDAD USUARIO DEL CASO DE USO ADMINISTRAR USUARIOS

N.- Tarea: 3	N.- Historia: 8
Nombre de la Tarea: Mapeo objeto relacional entidad usuario	
Tipo de Tarea: Programación	Tiempo Estimado: 25 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de mapeo de la entidad usuario, aplicando el paradigma de la <i>programación orientada a objetos</i> , para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente.	
Métodos implementados: comprobar() : Permite comprobar el usuario ingresado en la pantalla login. desactivar_usuario() : Permite desactivar el usuario tras tres intentos de ingreso al sistema. crear() : Permite guardar los datos de un usuario en base de datos. modifica_usuario() : Permite modificar los parámetros de una usuario. listar_usuario() : Permite listar todos los usuarios para mostrarlos en las vistas de la aplicación. elimina_usuario() : Permite eliminar un usuario.	

4. Creación de funciones PHP para administrar alerta

TABLA LXIX. TAREA 4: CREACIÓN DE FUNCIONES PHP PARA ADMINISTRAR USUARIO DEL CASO DE USO ADMINISTRAR USUARIOS

N.- Tarea: 4	N.- Historia: 8
Nombre de la Tarea: Creación de funciones PHP para administrar usuario	
Tipo de Tarea: Programación	Tiempo Estimado: 40 Horas
Programador Responsable: Franklin Michay	
Descripción: La presente tarea describe el proceso de creación de funciones <i>PHP</i> , que nos permitirá administrar usuarios. Function ingreso(): Verifica el número de intentos erróneos al intentar ingresar al sistema. Function logout(): Permite salir de la sesión. Function extrer_alertas(): Muestra la lista de alertas creadas por un usuario. Function extrer_tweets_programados(): Muestra la lista de tweets creados por un usuario.	

3.2.4. Pruebas Iteración III

1) Especificación de prueba: Publicar y programar tweets

Historia de Usuario: 6

Versión <1.6>

Historial de Revisiones

TABLA LXX. HISTORIAL DE REVISIONES ITERACIÓN III VERSIÓN
1.6

Versión	Descripción	Autor
1.6	Revisión	Franklin Michay
1.7	Modificación	Franklin Michay

En esta historia se verifica que los datos ingresados para la publicación y programación de un tweet, cumplan con las especificaciones establecidas en la historia 2. El usuario deberá ingresar el texto del tweet no mayor a 144 caracteres y seleccionar una imagen previo a la programación y/o publicación de un tweet.

Registro de información correcta

✓ Descripción:

Los valores introducidos por el usuario deben cumplir con las reglas establecidas por el API de Twitter para la publicación de un tweet.

✓ Condiciones de ejecución:

El usuario deberá estar previamente creado, para poder registrarse y acceder hacia el sistema.

✓ Entrada:

El usuario deberá ingresar el texto del tweet y seleccionar una imagen de su

ordenador.

✓ **Resultado esperado:**

Tras la introducción de datos, si son correctos se publicará y/o programará el tweet.

✓ **Evaluación de la prueba:** Prueba satisfactoria.

Registro de información incorrecta

✓ **Descripción:**

Si los datos ingresados por el usuario no cumplen con las reglas establecidas por el API de Twitter, se imprimirán alertas definidas por el API de Twitter, que indiquen el estado de los campos ingresados.

✓ **Condiciones de ejecución:**

El usuario deberá haber creado previamente una cuenta de Twitter para poder registrarse y acceder hacia el sistema.

✓ **Entrada:**

El usuario ingresará el texto y/o imagen.

✓ **Resultado esperado:**

El TimeLine de Twitter muestra el tweet (programado o publicado).

✓ **Evaluación de la prueba:** Prueba satisfactoria.

2) Especificación de prueba: Administrar Alertas

Historia de Usuario: 7

Versión <1.7>

Historial de Revisiones

TABLA LXXI. HISTORIAL DE REVISIONES ITERACIÓN III VERSIÓN
1.7

Versión	Descripción	Autor
1.7	Revisión	Franklin Michay
1.8	Modificación	Franklin Michay

En esta historia se verifica que las acciones que se realizan en las pantallas cumplan con las especificaciones establecidas en la historia 7. Los usuarios podrán realizar el respectivo CRUD (crear, leer, actualizar y eliminar) y ejecutar una alerta.

Registro de información correcta

✓ **Descripción:**

El usuario podrá crear, modificar, eliminar y ejecutar una alerta.

✓ **Condiciones de ejecución:**

El usuario deberá estar registrado y logueado con una cuenta de Twitter e ingresar a la pantalla Interacción.

✓ **Entrada:**

El usuario deberá ingresar un nombre para la alerta, así como sus campos.

✓ **Resultado esperado:**

Si el proceso ha sido correcto, los campos de texto se registrarán en la base de

<p>datos.</p>
<ul style="list-style-type: none"> ✓ Evaluación de la prueba: Prueba satisfactoria.
<h3>Registro de información incorrecta</h3>
<ul style="list-style-type: none"> ✓ Descripción: <p>Tras ejecutar una alerta no se encuentran tweets relacionados con la misma.</p>
<ul style="list-style-type: none"> ✓ Condiciones de ejecución: <p>El usuario deberá estar registrado y logueado con una cuenta de Twitter e ingresar a la pantalla Interacción.</p>
<ul style="list-style-type: none"> ✓ Entrada: <p>El usuario iniciará sesión con un usuario normal y posteriormente con un usuario de una cuenta de Twitter.</p>
<ul style="list-style-type: none"> ✓ Resultado esperado: <p>En caso de que no se encuentren tweets relacionados con una alerta el sistema mostrará un mensaje informando dicha acción.</p>
<ul style="list-style-type: none"> ✓ Evaluación de la prueba: Prueba satisfactoria.

3) Especificación de prueba: Administrar Usuarios

Historia de Usuario: 8

Versión <1.8>

Historial de Revisiones

TABLA LXXII. HISTORIAL DE REVISIONES ITERACIÓN III
VERSIÓN 1.8

Versión	Descripción	Autor
1.8	Revisión	Franklin Michay
1.9	Modificación	Franklin Michay

En esta historia se verifica que las acciones que se realizan en las pantallas cumplan con las especificaciones establecidas en la historia 8. Los usuarios podrán realizar el respectivo CRUD (crear, leer, actualizar y eliminar) y ejecutar un usuario.

Registro de información correcta
<p>✓ Descripción: El usuario administrador podrá crear, modificar, eliminar, ver alertas creadas, ver tweets programados por un usuario.</p>
<p>✓ Condiciones de ejecución: El usuario administrador deberá estar registrado en base de datos.</p>
<p>✓ Entrada: El usuario administrador deberá ingresar los datos de registro de un usuario.</p>
<p>✓ Resultado esperado: Si el proceso ha sido correcto, los campos de texto se registrarán en la base de</p>

datos.

- ✓ **Evaluación de la prueba:** Prueba satisfactoria.

Registro de información incorrecta

- ✓ **Descripción:**

Tras ingresar los datos de registro incorrectos, el sistema mostrará alertas de los mismos.

- ✓ **Condiciones de ejecución:**

El usuario administrador deberá ingresar de manera correcta los datos de registro de un usuario.

- ✓ **Entrada:**

El usuario administrador ingresará correctamente los datos de registro de un usuario.

- ✓ **Resultado esperado:**

Tras ingresar datos erróneos de una usuario, el sistema mostrará alertas de dicha acción.

- ✓ **Evaluación de la prueba:** Prueba satisfactoria.

3.2.5. Resultados de la Iteración III

- ✓ Plan de entrega inicial

Historia 6: Publicar y programar tweets

Historia 7: Administrar Alertas

Historia 8: Administrar Usuarios

Versión del sistema 1.9

3.2.6. Demo de la Versión: Iteración III

Historia 6: Publicar y programar tweet (8 días)

Publicar Tweet: El usuario debe seleccionar el botón (Publicar Tweet) de la pantalla principal, luego deberá ingresar el texto, seleccionar una imagen desde su ordenador y seleccionar el botón Twittear.

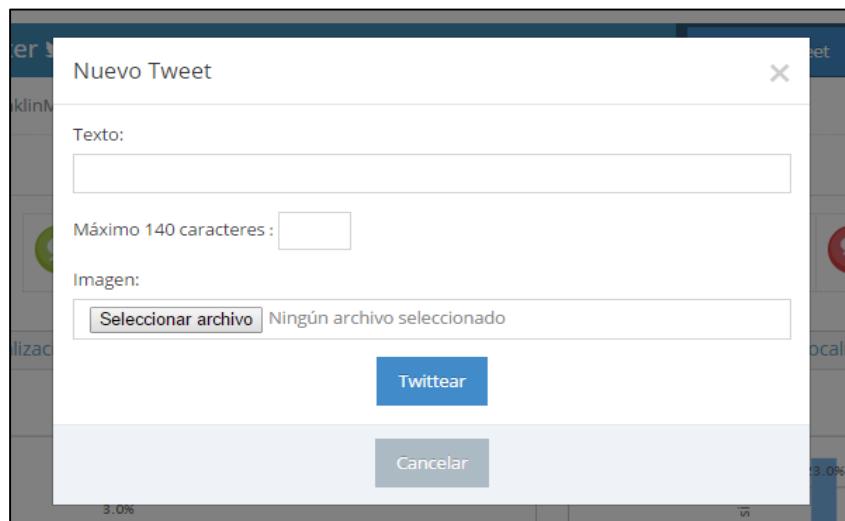


Figura 39. Publicar tweets

Programar Tweet: El usuario debe seleccionar el botón (Programar Tweet), luego deberá ingresar el texto, seleccionar una imagen desde su ordenador, seleccionar la fecha y hora de publicación y seleccionar el botón programar tweet.

Nuevo Tweet

Texto:

Máximo 140 caracteres

Imagen:
 Ningún archivo seleccionado

Selección el día

Seleccione las horas en las que se debe programar el tweet

Hora en la mañana :

Hora en la Tarde:

Hora en la Noche:

Figura 40. Programar tweets

Historia 7: Administrar Alertas (15 días)

Visualizar Alertas: El usuario debe ingresar al submenú **Alertas** del menú **Interacción**, en esta pantalla se mostrarán las alertas creadas.

The screenshot shows a web-based application interface for managing alerts. At the top left is the title 'Alertas'. In the center is a Twitter logo. Below the title is a blue button labeled 'Crear Alerta'. A search bar and a 'Show 10 entries' dropdown are also present. The main area contains a table with two rows of data. The columns are labeled 'Nombre', 'Usuario', 'Fecha Creación', 'Parámetros', and actions. The first row has a 'Nombre' of 'Accidente', 'Usuario' of 'Franklin', 'Fecha Creación' of '2016-07-05 21:57:28', and 'Parámetros' of 'accidente urgente estimado'. The second row has a 'Nombre' of 'asdasd', 'Usuario' of 'franklin', 'Fecha Creación' of '2016-09-28 14:50:35', and 'Parámetros' of 'sadas dasdas dasd'. Each row has three buttons at the end: 'Editar' (blue), 'Eliminar' (red), and 'Ejecutar' (green).

Nombre	Usuario	Fecha Creación	Parámetros	
Accidente	Franklin	2016-07-05 21:57:28	accidente urgente estimado	Editar Eliminar Ejecutar
asdasd	franklin	2016-09-28 14:50:35	sadas dasdas dasd	Editar Eliminar Ejecutar

Figura 41. Lista de alertas

Crear Alerta: El usuario debe ingresar al submenú **Crear Alerta** del menú **Interacción**, en esta pantalla podrá ingresar el nombre y los campos de la Alerta.

The screenshot shows a modal dialog titled 'Nueva Alerta' (New Alert). On the left is a sidebar menu with categories: 'Tu Comunidad', 'Analítica', 'Tweets', and 'Interacción'. Under 'Interacción', there are three options: 'Alertas', 'Nueva Alerta' (which is selected and highlighted in blue), and 'Tweets Programados'. The main area of the dialog has a 'Cancelar' button at the top right. It contains four input fields labeled 'Nombre', 'Parámetro 1', 'Parámetro 2', and 'Parámetro 3', each with a placeholder text. Below these is another input field labeled 'Parámetro 4'. At the bottom right is a blue 'Aceptar' (Accept) button with a checkmark icon.

Figura 42. Crear nueva alerta

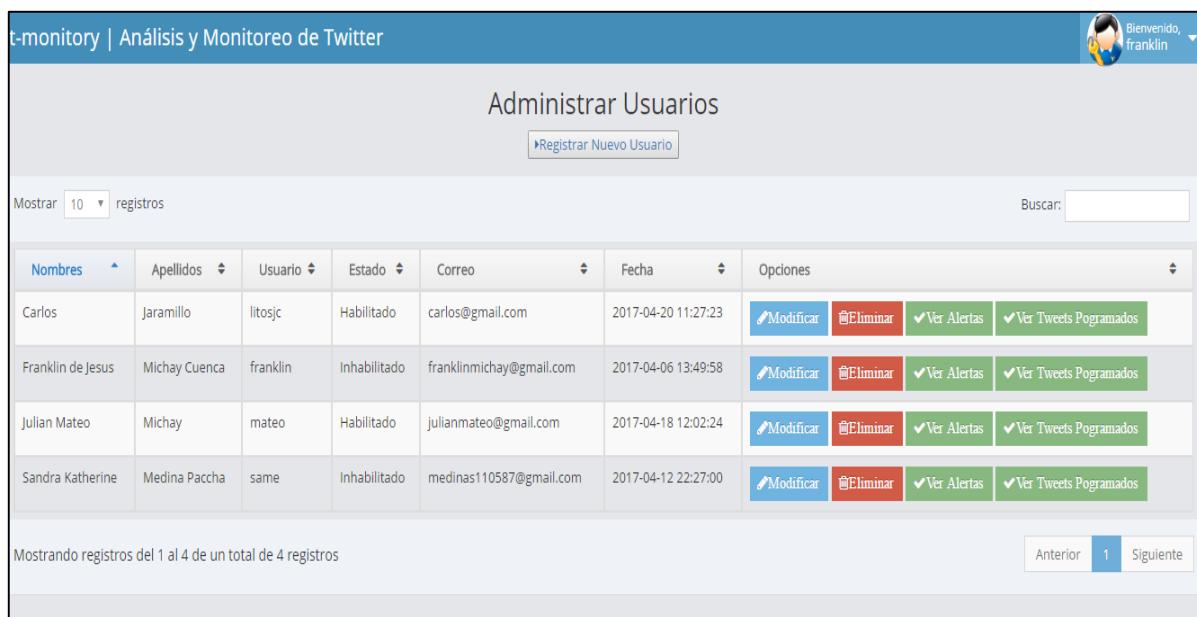
Historia 8: Administrar Usuarios (15 días)

Visualizar Usuarios: El usuario administrador debe ingresar al sistema con su clave y contraseña.



The image shows a login interface titled "Administración de Usuarios". At the top right is a "Ir a Inicio" button. Below it is a "Ingresar a su Cuenta" section with two input fields: one containing "franklin" and another showing a masked password. A blue "Ingresar" button is at the bottom.

Figura 43. Login: Administración de usuarios



The image shows a user management interface titled "t-monitory | Análisis y Monitoreo de Twitter". It displays a list of users with columns for Nombres, Apellidos, Usuario, Estado, Correo, Fecha, and Opciones. The users listed are Carlos Jaramillo, Franklin de Jesus, Julian Mateo, and Sandra Katherine. Each user has a "Modificar" and "Eliminar" button, along with checkboxes for "Ver Alertas" and "Ver Tweets Pogramados". The interface includes pagination and search functionality.

Nombres	Apellidos	Usuario	Estado	Correo	Fecha	Opciones
Carlos	Jaramillo	litosjc	Habilitado	carlos@gmail.com	2017-04-20 11:27:23	<input checked="" type="checkbox"/> Ver Alertas <input checked="" type="checkbox"/> Ver Tweets Pogramados
Franklin de Jesus	Michay Cuenca	franklin	Inhabilitado	franklinmichay@gmail.com	2017-04-06 13:49:58	<input checked="" type="checkbox"/> Ver Alertas <input checked="" type="checkbox"/> Ver Tweets Pogramados
Julian Mateo	Michay	mateo	Habilitado	julianmateo@gmail.com	2017-04-18 12:02:24	<input checked="" type="checkbox"/> Ver Alertas <input checked="" type="checkbox"/> Ver Tweets Pogramados
Sandra Katherine	Medina Paccha	same	Inhabilitado	medinas110587@gmail.com	2017-04-12 22:27:00	<input checked="" type="checkbox"/> Ver Alertas <input checked="" type="checkbox"/> Ver Tweets Pogramados

Figura 44. Lista de Usuarios creados

Visualizar Alertas y Tweets creados: El usuario administrador debe seleccionar los botones [Ver Alertas] y [Ver Tweets Programados].

Nombre	Usuario	Fecha Creación	Parámetros	Opciones
prueba alerta	mateo	2017-04-19 12:20:27	programación tweet @TTmonitor	Editar Eliminar

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

Anterior 1 Siguiente

Figura 45. Lista de Alertas creadas por un usuario

Texto	Nombre IMG	Fecha Creación	Horario	Día	Usuario	Opciones
Prueba 2	95df6ce44f3cd30b27a7a4668a4797a2.jpg	2016-10-13 12:57:01		2016-10-13	franklin	Editar Eliminar
prueba 4	Ashampoo_Snap_0001.png	2016-10-13 13:02:01		2016-10-13	franklin	Editar Eliminar

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

Anterior 1 Siguiente

Figura 46. Lista de Tweets programados por un usuario

3.2.7. Pruebas de Aceptación

Historia 6: Publicar y programar tweets

- ✓ La publicación del tweets funciona.
- ✓ La programación del tweets funciona.

Historia 7: Administrar Alertas

- ✓ La pantalla Alertas permite visualizar las alertas creadas por un usuario.
- ✓ La pantalla Crear Alerta permite crear y editar alertas.

Historia 8: Administrar Usuarios

- ✓ La pantalla Administrar Usuarios permite visualizar los usuarios creados.
- ✓ La pantalla Registrar Nuevo Usuario permite ingresar los datos de registro de un usuario de manera correcta.
- ✓ La pantalla Alertas Creadas permite visualizar las alertas creadas por un usuario.
- ✓ La pantalla Tweets Programados permite visualizar los tweets programados por un usuario.

3.2.8. Incidencias

- ✓ Se crearon nuevas funcionalidades debido a que en un inicio, no se contaba con la administración de usuarios.
- ✓ Se hicieron cambios en la base de datos.

3.2.9. Diagrama de Clases

Después de finalizar con cada iteración y culminar con las pruebas de aceptación, se presenta el modelo de clases final, que es el resultado de un modelado constante, pues de esta manera ajustamos el sistema a los requerimientos reales del usuario.

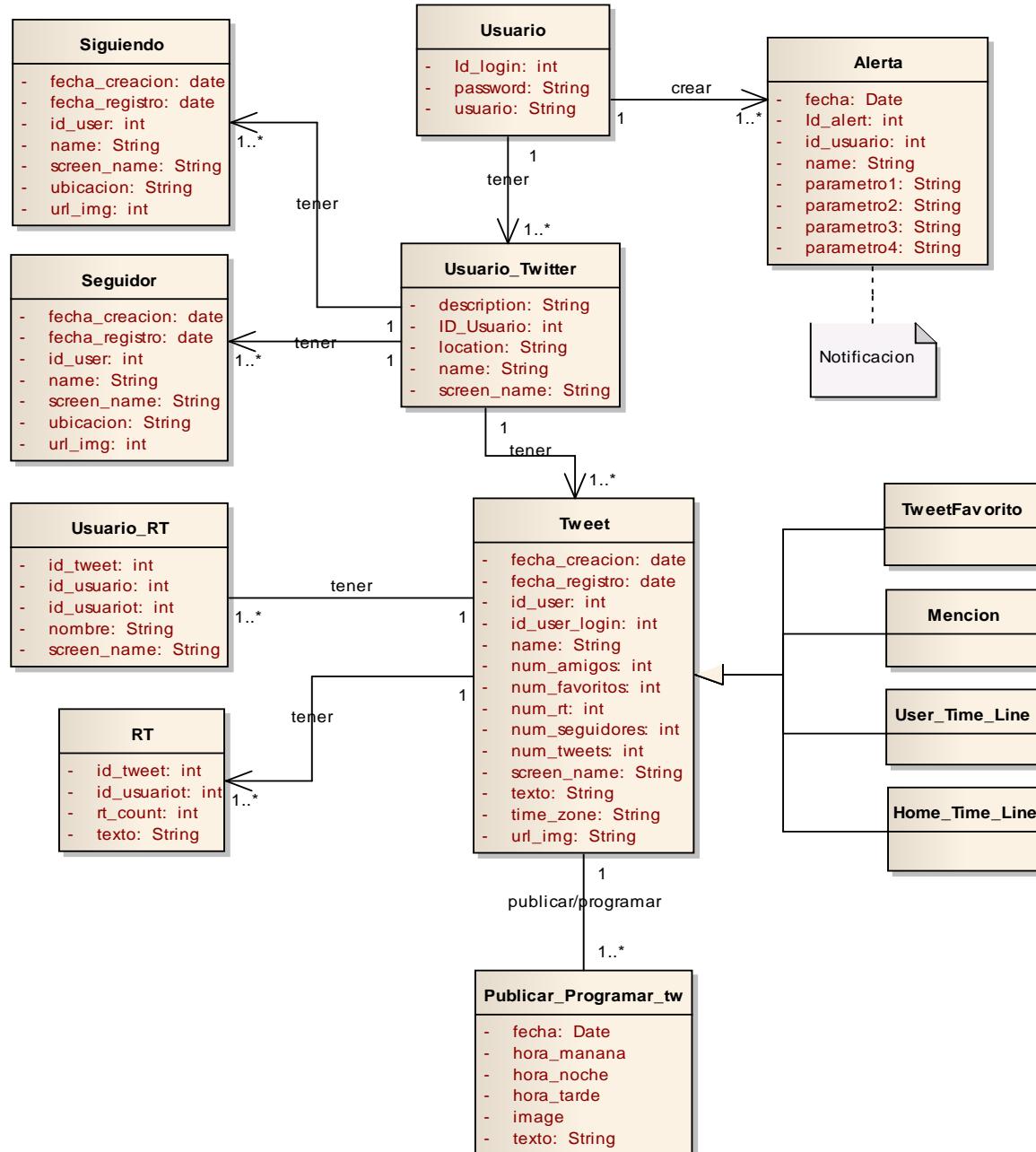


Figura 47. Diagrama de clases de la aplicación

3.2.10. Modelo Entidad Relación

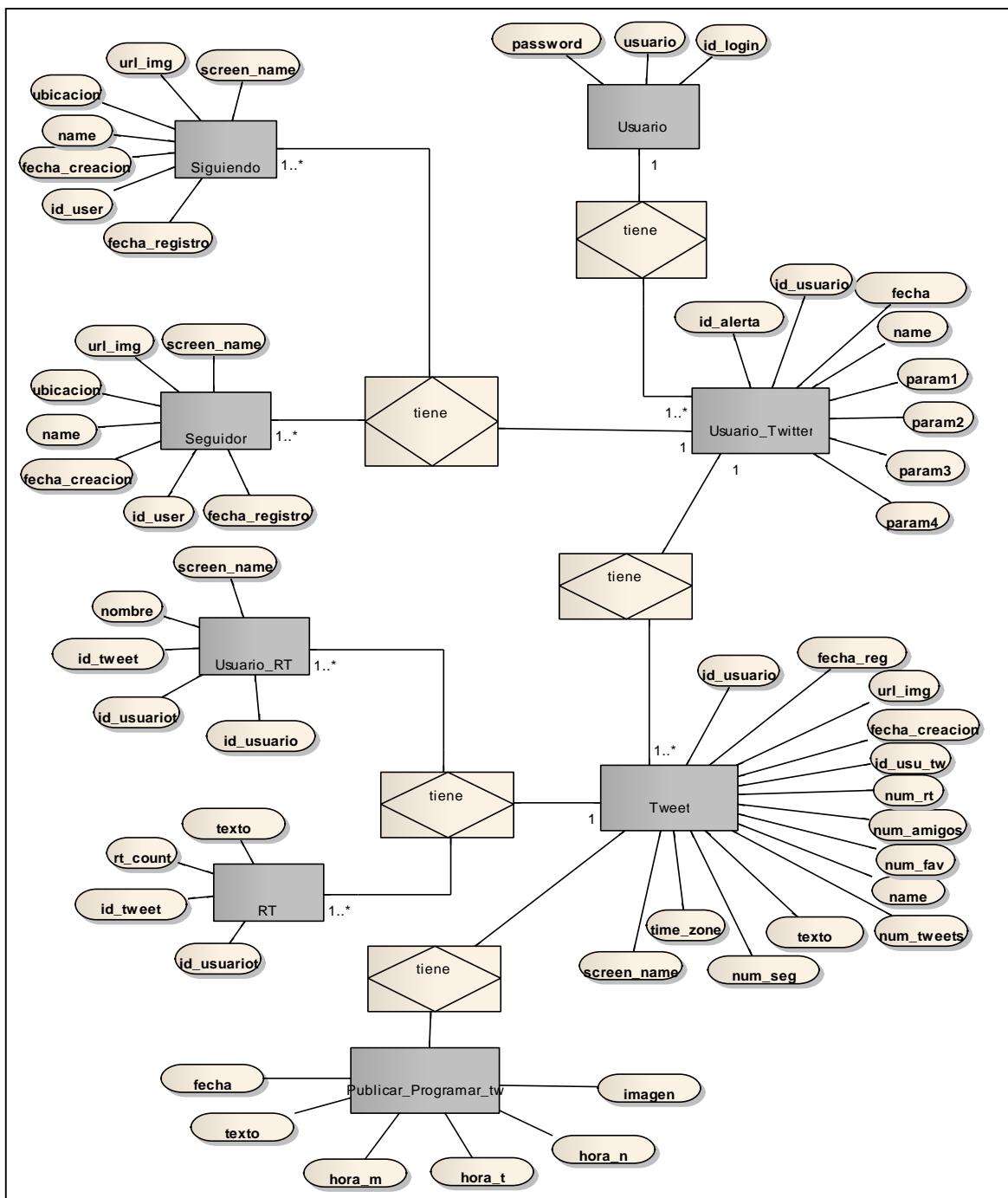


Figura 48. Modelo entidad relación de la aplicación

4. CUARTA FASE: Producción

4.1. Pruebas de aceptación de la aplicación

4.1.1. Pruebas de Carga o stress

JMeter es una herramienta OpenSource que permite realizar pruebas de carga sobre nuestras aplicaciones. Las pruebas se pueden realizar sobre casi cualquier elemento de software de nuestra aplicación (una función/procedimiento de base de datos, un método Java, un servicio web, una página web, etc.) y a través de diversos protocolos (TCP/IP, HTTP, JDBC, etc.).

La prueba de carga o stress se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Lo único que configuraremos es un escenario que simule 5 peticiones simultáneas al sitio "aplicacioneswebtwitter.com" y ver cómo se comporta cada petición realizada. Para esto seguiremos los siguientes pasos en la aplicación de JMeter:

1. Configurar un Grupo de Hilos

El elemento Grupo de Hilos es el punto de partida de cualquier Plan de Pruebas, este elemento controla el número de hilos que va a ejecutar nuestra prueba.

Entre las principales propiedades que vamos a editar están:

- ✓ La cantidad de hilos.
- ✓ El tiempo de subida (El tiempo que JMeter tendrá para levantar la cantidad de hilos configurados).
- ✓ La cantidad de veces (iteraciones) que se ejecutará la prueba.

Para configurarlo, seguiremos estos pasos:

a) Agregamos un grupo de hilos que serían los usuarios:

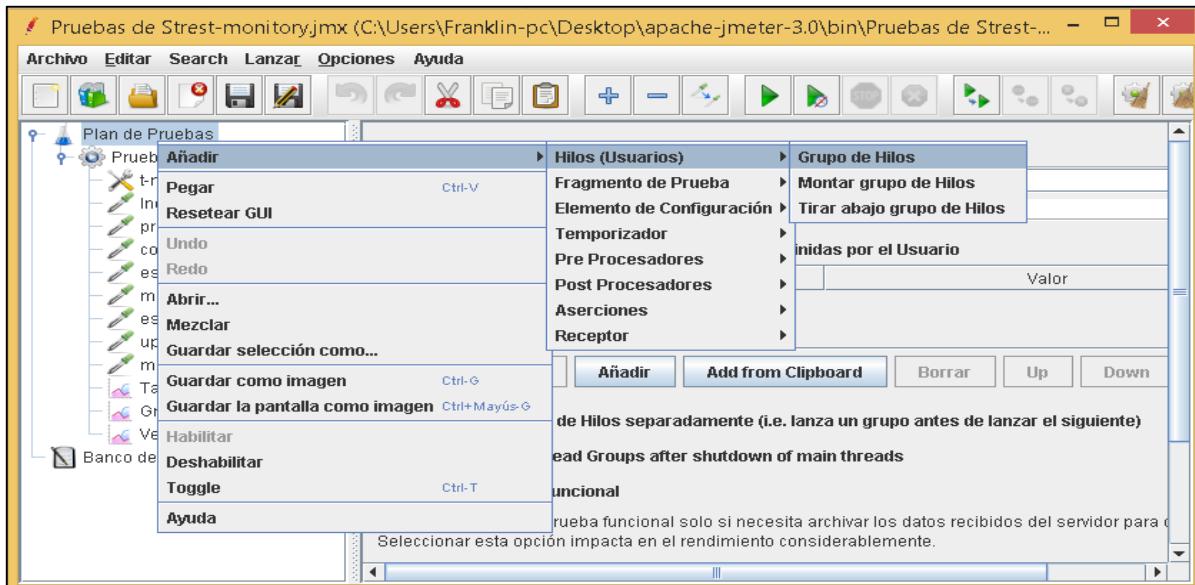


Figura 49. Añadir grupo de hilos en JMeter

En número de hilos: 50, Periodo de subida: 10, quiere decir que 50 usuarios van a ingresar en 10 segundos, es decir 5 usuarios por segundo.

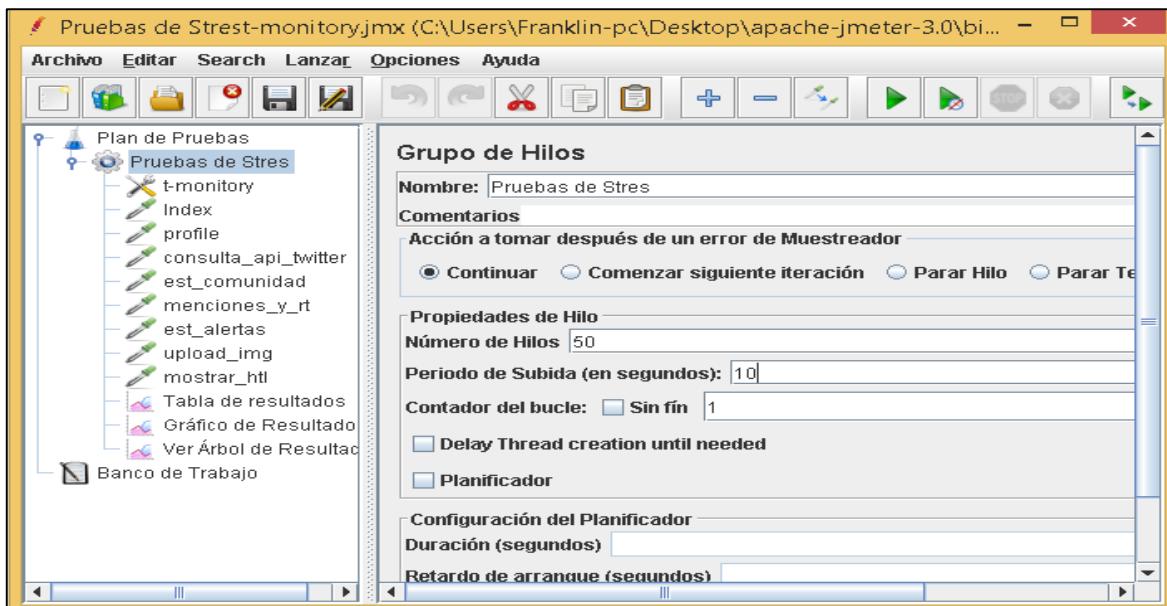


Figura 50. Ingreso de datos en JMeter

- b) Dentro del grupo de hilos agregamos un elemento de configuración, valor por defecto para las peticiones HTTP, donde colocaremos el dominio del sitio web:

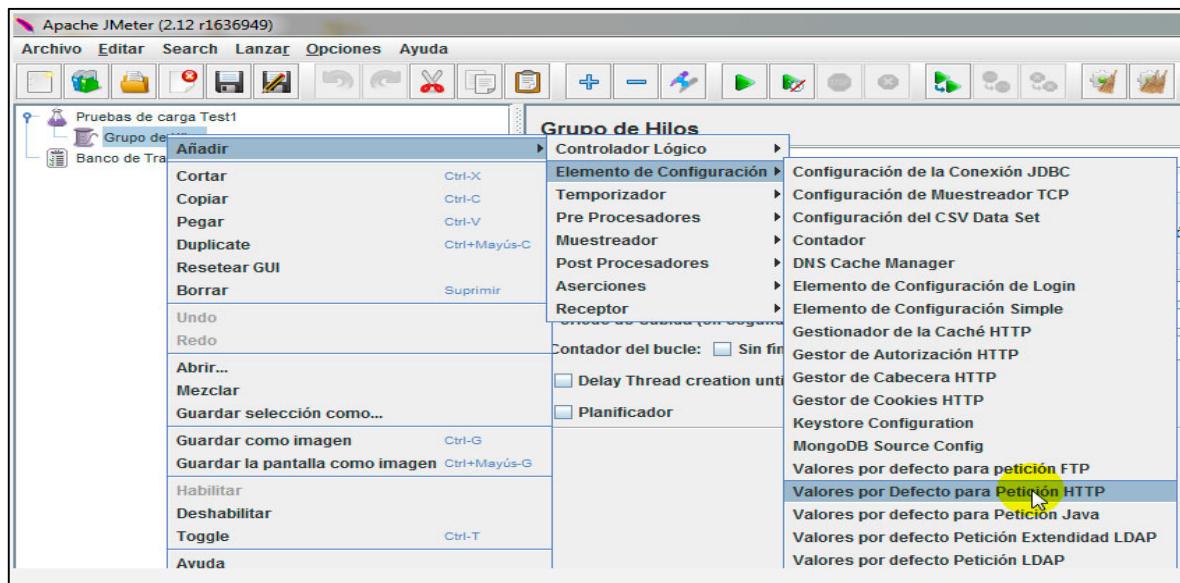


Figura 51. Valor por defecto para la configuración HTTP en JMeter

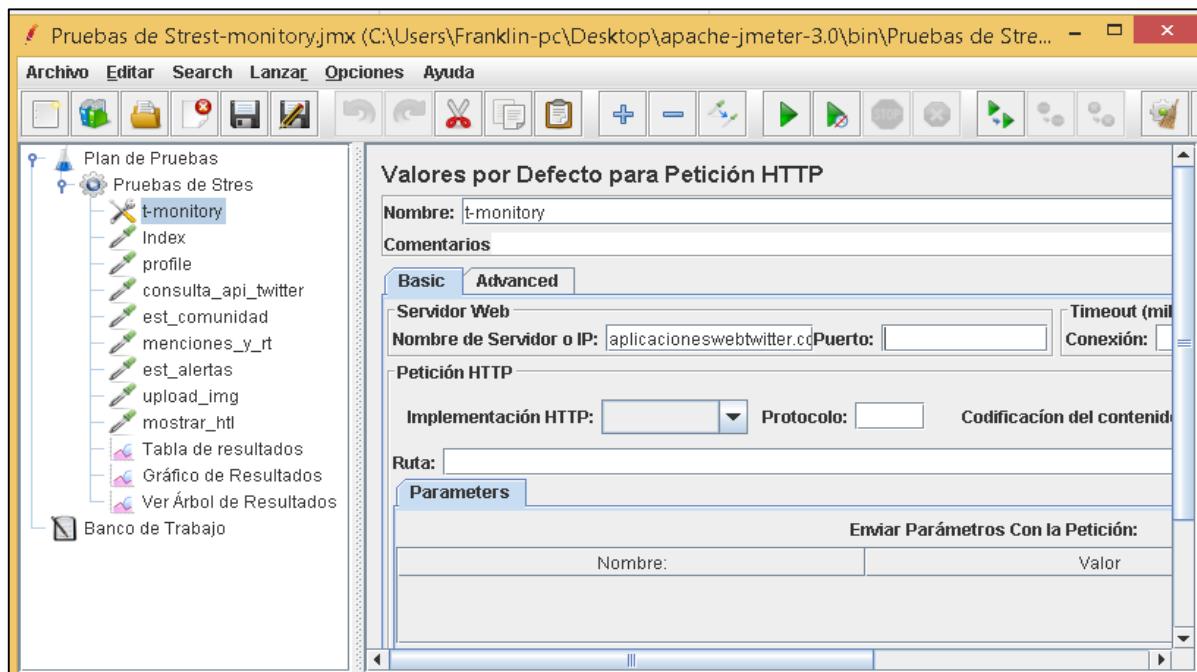


Figura 52. Ingreso del nombre del servidor en JMeter

2. Configurar una petición HTTP

Este elemento se comporta como un muestrador. Y su función principal es la de realizar una solicitud HTTP hacia un servidor web.

Entre las principales propiedades que vamos a editar están:

- ✓ El nombre del servidor o dirección IP.
- ✓ El método HTTP que se va a usar para realizar la solicitud.

Para configurarlo, seguiremos estos pasos:

a) Agregamos una nueva petición HTTP:

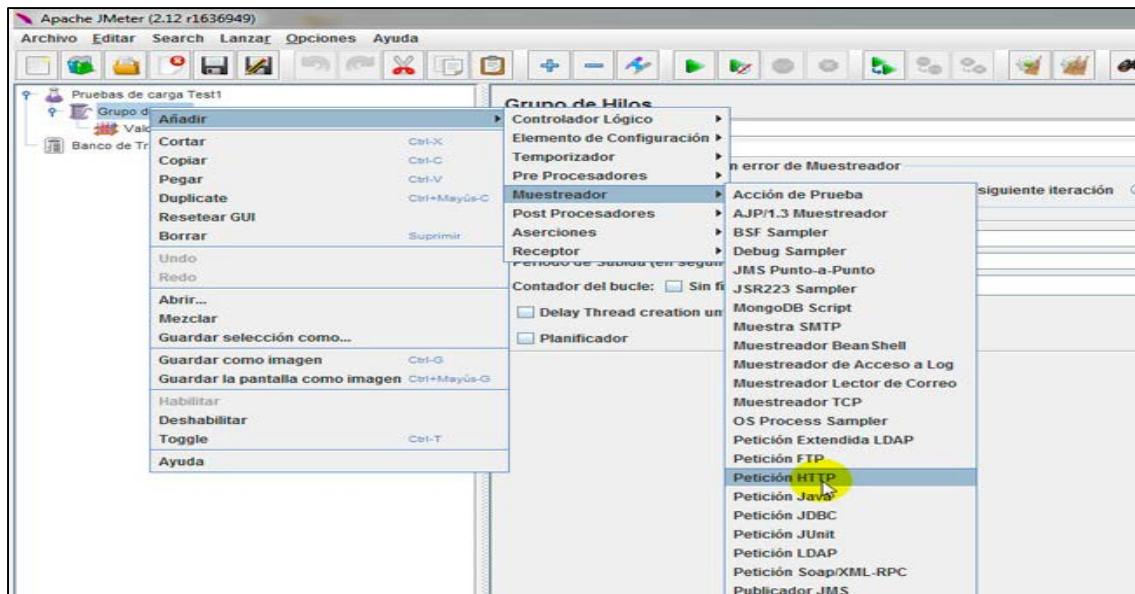


Figura 53. Nueva petición HTTP en JMeter

b) Agregamos la ruta de las peticiones HTTP:

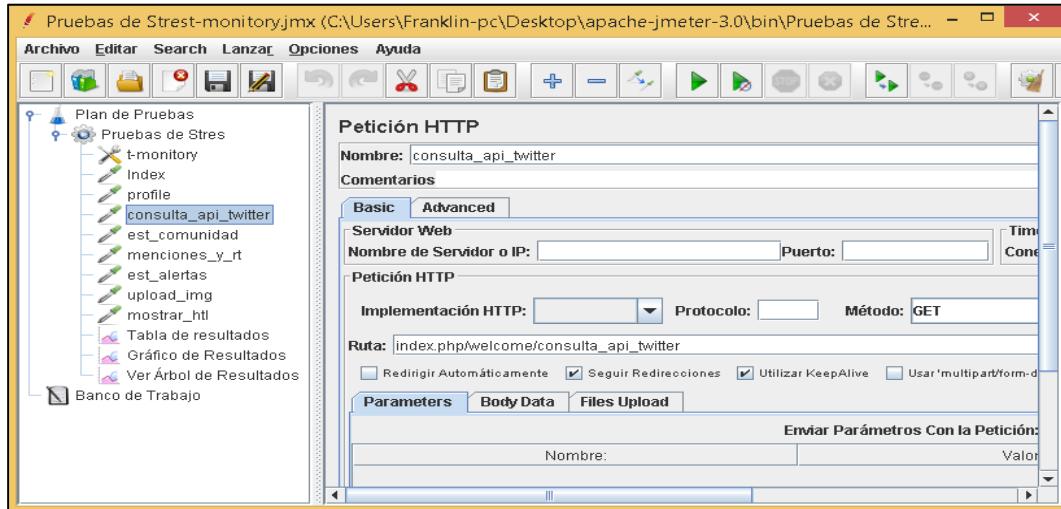


Figura 54. Nueva ruta de petición HTTP en JMeter

3. Configurar un reporte "Árbol de Resultados"

Este elemento se comporta como un receptor. Y permite visualizar a manera de árbol cada una de las respuestas obtenidas por la ejecución de los muestreadores. Además, se puede ver el tiempo que le tomó obtener la respuesta y algunos códigos de respuesta.

Para configurarlo, seguiremos estos pasos:

a) Agregamos un nuevo Árbol de Resultados

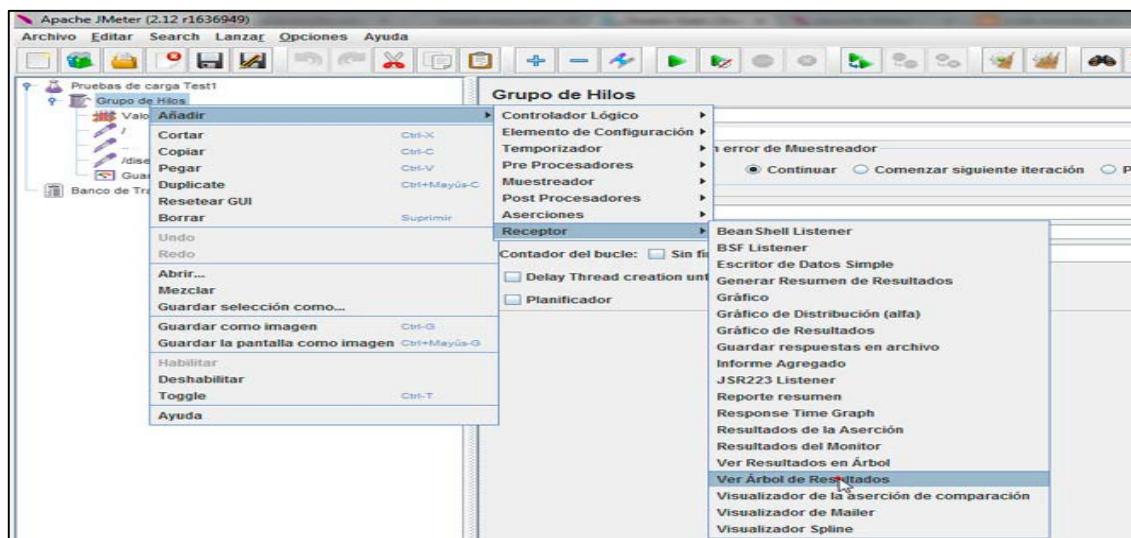


Figura 55. Nueva Árbol de Resultados en JMeter

Luego de esto procederemos a darle click al botón de la flecha verde (Arrancar) para iniciar nuestra prueba. En ese momento, el grupo de hilos lanzará 50 hilos (en nuestro caso) haciendo 50 peticiones HTTP simultáneas al servidor configurado. Los resultados de cada una de esas peticiones podremos verlo en nuestro elemento Árbol de Resultados:

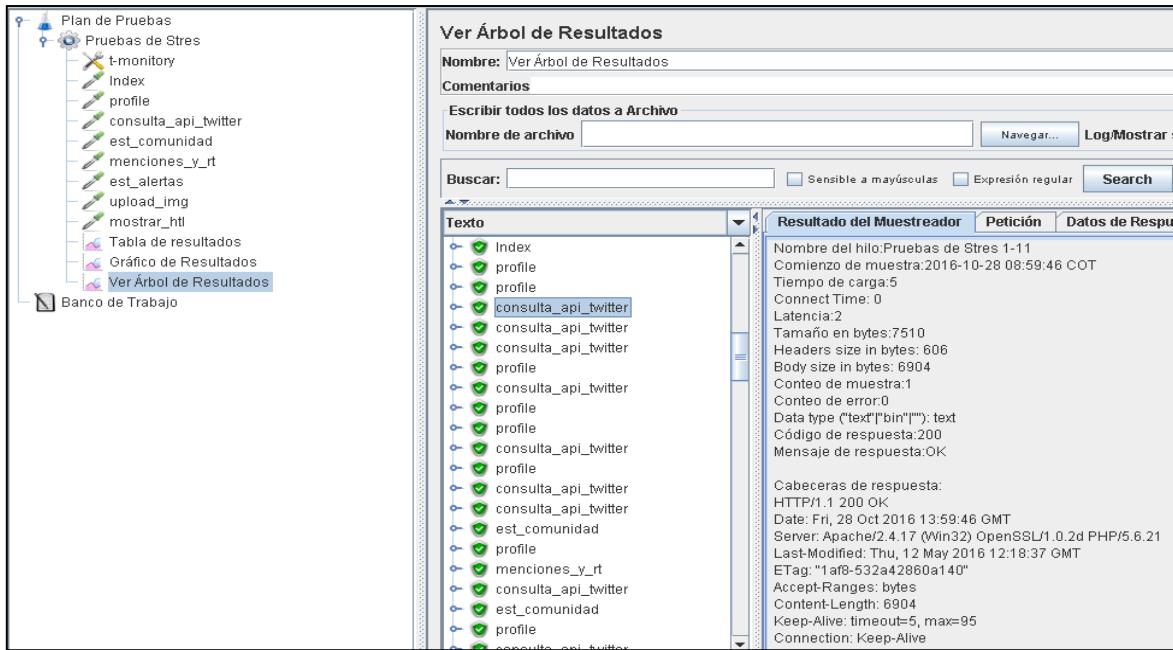


Figura 56. Árbol de Resultados en JMeter

4. Configurar un reporte de “Tabla de Resultados”

Muestra una tabla con todas las respuestas, la URL, tiempo y resultado de ejecución de cada una de ellas.

a) Agregamos una tabla de resultados

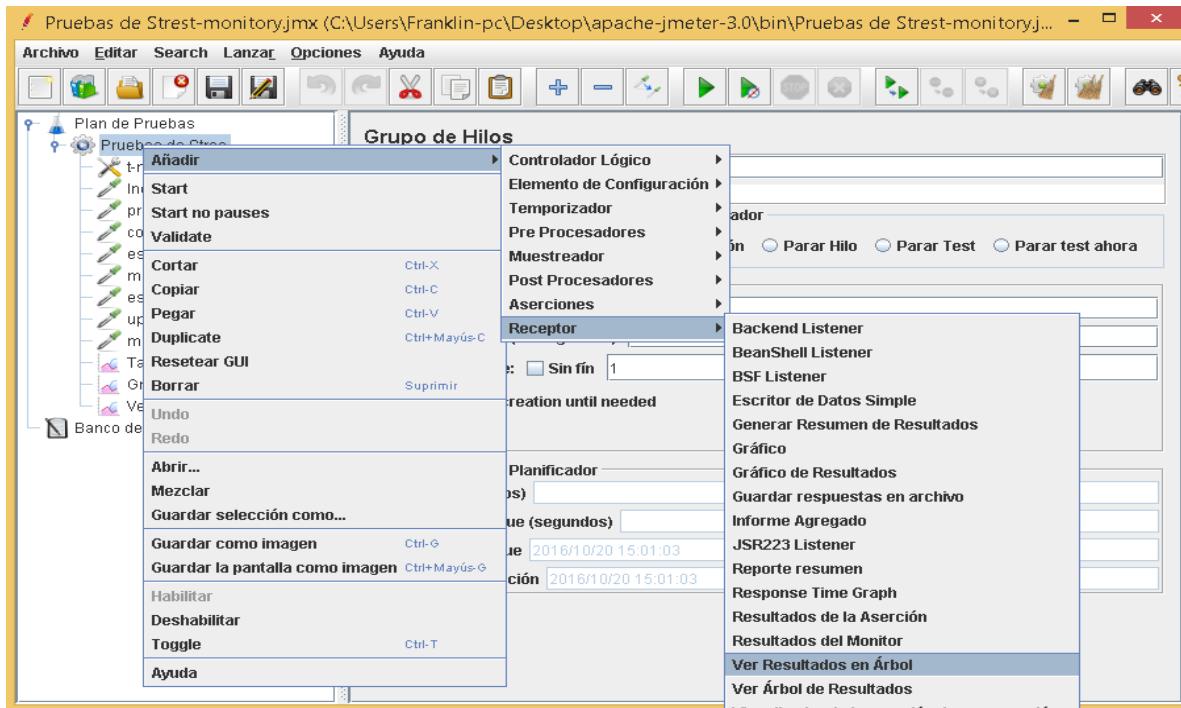
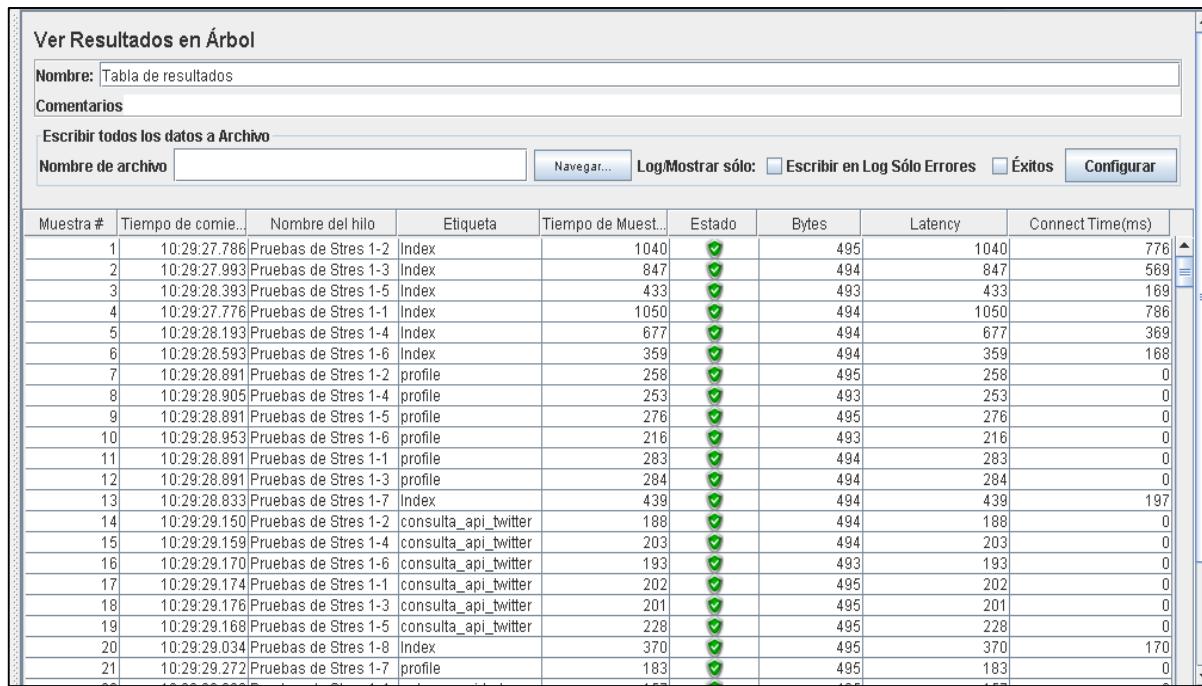


Figura 57. Nueva Tabla de Resultados en JMeter

Luego de esto procederemos a darle click al botón de la flecha verde (Arrancar) para iniciar nuestra prueba. En ese momento, el grupo de hilos lanzará 50 hilos (en nuestro caso) haciendo 50 peticiones HTTP simultáneas al servidor configurado. Los resultados de cada una de esas peticiones podremos verlo en nuestro elemento Tabla de Resultados:



The screenshot shows the 'Ver Resultados en Árbol' (View Results in Tree) window in JMeter. At the top, there are fields for 'Nombre:' (Name: Tabla de resultados), 'Comentarios' (Comments), and 'Escribir todos los datos a Archivo' (Write all data to file). Below these are buttons for 'Nombre de archivo' (File name), 'Navegar...' (Browse...), 'Log/Mostrar sólo:' (Log/Show only:), and checkboxes for 'Escribir en Log Sólo Errores' (Write only errors to log) and 'Éxitos' (Successes). A 'Configurar' (Configure) button is also present. The main area is a table with the following data:

Muestra #	Tiempo de comie..	Nombre del hilo	Etiqueta	Tiempo de Muest..	Estado	Bytes	Latency	Connect Time(ms)
1	10:29:27.786	Pruebas de Stres 1-2	Index		1040	495	1040	776
2	10:29:27.993	Pruebas de Stres 1-3	Index		847	494	847	569
3	10:29:28.393	Pruebas de Stres 1-5	Index		433	493	433	169
4	10:29:27.778	Pruebas de Stres 1-1	Index		1050	494	1050	786
5	10:29:28.193	Pruebas de Stres 1-4	Index		677	494	677	389
6	10:29:28.593	Pruebas de Stres 1-6	Index		359	494	359	168
7	10:29:28.891	Pruebas de Stres 1-2	profile		258	495	258	0
8	10:29:28.905	Pruebas de Stres 1-4	profile		253	493	253	0
9	10:29:28.891	Pruebas de Stres 1-5	profile		276	495	276	0
10	10:29:28.953	Pruebas de Stres 1-6	profile		216	493	216	0
11	10:29:28.891	Pruebas de Stres 1-1	profile		283	494	283	0
12	10:29:28.891	Pruebas de Stres 1-3	profile		284	494	284	0
13	10:29:28.833	Pruebas de Stres 1-7	Index		439	494	439	197
14	10:29:29.150	Pruebas de Stres 1-2	consulta_api_twitter		188	494	188	0
15	10:29:29.159	Pruebas de Stres 1-4	consulta_api_twitter		203	494	203	0
16	10:29:29.170	Pruebas de Stres 1-6	consulta_api_twitter		193	493	193	0
17	10:29:29.174	Pruebas de Stres 1-1	consulta_api_twitter		202	495	202	0
18	10:29:29.176	Pruebas de Stres 1-3	consulta_api_twitter		201	495	201	0
19	10:29:29.168	Pruebas de Stres 1-5	consulta_api_twitter		228	495	228	0
20	10:29:29.034	Pruebas de Stres 1-8	Index		370	495	370	170
21	10:29:29.272	Pruebas de Stres 1-7	profile		183	495	183	0

Figura 58. Tabla de Resultados en JMeter

5. Configurar un reporte “Gráfico de Resultados”

Muestra un gráfico con los resultados obtenidos de ejecutar las peticiones HTTP especificadas ofreciendo en una gráfica de líneas los valores de tiempo en milisegundos, rendimiento, y los valores de media, mediana y desviación típica.

a) Agregamos un Gráfico de Resultados

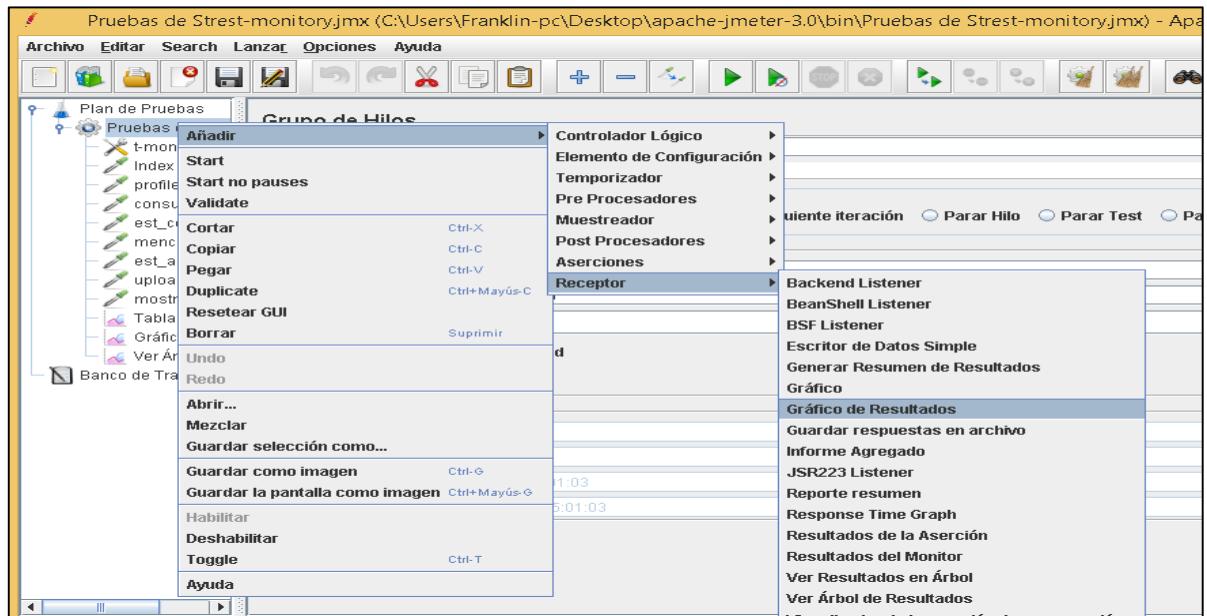


Figura 59. Nueva Gráfico de Resultados en JMeter

Luego de esto procederemos a darle click al botón de la flecha verde (Arrancar) para iniciar nuestra prueba. En ese momento, el grupo de hilos lanzará 50 hilos (en nuestro caso) haciendo 50 peticiones HTTP simultáneas al servidor configurado. Los resultados de cada una de esas peticiones podremos verlo en nuestro elemento Gráfico de Resultados:

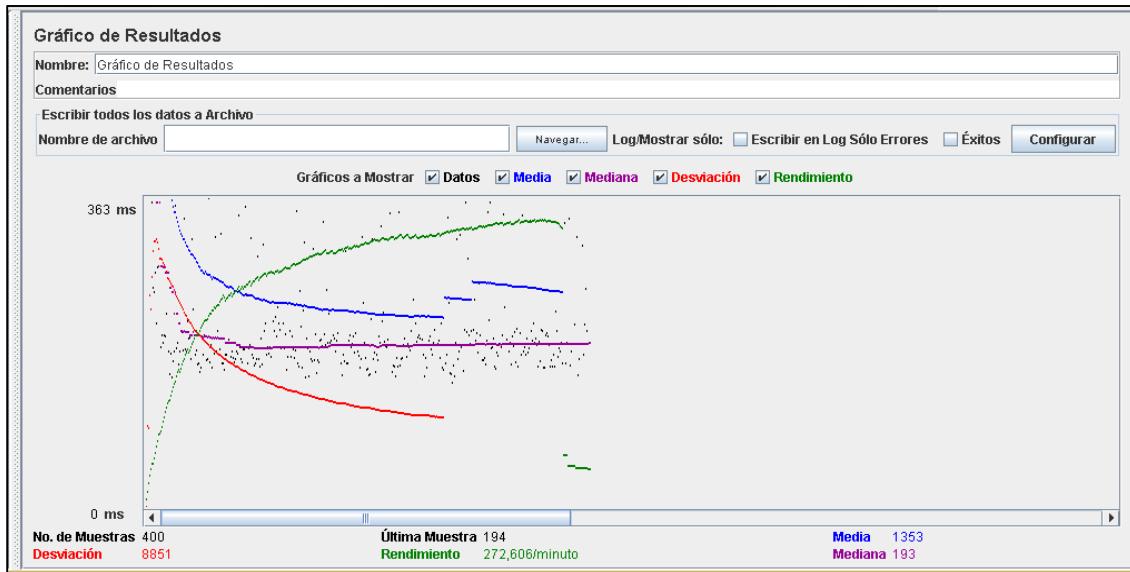


Figura 60. Gráfico de resultados al realizar una petición HTTP

El procedimiento aplicado para monitorizar el rendimiento de las diferentes configuraciones a testear sobre **aplicacioneswebtwitter.com** ha consistido en tomar los datos del Informe de resultados obtenidos a partir de repetidas ejecuciones de peticiones HTTP, escenificando para cada conjunto de parámetros diferentes situaciones de carga, incrementando progresivamente el número de usuarios, así como el tiempo de subida para los mismos o el número de iteraciones. De este modo se obtienen diferentes valores de rendimiento, tiempo medio, máximo y mínimo de ejecución, así como tasa de transferencia en kilobytes/seg, número de Kbs recibidos u otros valores que nos permiten establecer una pauta de comportamiento y analizar dichos resultados desde una perspectiva gráfica.

4.1.2. Pruebas de Usabilidad

La usabilidad es una característica muy importante que debe poseer todo software, t-monitory no es la excepción. Aunque durante el desarrollo de esta aplicación se mantuvo un enfoque basado en permitir el uso a todo tipo de usuarios es importante conocer el grado de usabilidad que permite t-monitory con los usuarios.

La meta de esta prueba de usabilidad es identificar y rectificar las deficiencias de usabilidad y errores existentes. Se intenta asegurar que la aplicación sea fácil de aprender y de usar, su uso sea satisfactorio provea utilidad y funcionalidad y cumpla las tareas para las cuales fue desarrollada la aplicación.

Según el autor Jeffrey Rubin existen cuatro tipos diferentes de pruebas que son: prueba de exploración, prueba de evaluación, prueba de validación y prueba de comparación.

Para realizar este estudio se implementaron los conceptos y metodologías de la llamada Prueba de evaluación. La cual es la más común de todas las pruebas y cuya finalidad es analizar la facilidad con que un usuario puede realizar tareas previamente definidas e identificar las deficiencias de usabilidad que se presenten.

Partiendo de las ideas previamente mencionadas, se diseñó y construyó una prueba de usabilidad para la aplicación t-monitory. Con el fin de examinar en diferentes escenarios y usuarios reales, la capacidad de respuesta de la aplicación. Para llevar a cabo este estudio se le aplicó una encuesta al jefe del Área de Comunicación del Servicio Integrado de Seguridad Ecu911 de la ciudad de Loja, puesto que es la persona a quien va dirigida la aplicación, el cual luego de probar la aplicación emitió un certificado de aprobación de la aplicación. Ver anexo 6.

4.2. Alojamiento de la aplicación en un servidor

De acuerdo a la arquitectura se proceda a configurar el servidor privado (nodored) de las siguientes características:

- ✓ Sistema Operativo: CentOS 5.5 | 64 bits.
- ✓ Almacenamiento en Disco: 2GB
- ✓ Transferencia: 20GB
- ✓ Memoria RAM: 1 GB

✓ Configuración del Servidor

A continuación, se procede a configurar e archivo **database.php** con el nombre del host, la base de datos, nombre de usuario y contraseña del servidor donde se va a alojar la aplicación:

```
$db['default'] = array(  
    'dsn' => '',  
    'hostname'=>'aplicacioneswebtwitter.com',  
    'username' => '*****',  
    'password' => '*****',  
    'database' => 'bd_twitter',  
    'dbdriver' => 'mysqli',  
) ;
```

Así mismo, se tendrá que configurar el archivo **config.php** con el nombre del dominio a partir del cual construiremos la estructura de URLs con el controlador “**Welcome**”.

```
$config['base_url'] = 'http://aplicacioneswebtwitter.com';
```

Para completar la instalación de CodeIgniter en nuestro host, vamos a editar **routes.php**, aquí definiremos cuál será nuestro controlador por defecto, el cual nos llevará la raíz de nuestro dominio.

```
$route['default_controller'] = 'welcome';  
$route['404_override'] = '';  
$route['translate_uri_dashes'] = FALSE
```

g. Discusión

1. Desarrollo de la propuesta alternativa

El Servicio Integrado de Seguridad ECU 911 tiene como misión el brindar atención en todo tipo de emergencia y desastre; para lograr este objetivo, esta empresa ha implementado un sistema de video-vigilancia en sectores estratégicos de la ciudad de Loja, con la finalidad de procurar un mayor control y rapidez en la prestación de este auxilio. La línea habilitada y única para emergencias de acceso a este servicio es el 911.

En la actualidad, el uso de las redes sociales por parte de las Instituciones o Empresas es fundamental para dar a conocer sus servicios, conocer opiniones e interactuar con el público; esto no es una excepción para el Servicio Integrado de Seguridad ECU 911 que también posee cuentas oficiales en Twitter y Facebook, para ofrecer información al público.

Twitter es una herramienta muy poderosa con gran acogida, en Ecuador existen más de un millón de usuarios, a través del cual se difunde cualquier noticia con gran rapidez. Por ello, las personas no dudan en hacer conocer de algún percance, accidente, delito, entre otros mediante esta red social; muchas veces prefieren etiquetar al ECU 911, antes que llamar a su línea única para emergencias.

Por ende, esto ocasiona que muchas emergencias no sean atendidas o el ECU911 no brinde su apoyo a tiempo, puesto que las personas encargadas de las redes sociales no ofrecen el mismo servicio que los operadores.

Ante esto, se evidencia la necesidad de monitorear constantemente la red social Twitter, puesto que es la cuenta más activa del ECU 911, por lo que evidencia la necesidad de crear una herramienta que gestione la cuenta Twitter, con la finalidad de brindar un mejor servicio. Cabe mencionar que en el mercado existen herramientas que brindan este servicio, pero pagado. Los problemas que presenta en la actualidad el Servicio Integrado de Seguridad ECU 911 al ofrecer sus servicios es la falta de atención oportuna a emergencias, debido a que esta información es publicada en una red social y no a través de la línea de emergencias.

Otro problema es el desconocimiento de la situación real de accidentes, puesto que sus estadísticas se basan en las llamadas telefónicas y no en las emergencias publicadas en las redes sociales; información que sería importante para mejorar los servicios que ofrecen.

1.1. Objetivo específico 1: Obtener y analizar la meta data necesaria para el desarrollo de la aplicación web.

Para llevar a cabo este objetivo se realizó una fase de análisis, donde se recabó información para poder obtener la data necesaria para el desarrollo de este tipo de aplicación, es así que se buscó información referente al api de Twitter puesto que esta nos permite acceder a la información de una cuenta de Twitter.

Mediante la utilización de la librería Twconnect se pudo acceder a todas las funciones del API de Twitter pudiendo así obtener la información deseada.

1.2. Objetivo específico 2: Determinar las herramientas necesarias para el desarrollo de la aplicación web.

Este objetivo se llevó a cabo tomando en cuenta las comparativas que se encuentran en la revisión bibliográfica, mediante estas se pudo determinar las mejores herramientas para desarrollar este tipo de aplicaciones.

Para el desarrollo de la aplicación se tomó en cuenta el lenguaje de programación PHP, puesto que es uno de los lenguajes más fáciles de comprender y es el que mejor se adapta a este tipo de desarrollo, además se eligió a Codeigniter como framework para este lenguaje el cual nos permitió desarrollar de manera más fácil y rápida. También se utilizó Xampp como servidor local.

1.3. Objetivo específico 3: Establecer la herramienta adecuada para el almacenamiento de los datos.

En base a la comparativa realizada en el apartado de la revisión bibliográfica se ha dado cumplimiento con este objetivo, ya que se ha seleccionado a MYSQL como el motor de base de datos indicado para el almacenamiento de los datos de la aplicación ya que

muestra algunas ventajas como es el uso de pocos recursos del sistema con relación a otros gestores de bases de datos, así mismo al momento de realizar las consultas a la base de datos nos dio un gran rendimiento y velocidad, además se pudo constatar que se integra de muy buena forma con PHP, tomando en cuenta que este fue el lenguaje con el que se desarrolló la aplicación.

1.4. Objetivo específico 4: Diseñar la aplicación web para el análisis estadístico y monitoreo de una cuenta de Twitter.

El cumplimiento de este objetivo, fue posible gracias al API de Twitter y a la integración de la librería Twconnect, puesto que mediante estas dos herramientas se pudo crear las funciones necesarias para la obtención de la información de una cuenta de Twitter.

Otra de las herramientas importantes que ayudó al cumplimiento de este objetivo fue la librería Highcharts, mediante la cual pudimos integrar los datos obtenidos desde el API de Twitter para mostrarlos en diagramas de barras y pasteles. Así mismo, utilizó plantillas bootstrap para el diseño de las interfaces de la aplicación.

Finalmente, la fase de pruebas permitió comprobar el funcionamiento de los procesos desarrollados como parte del cumplimiento de los objetivos.

h. Conclusiones

Luego de haber concluido con el presente trabajo de titulación se ha llegado a las siguientes conclusiones.

- ✓ Al momento de la recopilación de la información referente al lenguaje de programación para desarrollar la aplicación se pudo evidenciar que, el lenguaje de programación PHP frente a otros lenguajes como Java, tiene una extensa variedad de librerías para el desarrollo de aplicaciones web, especialmente para interactuar y gestionar de manera eficaz una cuenta de Twitter a través de su API.
- ✓ A través de las API de Twitter no solo se puede crear aplicaciones que permiten el análisis y monitoreo de la información de esta red social, también se pueden crear nuevas aplicaciones que permitan realizar un análisis más profundo de los datos.
- ✓ La implementación de la librería Twconnect al desarrollar la aplicación fue de gran ayuda, puesto que mediante sus métodos se pudo conectar con el API de Twitter de manera rápida optimizando el tiempo de desarrollo.

i. Recomendaciones

- Al desarrollador, incorporar nuevas funcionalidades a la aplicación, con el fin de que se realice una buena gestión de Twitter y así administrar de manera correcta la información que se comparte mediante esta red social.
- Al usuario, interactuar activamente con sus cuentas proporcionando información real, relevante y pertinente garantizando así un correcto funcionamiento de la aplicación desarrollada.
- Al desarrollador, investigar profundamente los métodos del API de Twitter se debe tener claro cuál es su funcionalidad, para que al momento de aplicarlos éstos no vayan en contra de las políticas de Twitter y así evitar futuros inconvenientes.

j. Bibliografía

- [1] "PLATICAR," [Online]. Available: <http://www.platicar.go.cr/glosario/45-informatica/286-pagina-web>. [Accessed 22 Septiembre 2015].
- [2] R. S. Porras, "ithinkweb," [Online]. Available: <http://www.ithinkweb.mx/capacita/internet.html>. [Accessed 22 Octubre 2015].
- [3] "Guía Breve sobre Estándares Web," [Online]. Available: <http://www.w3c.es/Divulgacion/GuiasBreves/Estandares>. [Accessed 22 Septiembre 2015].
- [4] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," pp. 2-3, 2014.
- [5] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," p. 7, 2014.
- [6] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," p. 8, 2014.
- [7] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," p. 10, 2014.
- [8] A. Paola Bográn, J. L. Alonso Berrocal and L. Carlos García, "Análisis Léxico sobre los Tweets de Twitter," p. 24, 2013.
- [9] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," p. 12, 2014.
- [10] Twitter, "Twitter Developer Documentation," [Online]. Available: <https://dev.twitter.com/rest/public/search>. [Accessed 22 Junio 2015].
- [11] Twitter, "Twitter Developer Documentation," [Online]. Available: <https://dev.twitter.com/rest/public>. [Accessed 23 Junio 2015].
- [12] Twitter, "Twitter Developer Documentation," [Online]. Available: <https://dev.twitter.com/streaming/overview>. [Accessed 29 Junio 2015].
- [13] H. Russo, "Twitonomy, poderosa herramienta de analíticas de Twitter," 12 Diciembre 2012. [Online]. Available: <http://geeksroom.com/2012/12/twitonomy-poderosa-herramienta-de-analiticas-de-twitter/69711/>. [Accessed 09 Junio 2015].
- [14] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," p. 13, 2014.

- [15] "La Institución," [Online]. Available: <http://www.ecu911.gob.ec/la-institucion/>. [Accessed 02 Mayo 2015].
- [16] "Entidades Coordinadas," [Online]. Available: <http://www.ecu911.gob.ec/entidades-coordinadas/>. [Accessed 05 Mayo 2015].
- [17] [Online]. Available: <http://www.cruzroja.org.ec/>. [Accessed 08 Mayo 2015].
- [18] [Online]. Available: <http://www riesgos.gob.ec/>. [Accessed 08 Mayo 2015].
- [19] [Online]. Available: <http://www.salud.gob.ec/>. [Accessed 08 Mayo 2015].
- [20] [Online]. Available: <http://www.iess.gob.ec/>. [Accessed 08 Mayo 2015].
- [21] [Online]. Available: <http://www.policiaecuador.gob.ec/>. [Accessed 09 Mayo 2015].
- [22] [Online]. Available: <http://www.defensa.gob.ec/>. [Accessed 09 Mayo 2015].
- [23] [Online]. Available: <http://www.ecu911.gob.ec/numerosancionado/>.
- [24] [Online]. Available: <http://www.ecu911.gob.ec/aplicacionparacelulares/>.
- [25] [Online]. Available: <http://www.ecu911.gob.ec/localizacionmovil/>.
- [26] [Online]. Available: <http://www.ecu911.gob.ec/correseguro2/>.
- [27] [Online]. Available: <http://www.ecu911.gob.ec/correseguro2/>.
- [28] [Online]. Available: <http://www.ecu911.gob.ec/localizacionmovil/>. [Accessed 12 Agosto 2015].
- [29] [Online]. Available: <http://www.ecu911.gob.ec/aplicacionparacelulares/>. [Accessed 18 Agosto 2015].
- [30] [Online]. Available: <http://www.ecu911.gob.ec/feriadoseguro2/>. [Accessed 13 Agosto 2015].
- [31] [Online]. Available: <http://www.ecu911.gob.ec/correseguro2/>. [Accessed 15 Agosto 2015].
- [32] MYSQL, "MYSQL Community Server," [Online]. Available: <http://trabajosfce.blogspot.com/2010/06/mysql.html>. [Accessed 26 Septiembre 2015].
- [33] POSTGRESQL, "Postgresql," [Online]. Available: <https://www.postgresql.org/about/>. [Accessed 12 Octubre 2015].
- [34] POSTGRESQL, "Informática," 2011. [Online]. Available: <https://microbuffer.wordpress.com/2011/05/04/que-es-postgresql/>. [Accessed 16 Octubre

2015].

- [35] NICERMA, "Un poco de Internet, Java vs PHP," 2014. [Online]. Available: <https://www.nicearma.com/2015/09/08/java-vs-php/>. [Accessed 11 Diciembre 2015].
- [36] R. G. L. (RGL), "Web, El lenguaje de programación PHP," 2013. [Online]. Available: <http://redgrafica.com/El-lenguaje-de-programacion-PHP>. [Accessed 25 Octubre 2015].
- [37] J. A. MORALES DE LA TORRE, "APLICACIÓN DISTRIBUIDA WEB-MÓVIL ADMINISTRABLE PARA LA GESTIÓN Y DIFUSIÓN GEO-LOCALIZADA DE ATRACTIVOS TURÍSTICOS Y HOTELES PARA LA CIUDAD DE IBARRA, CON TECNOLOGÍA GIS Y SOFTWARE LIBRE," pp. 35-36, 2013.
- [38] "Soluciones en Internet," 11 Marzo 2013. [Online]. Available: <https://enboliviacom.wordpress.com/2013/03/01/highcharts-libreria-para-creacion-de-graficos/>. [Accessed 01 Septiembre 2015].
- [39] W. Lara, "Cómo funciona la Metodología de trabajo Scrum," [Online]. Available: <https://platzi.com/blog/guia-scrum/>. [Accessed 22 Septiembre 2015].
- [40] "METODOLOGIAS AGILES RUP," [Online]. Available: http://ingenieriadessoftware.mex.tl/52788_Rup-Agil.html. [Accessed 10 Mayo 2015].
- [41] "METODOLOGIA XP," 2012. [Online]. Available: <http://ingsoftware072301.obolog.es/metodologia-xp-2012877>. [Accessed 18 Noviembre 2015].
- [42] K. Schwaber and . J. Sutherland, "La Guia de Scrum," pp. 9-10, 2013.
- [43] Á. L. Ortiz Folgado, Ó. E. Pérez la Madrid and E. Vargas Rastrollo, "ESTUDIO DE TENDENCIAS DIARIAS EN TWITTER," pp. 2-3, 2014.

k. Anexos

1. Anexo 1: Artículo Científico

Análisis estadístico y monitoreo de una cuenta de Twitter

Franklin de Jesús Michay Cuenca, Ing. Luis Roberto Jácome Galarza, Mg. Ms.

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Universidad Nacional de Loja

franklinmichay@gmail.com, lrjacome@unl.edu.ec

Resumen- Este artículo presenta una aplicación web que permitirá a los usuarios obtener información estadística y monitorear una cuenta de Twitter. La aplicación muestra estadísticas diarias de las menciones, tweets, retweets, y seguidores así mismo, mediante la creación de patrones de búsqueda se monitorea la actividad diaria de Twitter.

Abstract- This article presents a web application that will allow users to obtain statistical information and monitor a Twitter account. The app displays daily statistics of mentions, tweets, retweets, and followers as well, by creating search patterns monitored daily activity of Twitter.

/Keywords- análisis, monitoreo, alerta, estadística, publicar tweets, Twitter

I. INTRODUCCIÓN

En la actualidad el uso de las redes sociales por parte de las instituciones públicas o privadas es fundamental para dar a conocer sus servicios, opiniones e interactuar con el público. En nuestro país existen más de un millón de usuarios de Twitter a través del cual se difunden noticias con gran rapidez, es por esto que las personas no dudan en hacer conocer algún percance, accidente, delito, etc, mediante esta red social; muchas de las veces prefieren etiquetar a las entidades de socorro, antes que llamar a su líneas únicas para emergencias esto ocasiona que muchas emergencias no sean atendidas a tiempo. Es por ello que se plantea el desarrollo de una aplicación que permita el seguimiento de la actividad de una cuenta de Twitter para optimizar la atención de la información que circula por esta red social.

Además los usuarios podrán visualizar estadísticas de la data (seguidores, siguiendo, tweets, menciones, etc) de una cuenta de Twitter.

La aplicación basa su funcionamiento en los servicios ofrecidos por Twitter como es el API REST de Twitter. Para el desarrollo de la aplicación web se

han establecido las siguientes fases: exploración, planeación, iteraciones y producción.

II. ESTADO DEL ARTE

A. Aplicaciones web de análisis de Twitter

Las aplicaciones web de análisis de datos y monitoreo permiten realizar el seguimiento de toda la actividad de Twitter analizando paso a paso la información que circula por esta red social.

Existen un sinnúmero de aplicaciones las cuales están diseñadas para cumplir diferentes objetivos vinculados al análisis y seguimiento de una cuenta de Twitter.

B. Twitter API

El servicio de micro-blogging Twitter incluye dos API REST. Los métodos de la API REST Twitter permiten a los desarrolladores acceder a los datos básicos de Twitter. Esto incluye líneas de tiempo de actualización, datos de estado y la información del usuario. Los métodos de la API de búsqueda dan métodos para interactuar con los desarrolladores de Twitter de búsqueda y tendencias de los datos. La API actualmente soporta los siguientes formatos de datos: XML, JSON, y el RSS.

C. API REST de Twitter

El REST API ofrece a los desarrolladores el acceso al core de los datos de Twitter. Todas las operaciones que se pueden hacer vía web son posibles realizarlas desde su API.

Dependiendo de la operación requiere o no autenticación, con el mismo criterio que en el acceso web. Soporta los formatos: xml, json, rss, atom.

Proporciona acceso mediante programación para leer y escribir datos de Twitter, identifica las aplicaciones y los usuarios que utilizan OAuth de Twitter; las respuestas están disponibles en formato JSON.

III. EXPERIMENTOS

La aplicación web para el análisis y monitoreo de una cuenta de Twitter tiene como funciones el visualizar información general de una cuenta de Twitter; generar estadísticas de los seguidores, siguiendo, tweets, retweets y menciones; publicar y programar tweets.



Figura 1. Menú de navegación principal de la aplicación

IV. RESULTADOS

A. Exploración

Para obtener la información se utilizó el API REST de Twitter mediante sus métodos que permiten a los desarrolladores acceder a los datos de una cuenta de Twitter.

Las estadísticas de los tweets y retweets se generaron de acuerdo al día, semana y mes. El monitoreo de los tweets se lo realiza a través de la creación de patrones de búsqueda, para la publicación y programación de tweets se utilizó el MEDIA API de Twitter el cual permite la publicación de contenido multimedia mediante la creación de un identificador (para que sea reconocido por Twitter) para cada objeto multimedia que se quiera adjuntar a la publicación y/o programación.

Para determinar los requerimientos se crearon las historias de usuario con las cuales se pudo entender de manera más clara las necesidades de los usuarios.

B. Planeación

Se crearon historias de usuario a las cuales se les estimó un tiempo necesario para llevarlas a cabo, a partir de allí se definió un cronograma de actividades.

Para cada historia de usuario se crearon actividades que permitieron cumplir con el cronograma fijado.

C. Iteraciones

La aplicación se realizó en tres iteraciones, en cada iteración se desarrollaron varias historias de usuario. La

codificación se realizó en Sublime Text con la utilización del framework Codeigniter.

A la aplicación se accederá mediante la utilización de una cuenta de Twitter la cual permitirá obtener a los datos mediante los métodos de su API.

La información obtenida se almacenará en una base de datos desde donde se tomarán los datos para su análisis.

- Acceso a la información de Twitter

La información de Twitter se la obtiene mediante la utilización de la librería **twconnect**, cada función guarda un determinado lote de datos en formato JSON el cual para su manejo se lo descompone mediante funciones PHP.

```
public function twaccount_verify_credentials() {
    $this->tw_user_info = $this->tw_get('account/verify_credentials');
    return $this->tw_user_info;
}

public function user_time_line() {
    $this->tw_user_tw = $this->tw_get('statuses/user_timeline', ["count"=>200]);
    return $this->tw_user_tw;
}

public function home_time_line() {
    $this->tw_user_tw = $this->tw_get('statuses/home_timeline', ["count"=>200]);
    return $this->tw_user_tw;
}

public function tw_mentions() {
    $this->tw_user_tw = $this->tw_get('statuses/mentions_timeline');
    return $this->tw_user_tw;
}

public function tw_friends_list() {
    $this->tw_user_tw = $this->tw_get('friends/list', ["count"=>200]);
    return $this->tw_user_tw;
}

public function tw_followers_list() {
    $this->tw_user_tw = $this->tw_get('followers/list', ["count"=>200]);
    return $this->tw_user_tw;
}

public function buscar($palabra) {
    $all_tw = $this->tw_get("search/tweets", ["q"=>$palabra]);
    return $this->all_tw;
}

public function busca_statuses_retweets_of_me() {
    $rt_of_me = $this->tw_get('statuses/retweets_of_me.json');
    return $this->rt_of_me;
}
```

- Convertir el formato JSON de Twitter en un arreglo PHP para poder almacenarlos en la base de datos relacional.

- a) Guardar tweets de la comunidad de un usuario de Twitter.

```
public function guardar_tw(){
    $json = $this->twconnect->home_time_line();
    $rows = $json;

    for($i=0;$i<count($rows);$i++){
        $data = array('id_tweet' => $rows[$i]->id_str,
                    'text' => $rows[$i]->text,
                    'retweet_count' => $rows[$i]->retweet_count,
                    'favorite_count' => $rows[$i]->favorite_count,
                    'created_at' => date('Y-m-d', strtotime($rows[$i]->created_at)),
                    'created_at_time' => date('H:i:s', strtotime($rows[$i]->created_at)),
                    'screen_name' => $rows[$i]->user->screen_name,
                    'profile_image_url' => $rows[$i]->user->profile_image_url,
                    'followers_count' => $rows[$i]->user->followers_count,
                    'followings_count' => $rows[$i]->user->friends_count,
                    'statuses_count' => $rows[$i]->user->statuses_count,
                    'fecha_registro' => date('Y-m-j'),
                    'id_user_login' => $this->session->userdata('id_login'));

        $resp = $this->Persona->verifica_tw($rows[$i]->id_str);

        if($resp){
            }else{
                $this->Persona->guardar($data);
            }
        }
    }
}
```

b) Guardar menciones de un usuario de Twitter.

```

public function mentions(){
    $contm = 0;
    $json= $this->twconnect->tw_mentions();
    $json = json_encode($json);
    $json = json_decode($json, TRUE);
    $row = $json;

    if (count($rows)>0) {
        foreach ($rows as $valor) {
            $data = array('id_user' => $valor['id_str'],
                'text' => $valor['text'],
                'name' => $valor['user']['name'],
                'screen_name' => $valor['user'][['screen_name']],
                'created_at' => date('Y-m-d', strtotime($valor['created_at'])),
                'favorite_count' => $valor['user'][['favourites_count']],
                'profile_image_url' => $valor['user'][['profile_image_url']],
                'friends_count' => $valor['user'][['friends_count']],
                'followers_count' => $valor['user'][['followers_count']],
                'statuses_count' => $valor['user'][['statuses_count']],
                'register' => date('Y-m-j'),
                'retweet_count' => $valor['retweet_count'],
                'time_zone' => $valor['user'][['time_zone']],
                'id_user_login' => $this->session->userdata('id_login'));
            $resp = $this->Persona->verifica_mentions($valor['id_str']);
            if(!$resp){
                $this->Persona->guardar_mentions($data);
                $contm++;
            }
        }
    }
}

```

D. Producción

En esta fase de realizaron pruebas las cuales comprobaron el funcionamiento y el correcto rendimiento de la aplicación. Así mismo, se determinaron las características del servidor donde se aloja la aplicación.

- Visualización de la información general de una cuenta de Twitter.

La aplicación permite visualizar la información general de una cuenta de Twitter (tweets, seguidores, siguiendo, favoritos, ubicación, etc.).



Figura 2. Perfil de usuario de una cuenta de Twitter

- Visualización de estadísticas de la aplicación.

La aplicación permite visualizar estadísticas de los tweets, retweets y menciones. La estadística de los tweets se visualizarán de acuerdo al número: por día, mes y al seleccionar un rango de fechas.

Actividad de tu cuenta de Twitter - Resumen por día



Figura 3. Estadísticas diarias de t-monitor

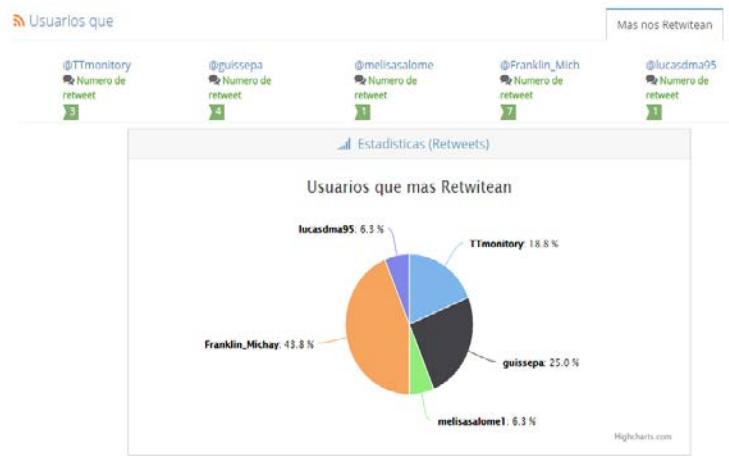


Figura 4. Estadísticas de t-monitor: Usuarios que más Retuitean



Figura 5. Estadísticas t-monitor: Tweets por mes

- Programación y publicación de tweets.

La aplicación permitirá la publicación y/o programación de tweets.

Figura 6. Publicar tweets

Programar Tweet

Texto:
Máximo 140 caracteres

Imagen:
[Seleccionar archivo] Ningún archivo seleccionado

Selección el día
dd ---- aaaa

Seleccione las horas en las que se debe programar el tweet

Hora en la mañana: 00:00 ▾
Hora en la Tarde: 13:00 ▾
Hora en la Noche: 19:00 ▾

Twittear

Cancelar

Figura 7. Programar tweets

V. CONCLUSIONES

- Mediante la creación de alertas se pudo determinar de manera rápida la localización de las emergencias que se publican en la cuenta de Twitter del Sistema Integrado de Seguridad ECU911 Loja.
- La programación de tweets permite interactuar de manera activa con los usuarios de Twitter.
- Interactuar de forma activa con una cuenta de Twitter, permite el correcto funcionamiento de la aplicación t-monitor, puesto que la información que proporcionan los usuarios es la que permite obtener estadísticas, seguimiento y monitoreo con mayor exactitud.

VI. TRABAJOS FUTUROS

A la presente aplicación se le pueden agregar las siguientes características que ayuden a optimizar su eficiencia:

- Análisis semántico para la búsqueda de información en Twitter: análisis semántico de los tweets para que sean automáticamente procesados y así poder distinguir la información relevante y lograr un mayor nivel de abstracción.
- Aplicado a varias cuentas de Twitter: realizar el análisis, seguimiento y mostrar estadísticas de varias cuentas de Twitter.

VII. REFERENCIAS

- [1] GCFAprendeLibre, [en linea], <https://www.gcfaprendelibre.org>
- [2] Programable Web, [en linea], <http://www.programmableweb.com/api/twitter>
- [3] TweetAlert: Sistema de Análisis Semántico de la Voz de los Ciudadanos en Redes Sociales en la Ciudad del Futuro, [en linea], <http://oa.upm.es/35357>
- [4] Análisis de contenidos en Twitter: clasificación de mensajes e identificación de la tendencia política de los usuarios, [en linea], <http://www.grupolys.org/biblioteca/Vil2014a.pdf>



Franklin de Jesús Michay Cuenca Egresado de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, Provincia Loja, Ciudad Loja, Ecuador 2017



Luis Roberto Jácome Galarza Ingeniero en Sistemas, Docente de la Carrera de Ingeniería en Sistemas, Provincia Loja, Ciudad Loja, Ecuador 2017

2. Anexo 2: Licencia Creative Commons



Desarrollo de una aplicación web de análisis estadístico y monitoreo de una cuenta de Twitter by Michay Cuenca Franklin de Jesús is licensed under a [Creative Commons Reconocimiento-NoComercial 4.0 Internacional License](#).

3. Anexo 3: Anteproyecto



UNIVERSIDAD
NACIONAL
DE LOJA

CIS-IXB-001



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“Desarrollo de una aplicación web de análisis estadístico y monitoreo de una cuenta de Twitter”

TRABAJO DE TITULACIÓN

Autor:

- Franklin de Jesús Michay Cuenca

Docente:

- Ing. Ana Colala

LOJA-ECUADOR

2015

ÍNDICE

ÍNDICE-----	2
A. Tema-----	3
B. Problemática -----	4
C. Justificación-----	6
D. Objetivos-----	7
1. Objetivo General: -----	7
2. Objetivos Específicos:-----	7
E. Alcance-----	8
1. FASE 1: ANÁLISIS -----	8
2. FASE 2: DISEÑO -----	8
2. ETAPA 3: CODIFICACIÓN-----	9
3. ETAPA 4: PRUEBAS-----	9
4. ETAPA 5: IMPLEMENTACIÓN-----	9
F. Metodología-----	10
1. Metodología-----	10
2. Métodos -----	10
3. Técnicas-----	10
G. Cronograma -----	11
H. Presupuesto y Financiamiento -----	14
1. Presupuesto -----	14
2. Financiamiento -----	15
I. Bibliografía-----	15

A. Tema

***“DESARROLLO DE UNA APLICACIÓN WEB DE ANÁLISIS ESTADÍSTICO Y
MONITOREO DE UNA CUENTA DE TWITTER”***

B. Problemática

1. Situación Problemática.

El Servicio Integrado de Seguridad ECU 911 tiene como misión el brindar atención en todo tipo de emergencia y desastre [1]; para lograr este objetivo, esta empresa pública ha implementado un sistema de video-vigilancia en sectores estratégicos de la ciudad de Loja, con la finalidad de procurar un mayor control y rapidez en la prestación de este auxilio. La línea habilitada y única para emergencias de acceso a este servicio es el 911[2].

En la actualidad, el uso de las redes sociales por parte de las Instituciones o Empresas es fundamental para dar a conocer sus servicios, conocer opiniones e interactuar con el público; esto no es una excepción para el Servicio Integrado de Seguridad ECU 911 que también posee cuentas oficiales en Twitter y Facebook, para ofrecer información al público.

Twitter es una herramienta muy poderosa con gran acogida, en Ecuador existen más de un millón de usuarios [3], a través del cual se difunde cualquier noticia con gran rapidez. Por ello, las personas no dudan en hacer conocer de algún percance, accidente, delito, entre otros mediante esta red social; muchas veces prefieren etiquetar al ECU 911, antes que llamar a su línea única para emergencias.

Por ende, esto ocasiona que muchas emergencias no sean atendidas o el ECU911 no brinde su apoyo a tiempo, puesto que las personas encargadas de las redes sociales no ofrecen el mismo servicio que los operadores.

Ante esto, se evidencia la necesidad de monitorear constantemente la red social Twitter, puesto que es la cuenta más activa del ECU 911, a través de una herramienta que gestione la cuenta de Twitter, con la finalidad de brindar un mejor servicio. Cabe mencionar que en el mercado existen herramientas que brindan este servicio pero pagado, tales herramientas se mencionan a continuación:

- ✓ Twitonomy es una herramienta de monitoreo y análisis que le permite medir las menciones, alcance, e incluso las conversiones.
- ✓ Retweetrank: muestra la cantidad de re-tweets que haces y de tus seguidores.
- ✓ Twitterverse: informes de las palabras clave más utilizadas en Twitter.
- ✓ Twinfluence: una forma de medir la influencia social de una cuenta Twitter.
- ✓ Mr. Milestone: envía un mensaje por cada 100 seguidores nuevos para saber si está funcionando tu estrategia social.

Los problemas que presenta en la actualidad el Servicio Integrado de Seguridad ECU 911 al ofrecer sus servicios es la falta de atención oportuna a emergencias, debido a que muchas de las veces la ciudadanía publica esta información publicada en las redes sociales y no a través de la línea de emergencias.

Otro problema es el desconocimiento de la situación real de accidentes, puesto que sus estadísticas se basan en las llamadas telefónicas y no en las emergencias publicadas en las redes sociales; información que sería importante para mejorar los servicios que ofrecen.

2. Problema de Investigación

En base a la problemática anteriormente descrita se ha planteado el siguiente problema:
¿El desconocimiento de las emergencias publicadas en la cuenta del ECU 911 de la red social Twitter ocasiona demora en su atención o que éstas no sean atendidas?

C. Justificación

Unos de los principales motivos de la creación de esta herramienta Web es permitir al personal del departamento de comunicación tener un mayor control de la información que se maneja a través de la cuenta de Twitter para procesarla con mayor rapidez. Es por esto que en este trabajo de tesis se pretende crear una herramienta que permita llevar de forma automatizada las actividades que se realizan en la cuenta de Twitter del Sistema Integrado de Seguridad Social ECU 911, permitiendo al personal del departamento de Comunicación de esta entidad, en un menor tiempo, con menor cantidad de errores y control, llevar el control de los reportes estadísticos de la actividad de las cuentas del Twitter, tomando en consideración que la red social Twitter es la herramienta más utilizada para estar con contacto con la comunidad, y así poder cubrir las emergencias suscitadas de forma eficiente y eficaz. Con el desarrollo de esta aplicación Web, se pretende obtener datos reales y exactos de la información ingresada a través de los tweets, los mismos que mediante la Web serán almacenados en la base de datos de la aplicación, lo que permitirá conocer los reportes inmediatamente después de que la información llegue a conocimiento del ECU 911 a través de esta red social, evitando así la pérdida de tiempo que toma el proceso manual de filtrar la información pertinente y del tiempo del personal que lo va a realizar.

Las funcionalidades que se cubrirán con la aplicación son:

- ✓ Reportes estadísticos de la actividad de una cuenta Twitter.
- ✓ Permitirá realizar búsquedas personalizadas mediante los hashtags, palabras más utilizadas, seguidores, nombre de alguna entidad, fecha y hora de los tweets.
- ✓ Alertar mediante correo electrónico o notificación cada vez que la cuenta Twitter alcance un hito (acontecimiento de importancia).
- ✓ Reportes estadísticos de accidentes falsos, verdaderos y atendidos por el Sistema Integrado de Seguridad ECU911.

En la elaboración del presente proyecto se pondrá en práctica los conocimientos adquiridos a lo largo de la carrera, como es el análisis y diseño de sistemas para la elaboración de una aplicación; de la misma forma se adquirirán conocimientos para

conectar el API de Twitter con la aplicación, lo cual nos va a permitir obtener información relevante de la cuenta de Twitter para el desarrollo de la misma.

D. Objetivos

1. Objetivo General:

Desarrollar una aplicación web que permita realizar un análisis estadístico y monitoreo de la actividad de las cuentas de Twitter asociado al Sistema Integrado de Seguridad ECU911.

2. Objetivos Específicos:

- ✓ Obtener y analizar la meta data necesaria para el desarrollo de la aplicación web.
- ✓ Determinar las herramientas necesarias para el desarrollo de la aplicación web.
- ✓ Establecer la herramienta adecuada para el almacenamiento de los datos.
- ✓ Diseñar la aplicación web para el análisis estadístico y monitoreo de una cuenta de Twitter.

E. Alcance

1. FASE 1: ANÁLISIS

- ✓ **Recolectar y analizar información:** se buscará información de los problemas existentes en el Sistema Integrado de Seguridad ECU911 con respecto al uso de las redes sociales, específicamente Twitter, para en base a esta información analizar los aspectos necesarios para el desarrollo de la aplicación.
- ✓ **Analizar las herramientas necesarias para el desarrollo de la aplicación:** se analizarán las herramientas necesarias para el correcto desarrollo de la aplicación.
- ✓ **Determinar y elaborar el documento de requerimientos:** una vez que se haya analizado la información recolectada se determinará cuáles son los requerimientos funcionales y no funcionales que debe cumplir la aplicación.

2. FASE 2: DISEÑO

- ✓ **Elaborar el modelo del dominio:** este diagrama permitirá conocer la relación entre los objetos de interrelación en la aplicación y su funcionamiento.
- ✓ **Elaborar el diagrama de Casos de Uso:** Al elaborar este diagrama se determinarán cuáles son los casos de uso que se darán en la aplicación y que permitirán cumplir con los requerimientos.
- ✓ **Elaborar los Prototipos de Pantalla:** En esta actividad se diseña la interfaz y el comportamiento de la aplicación.
- ✓ **Determinar y elaborar el diagrama de actividades:** Con el diagrama de actividades se busca determinar los eventos y principales actividades de la aplicación.
- ✓ **Determinar y elaborar el diagrama de Colaboración:** Con este diagrama se busca determinar todos los entes que intervienen en la aplicación, agregando detalles extras al momento del paso de mensajes entre objetos.
- ✓ **Determinar y elaborar el Diagrama de Robustez:** Este diagrama permite determinar los diferentes tipos de objetos que intervienen (fronterizos, de control y de entidad) que permitirán entender de mejor manera los casos de uso.

2. ETAPA 3: CODIFICACIÓN

- ✓ **Realizar la conexión de la aplicación con la API de Twitter y la Base de Datos:** Se conectará la aplicación con la API de Twitter para determinar los datos de las cuentas y así poder enlazar esa información a las bases de datos.
- ✓ **Desarrollar la aplicación para el análisis estadístico del Twitter:** Se realizará la codificación de la aplicación de acuerdo al comportamiento diseñado y que permita obtener la información estadística acerca de las cuentas de Twitter.

3. ETAPA 4: PRUEBAS

- ✓ **Realizar las pruebas de ejecución:** En este tipo de pruebas se buscará comprobar el correcto funcionamiento de la aplicación en cuanto a la navegabilidad, sin el ingreso de datos.
- ✓ **Realizar las pruebas de resultados:** Para esta prueba se introducirán datos y se testeará el correcto funcionamiento de la aplicación.

4. ETAPA 5: IMPLEMENTACIÓN

- ✓ **Implementación de la aplicación en los servidores del Sistema Integrado de Seguridad ECU911:** se instalará la aplicación en los servidores del Sistema Integrado de Seguridad ECU911 para ponerla en producción.

F. Metodología

Para el desarrollo del presente proyecto se utilizará diferentes métodos y técnicas que ayuden a lograr cumplir con los objetivos planteados.

1. Metodología

Se plantea la utilización de la metodología ágil ICONIX que permitirá la elaboración de la aplicación de manera rápida y con un nivel de documentación aceptable.

2. Métodos

- ✓ **Método Deductivo:** Se aplicará este método para obtener la información necesaria sobre el uso de la red social Twitter para determinar posibles emergencias y determinar la veracidad de éstas.
- ✓ **Método Deductivo:** A través de este método se plantean las principales dificultades al determinar posibles emergencias mediante el uso de Twitter y determinar si la solución planteada permite resolverlas.

3. Técnicas

- ✓ **Encuesta:** Con esta técnica se podrá obtener información relevante y necesaria referente al tema de la investigación, la cual ayudara a sustentarlo y justificarlo.
- ✓ **Técnica de la investigación bibliográfica:** Con esta técnica se sustentara la base teórica de la investigación, mediante consultas a: fuentes bibliográficas confiables, libros, revistas indexadas, artículos científicos, base de datos científicas entre otras.

G. Cronograma

		Inicio	Término	Duración (Horas)	Nombre del Recurso
1	Desarrollo de una aplicación web de análisis estadístico y monitoreo de una cuenta de Twitter.	04/05/2015 15:00	31/07/2015 21:00	400	Franklin Michay
2	ETAPA 1: ANÁLISIS	04/05/2015 15:00	22/05/2015 21:00	90	Franklin Michay
3	Recolectar y analizar información.	04/05/2015 15:00	08/05/2015 21:00	30	Franklin Michay
4	Analizar las herramientas necesarias para el desarrollo de la aplicación	11/05/2015 15:00	15/05/2015 21:00	30	Franklin Michay
5	Determinar y elaborar el documento de requerimientos	18/05/2015 15:00	22/05/2015 21:00	30	Franklin Michay
6	ETAPA 2: DISEÑO	25/05/2015 15:00	05/06/2015 21:00	60	Franklin Michay
7	Elaborar el modelo del dominio	25/05/2015 15:00	27/05/2015 21:00	18	Franklin Michay
8	Elaborar el diagrama de Casos de Uso	28/05/2015 15:00	29/05/2015 21:00	12	Franklin Michay
9	Elaborar los Prototipos de Pantalla	01/06/2015 15:00	03/06/2015 21:00	18	Franklin Michay
10	Determinar y elaborar el diagrama de actividades	04/06/2015 15:00	05/06/2015 21:00	4	Franklin Michay
11	Determinar y elaborar el diagrama de Colaboración	04/06/2015 15:00	05/06/2015 21:00	4	Franklin Michay
12	Determinar y elaborar el Diagrama de Robustez	04/06/2015 15:00	05/06/2015 21:00	4	Franklin Michay
13	ETAPA 3: CODIFICACIÓN	08/06/2015 15:00	17/07/2015 21:00	210	Franklin Michay
14	Realizar la conexión de la aplicación con el api de Twitter y la Base de Datos.	08/06/2015 14:00	12/06/2015 21:00	35	Franklin Michay
15	Desarrollar la aplicación para el análisis estadístico del Twitter.	15/06/2015 14:00	17/07/2015 21:00	175	Franklin Michay
16	ETAPA 4: PRUEBAS	20/07/2015 14:00	31/07/2015 18:00	40	Franklin Michay
17	Realizar las pruebas de ejecución	20/07/2015 14:00	24/07/2015 18:00	20	Franklin Michay
18	Realizar las pruebas de resultados	27/07/2015 14:00	29/07/2015 18:00	10	Franklin Michay
19	ETAPA 5: IMPLEMENTACIÓN	30/07/2015 14:00	31/07/2015 18:00	10	Franklin Michay
20	Implementación de la aplicación en los servidores del Sistema Integrado de Seguridad ECU911.	30/07/2015 14:00	31/07/2015 18:00	10	Franklin Michay

cronograma

20/04/2017

Tarea

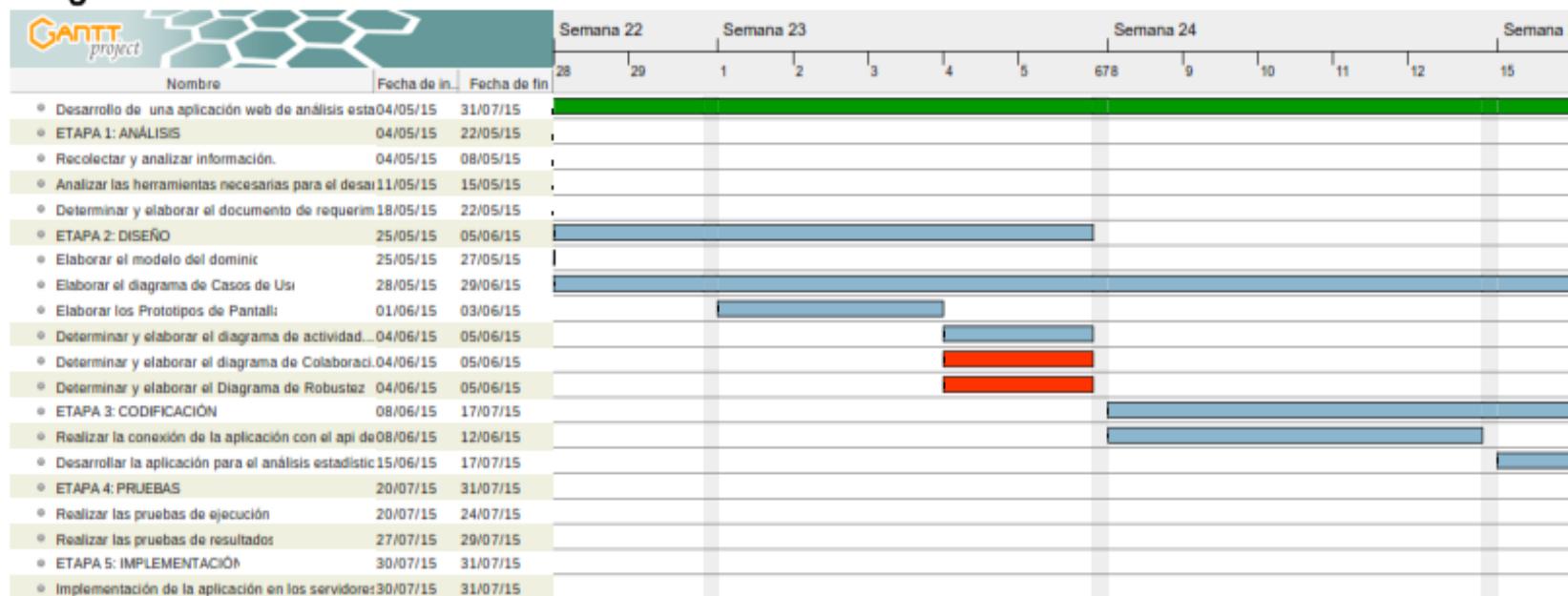
Nombre	Fecha de inicio	Fecha de fin
Desarrollo de una aplicación web de análisis estadístico y monitoreo de una cuenta de twitter.	04/05/15	31/07/15
ETAPA 1: ANÁLISIS	04/05/15	22/05/15
Recolectar y analizar información.	04/05/15	08/05/15
Analizar las herramientas necesarias para el desarrollo de la aplicación	11/05/15	15/05/15
Determinar y elaborar el documento de requerimientos	18/05/15	22/05/15
ETAPA 2: DISEÑO	25/05/15	05/06/15
Elaborar el modelo del dominio	25/05/15	27/05/15
Elaborar el diagrama de Casos de Uso	28/05/15	29/06/15
Elaborar los Prototipos de Pantalla	01/06/15	03/06/15
Determinar y elaborar el diagrama de actividades	04/06/15	05/06/15
Determinar y elaborar el diagrama de Colaboración	04/06/15	05/06/15
Determinar y elaborar el Diagrama de Robustez	04/06/15	05/06/15
ETAPA 3: CODIFICACIÓN	08/06/15	17/07/15
Realizar la conexión de la aplicación con el api de twitter y la Base de Datos.	08/06/15	12/06/15
Desarrollar la aplicación para el análisis estadístico del twitter.	15/06/15	17/07/15
ETAPA 4: PRUEBAS	20/07/15	31/07/15
Realizar las pruebas de ejecución	20/07/15	24/07/15
Realizar las pruebas de resultados	27/07/15	29/07/15
ETAPA 5: IMPLEMENTACIÓN	30/07/15	31/07/15
Implementación de la aplicación en los servidores del Sistema Integrado de Seguridad ECU911.	30/07/15	31/07/15

cronograma

20/04/2017

3

Diagrama de Gantt



H. Presupuesto y Financiamiento

1. Presupuesto

TALENTO HUMANO				
Rol	Horas	Número Integrantes	Remuneración (Hora)	Valor Total
Analista Programador	400	1	8,00	3200,00
TOTAL				3200,00

RECURSOS TÉCNICOS				
Descripción	Horas	Número Integrantes	Valor Unitario	Valor Total
Computador		1	900,00	900,00
Windows		1	150,00	150,00
Internet	400	1	0,60	240,00
Software Libre		1	0,00	00,00
TOTAL				1290,00

RECURSOS MATERIALES				
Material	Cantidad	Número Integrantes	Valor Unitario	Valor Total
Impresiones	600	1	0,05	30,00
Copias	500	1	0,03	15,00
Transporte		1		120,00
TOTAL				165,00

RECURSO	SUBTOTAL
Recurso Humano	3200,00
Recursos Técnicos	1290,00
Recursos Materiales	165,00
Imprevistos	465,50
TOTAL	5120,00

2. Financiamiento

El estudiante asumirá todos los gastos que involucra el Proyecto de Fin de Carrera ya que es de carácter académico.

I. Bibliografía

[1].Misión, Visión Objetivos, [2015], [17/04/2015],
<http://205.186.151.157/es/pagina/misi%C3%B3n-visi%C3%B3n-y-objetivos>.

[2].Ventas y capacidades ECU911, [2015], [17/04/2015],
<http://205.186.151.157/es/pagina/ventas-y-capacidades-ecu-911>.

[3].Twiter en Ecuador, [01/04/2014], [17/04/2015],
<http://www.coberturadigital.com/2014/04/01/twitter-en-ecuador-sobrepasa-la-cifra-del-millon-de-usuarios/>

[4].Twitter 5 años, El recorrido por la herramienta que se convirtió en plataforma, [01/04/2015], [17/04/2015], <http://www.antonioconstantino.com/pdf/twitter.pdf>

4. Autorización del desarrollo del proyecto de titulación y pruebas de campo.

0000050



UNIVERSIDAD
NACIONAL
DE LOJA

OFICIO-CIS-UNL

Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

OF. N° 632 CIS-AEIRNNR-UNL
Loja, 19 de junio de 2015

Ingeniero
Johan Coronel Peñafiel
JEFE LOCAL DEL CENTRO OPERATIVO CIS ECU 911 LOJA.
Ciudad -

De mi consideración:

Con un cordial saludo me dirijo a usted con la finalidad de solicitarle muy comedidamente se sirva dar las facilidades necesarias para que el Sr. FRANKLIN DE JESÚS MICHAY CUENCA, con número de cédula 1104993157, tesis de la Carrera de Ingeniería en Sistemas de la UNL, pueda desarrollar y realizar pruebas de campo para su proyecto de titulación, cuyo tema es "DESARROLLO DE UNA APLICACIÓN WEB DE ANÁLISIS ESTADÍSTICO Y MONITOREO DE UNA CUENTA DE TWITTER".

Por la atención que se digne dar al presente, me suscribo de usted.

Atentamente,

Ing. Luis Roberto Jécome Galarza
COORDINADOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS.



Cdr., Archivo,
Elsa Orellana

5. Encuesta dirigida al jefe del Área de Comunicación del Servicio Integrado de Seguridad Ecu911 de la ciudad de Loja.



UNIVERSIDAD
NACIONAL
DE LOJA



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

Encuesta dirigida Jefe del Área de Comunicación del Servicio Integrado de Seguridad Ecu911 de la ciudad de Loja.

Estimados:

Como egresado de la carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, me encuentro desarrollando mi trabajo de tesis Titulada "Desarrollo de una aplicación web de análisis estadístico y monitoreo de una cuenta de Twitter"; motivo por el cual, le solicito a usted muy comedidamente se digne contestar las siguientes interrogantes:

1. ¿Cuáles son las actividades que realiza dentro del área de su regencia?

Me encargo de las comunicaciones internas y externas de la institución, en el caso de relaciones públicas y visibilización con la ciudadanía.

2. ¿Qué actividad(es) le generan más tiempo realizar?

*Bolígrafo de prensa
Manejo de cuenta de Facebook de la institución*

3. ¿Qué tipo de información maneja (física o digital)?

Digital y física

4. ¿Haciendo un análisis estadístico, respecto de la información que recibe de los medios habilitados para obtener información sobre las emergencias suscitadas en el medio, en qué porcentaje recibe dicha información a través de la cuenta oficial de Twitter del ECU911-Loja?

El porcentaje es bajo ya que la gente no se acuerda de reportar las emergencias

5. ¿Utiliza alguna herramienta para trabajar la información que recibe de la cuenta del Twitter de esta institución?

Excel

6. ¿Realiza usted respaldos de esta información?

Sí No

¿Mediante que herramienta?

Archivos Digitales

7. ¿Cree usted necesaria la implementación de una herramienta que permita gestionar automáticamente la información obtenida de la cuenta del Twitter?

Sí No

Debido a que la especificidad del servicio es 29/7

8. ¿Qué aspectos se debería tomar en cuenta al momento de desarrollar dicha herramienta?

Páginas claves

Menciones

Tipos de alertas

Programación y respuestas

9. ¿Cuáles son las ventajas, que se podrían obtener con el desarrollo de esta aplicación?

Mayor operatividad digital e inmediata al momento de monitorear servicios en twitter

Loja, 28 de Mayo de 2015



Director

Responsable



6. Certificado de culminación y aprobación de la aplicación web desarrollada.



SERVICIO INTEGRADO DE SEGURIDAD
SIS ECU-911 LOJA



Loja 19 de abril del 2017

Ing. Juan Pablo Cabrera Samaniego, Especialista Local del Soporte Tecnológico
SIS ECU911 Loja

CERTIFICO

Que el Sr. Franklin de Jesús Michay Cuenca, con cédula de identidad 1104993157, ha desarrollado su tema te tesis que versa: "DESARROLLO DE UNA APLICACIÓN WEB DE ANÁLISIS ESTADÍSTICO Y MONITOREO DE UNA CUENTA DE TWITTER", para la Unidad Local de Comunicación Social en base a los requisitos planteados por la Unidad Local de Soporte Tecnológico llegando a finalizar el mencionado tema con todos los requerimientos y necesidades que fueron planteadas a lo largo de su desarrollo.

Constancia que se expide a petición de la parte interesada en la Ciudad de Loja a los 19 días del mes de Abril del 2017.



Ing. Juan Pablo Cabrera S. Mg. Sc.
Especialista Local de Soporte Tecnológico
SIS ECU-911-LOJA
(07)3701239

