



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TEMA:

Aplicación para Creación e Información de Alertas, Eventos y
Actividades que se Generen en el Campus Universitario con
Geolocalización aplicado a la Facultad de Ingeniería de la Universidad
Católica de Santiago de Guayaquil

AUTOR:

Medina Guerrero Rodrigo Fernando

**Trabajo de titulación previo a la obtención del grado de
INGENIERO EN SISTEMAS COMPUTACIONALES**

TUTOR:

Ing. Toala Quimí Edison José, Mgs.

Guayaquil, Ecuador

18 de marzo de 2019



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación **Aplicación para Creación e Información de Alertas, Eventos y Actividades que se Generen en el Campus Universitario con Geolocalización aplicado a la Facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil**, fue realizado en su totalidad por **Medina Guerrero Rodrigo Fernando** como requerimiento para la obtención del Título de **Ingeniero en Sistemas Computacionales**.

TUTOR

f. _____


Ing. Toala Quimi Edison José, Mgs.

DIRECTORA (e) DE LA CARRERA

f. _____


Ing. Camacho Coronel Ana Isabel, Mgs.



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

DECLARACIÓN DE RESPONSABILIDAD

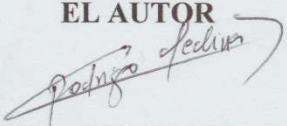
Yo, Medina Guerrero Rodrigo Fernando

DECLARO QUE:

El Trabajo de Titulación **Aplicación para Creación e Información de Alertas, Eventos y Actividades que se Generen en el Campus Universitario con Geolocalización aplicado a la Facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil** previo a la obtención del Título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 18 días del mes de marzo del año 2019

EL AUTOR

f. _____

Medina Guerrero Rodrigo Fernando



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

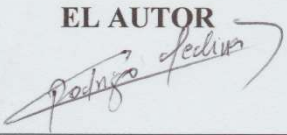
**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

AUTORIZACIÓN

Yo, Medina Guerrero Rodrigo Fernando

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación **Aplicación para Creación e Información de Alertas, Eventos y Actividades que se Generen en el Campus Universitario con Geolocalización aplicado a la Facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 18 días del mes de marzo del año 2019

EL AUTOR

f. _____

Medina Guerrero Rodrigo Fernando



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

REPORTE DE URKUND

URKUND	
Documento	Tesis Rodrigo Fernando Medina Guerrero Final-Formato Senecyt.docx (D48332340)
Presentado	2019-02-25 20:49 (-05:00)
Presentado por	EDISON TOALA QUIMI (edison.toala@cu.ucsg.edu.ec)
Recibido	edison.toala.ucsg@analysis.arkund.com
	2% de estas 58 páginas, se componen de texto presente en 2 fuentes.

DEDICATORIA

Este trabajo de titulación se lo dedico a mi familia: mi mamá Gloria Guerrero, mis hermanas Gabriela y Catherine Medina, mi hermano Cristhian Chávez, al más pequeño Rodrigo Francisco Medina, y en especial a mi padre Patricio Medina, quien creyó en mí y me dio su apoyo incondicional en todo momento.

RODRIGO FERNANDO MEDINA GUERRERO

AGRADECIMIENTO

A mi padre Patricio Medina por todo su esfuerzo y sacrificio para que alcance esta meta, a todos mis maestros de los cuales me siento orgulloso de haber sido su alumno, a mi esposa Geomara Rivera por su apoyo día a día, a mis amigos Adrián, Milton, Roberto y Mario que siempre me motivaron a seguir adelante, a mi tutor Ing. Edison Toala por haberme guiado en todo este proceso y por último a EXPALSA por abrirme las puertas de la empresa y permitirme desarrollarme como profesional y poder cumplir este objetivo.

RODRIGO FERNANDO MEDINA GUERRERO



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

TRIBUNAL DE SUSTENTACIÓN

f.

**Ing. Camacho Coronel Ana Isabel, Mgs.
DIRECTORA (e) DE LA CARRERA**

f.

**Ing. Cornejo Gómez Galo Enrique, Mgs.
COORDINADOR DEL ÁREA**

f.

**Ing. Morejón Campoverde José Lenin, Mgs.
OPONENTE**

ÍNDICE GENERAL

RESUMEN.....	xv
INTRODUCCIÓN	2
CAPÍTULO I EL PROBLEMA	3
1.1 Planteamiento del Problema	3
1.2 Hipótesis de Investigación	4
1.3 Objetivos de la Investigación.....	4
1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.4 Justificación	5
1.5 Alcance	5
CAPÍTULO II	6
MARCO TEÓRICO, CONCEPTUAL Y LEGAL	6
2.1 Teorías y principios relativos a la creación y difusión de eventos relacionados a la academia, con base a herramientas tecnológicas	6
2.1.1 Sistema de información y elementos que lo conforman.....	6
2.1.2 Sistema de información geográfico	8
2.1.3 Computación en la nube.....	8
2.1.3 Aplicación Web	9
2.2 Marco conceptual.....	12
2.2.1 Lenguajes de programación.....	12
2.2.2 Base de datos	13
2.2.3 MVC (Modelo, Vista, Controlador).....	15
2.2.4 Microsoft Azure.....	16
2.2.5 SignalR y SQL Dependency.....	17
2.2.6 SQL Server Service Broker	18
2.3 Sustento legal	19
2.3.1 Propiedad intelectual	19
2.3.2 Derecho de autor	19
2.3.3 Derechos de Propiedad Intelectual.....	20
2.4 Ámbito de aplicación	20
CAPÍTULO III	21

METODOLOGÍA DE LA INVESTIGACIÓN	21
3.1 Metodología cualitativa	21
3.2 Técnicas para recolección de datos.....	22
3.2.1 Entrevista	23
3.3 Análisis de resultados	24
3.4 Metodología de desarrollo para la propuesta técnica.....	25
3.4.1 Modelos para construcción de Software	25
3.4.2 Entorno de desarrollo Microsoft Visual Studio 2017 Community	26
3.4.3 Factores para escoger un proveedor de servicios de geolocalización	27
3.4.4 Google Maps Platform	30
CAPÍTULO IV PROPUESTA TECNOLÓGICA	31
4.1 Introducción	31
4.2 Objetivo.....	31
4.3 Descripción del aplicativo web.....	32
4.3.1 Módulos y componentes de la solución	32
4.3.2 Herramientas tecnológicas	32
4.3.3 Procesos	33
4.3.4 Diseño de la aplicación web	34
4.4 Plan de pruebas	49
4.4.1 Objetivos.....	49
4.4.2 Alcance	49
4.4.2 Resultados de las pruebas	50
4.5 Requerimientos para la implementación.....	50
4.5.1 Requisitos de hardware.....	50
4.5.2 Servicios en la nube.....	50
4.6 Resultados esperados del aplicativo.....	54
CONCLUSIONES	55
RECOMENDACIONES	56
BIBLIOGRAFÍA	57
ANEXOS	60
ANEXO A: SCRIPT DE CREACIÓN DE MODELO E-R DE LA BASE DE	
DATOS	60
Tablas	60

Triggers	66
Funciones.....	68
Procedimientos Almacenados	73
ANEXO B: MATRICES DE PRUEBA.....	87
Registro de usuario en el sistema	87
Creación de eventos con geolocalización.....	88
Visualización de Estadísticas de Asistencia	89
ANEXO C: RESULTADOS DE PRUEBAS.....	90
ANEXO D: MANUAL DE USUARIO	95
Acceso al portal web	95
Ingreso al sistema	95
Creación de usuario	95
Inicio de sesión.....	96
Olvido de contraseña	96
Dashboard.....	97
Menú de Opciones.....	98
Barra Superior	98
Creación de Eventos	99
Estadísticas de Asistentes	100
Parámetros	102
Administración de Usuarios	103
Roles de Usuario	104
Permisos por Rol	104
ANEXO E: MANUAL TÉCNICO	105

ÍNDICE DE TABLAS

Tabla 1. Descripción de los actores. Fuente: el autor	35
Tabla 2. Descripción de los casos de uso. Fuente: el autor.....	36
Tabla 3. Tabla Eventos. Fuente: el autor	39
Tabla 4. Tabla Entidad. Fuente: el autor	39
Tabla 5. Tabla EventoEntidad. Fuente: el autor.....	40
Tabla 6. Tabla TipoEvento. Fuente: el autor.....	40
Tabla 7. Tabla EventoTipoDetalle. Fuente: el autor	40
Tabla 8. Tabla Comentario. Fuente: el autor.....	40
Tabla 9. Tabla AspNetRoles. Fuente: el autor	41
Tabla 10. Tabla AspNetUserRoles. Fuente: el autor.....	41
Tabla 11. Tabla AspNetUsers. Fuente: el autor	41
Tabla 12. Tabla Auditoria. Fuente: el autor	42
Tabla 13. Tabla Parametro. Fuente: el autor	42
Tabla 14. Tabla Archivo. Fuente: el autor	42
Tabla 15. Tabla EventoAlertaUsuario. Fuente: el autor	43
Tabla 16. Tabla EventoInvitado. Fuente: el autor.....	43
Tabla 17. Prueba funcional de aplicación. Fuente: el autor	49
Tabla 18. Matriz de Prueba de Registro de Usuario en el Sistema. Fuente: el autor	87
Tabla 19. Matriz de Prueba de Creación de Evento con Geolocalización. Fuente: el autor.....	89
Tabla 20. Matriz de Prueba de Notificaciones y Alertas de Eventos con Geolocalización. Fuente: el autor.....	89
Tabla 21. Matriz de Prueba de Visualización de Estadísticas. Fuente: el autor.....	89

ÍNDICE DE FIGURAS

Figura 1. Taxonomía de aplicaciones web. Nota: Tomado de Manager (2016).....	9
Figura 2. Arquitectura para aplicaciones Web estáticas Nota: Tomado de Manager (2016)	10
Figura 3. Arquitectura de conexión e un sitio web. Nota: Tomado de Orosco (2014)	11
Figura 4. Modelo Vista Controlador. Tomado de Hernández (2015).....	16
Figura 5. Tomado de Modelo en Cascada por Gutierrez (2011).....	26
Figura 6. Arquitectura de la Aplicación. Fuente: el autor.....	33
Figura 7. Casos de uso de la aplicación web. Fuente: el autor.....	34
Figura 8. Tablas relacionadas al core de la aplicación. Fuente: el autor.....	37
Figura 9. Tablas aisladas. Fuente: el autor.....	37
Figura 10. Tablas relacionadas a la autenticación de la aplicación. Fuente: el autor	38
Figura 11. Tablas relacionadas a la notificación de eventos. Fuente: el autor.....	38
Figura 12. Vista de inicio de sesión. Fuente: el autor	44
Figura 13. Vista de registro de usuario. Fuente: el autor	44
Figura 14. Vista de recuperación de contraseña. Fuente: el autor	45
Figura 15. Vista de creación de evento. Fuente: el autor	46
Figura 16. Evento creado satisfactoriamente. Fuente: el autor	46
Figura 17. Vista de la bandeja principal de la aplicación desde un dispositivo móvil. Fuente: el autor	47
Figura 18. Vista de la bandeja principal de la aplicación desde un PC. Fuente: el autor.....	47
Figura 19. Sección para interactuar con los eventos. Fuente: el autor.....	47
Figura 20. Pantalla de Estadísticas gráfico tipo Lineal. Fuente: el autor.....	48
Figura 21. Pantalla de Estadísticas gráfico tipo Dona. Fuente: el autor	48
Figura 22. Pantalla de Administración de Usuarios. Fuente: el autor.....	49
Figura 23. Planes de servicios de SmarterASP.NET. Fuente: SmarterASP.NET.....	51
Figura 24. Planes de servicios de aplicaciones de Microsoft Azure. Fuente: Microsoft Azure	51
Figura 25. Planes de servicios de Google Maps Platform. Fuente: Google.....	52
Figura 26. Detalle de factura generada por uso de API de Google. Fuente: el autor.	52

Figura 27. Detalle de plan de envío de correos por SendGrid. Fuente: Microsoft Azure	53
Figura 28. Planes de servicios de Azure Schedule. Fuente: Azure	53
Figura 29. Ingreso de formulario incompleto. Fuente: El Autor.....	90
Figura 30. Ingreso de formulario con usuario ya existente. Fuente: El Autor.....	90
Figura 31. Ingreso de formulario con correo ya existente. Fuente: El Autor.....	90
Figura 32. Error al cargar pantalla para creación de evento sin validar geolocalización. Fuente: El Autor	91
Figura 33. Prueba de creación de evento: validaciones sobre campos obligatorios. Fuente: El Autor	91
Figura 34. Prueba de creación de evento: validación sobre fecha de inicio. Fuente: El Autor.....	91
Figura 35. Prueba de creación de evento: validación sobre fecha de fin. Fuente: El Autor.....	92
Figura 36. Prueba de creación de evento generación exitosa. Fuente: El Autor.....	92
Figura 37. Notificación sobre creación de evento. Fuente: El Autor.....	93
Figura 38. Notificación sobre alerta de evento. Fuente: El Autor	93
Figura 39. Prueba de Estadísticas gráfico tipo Lineal. Fuente: el autor.....	93
Figura 40. Prueba de Estadísticas gráfico tipo Dona. Fuente: el autor	94
Figura 41. Pantalla para creación de usuario. Fuente: El Autor.....	96
Figura 42. Notificación recibida con datos de acceso al sistema. Fuente: El Autor ..	96
Figura 43. Pantalla para restaurar contraseña. Fuente: El Autor.....	96
Figura 44. Dashboard. Fuente: El Autor	97
Figura 45. Menú de Opciones. Fuente: El Autor	98
Figura 46. Barra Superior. Fuente: El Autor.....	99
Figura 47. Creación de Eventos. Fuente: El Autor	100
Figura 48. Estadística de Asistentes. Fuente: El Autor.....	101
Figura 49. Parámetros. Fuente: El Autor	102
Figura 50. Administración de Usuarios. Fuente: El Autor.....	103
Figura 51. Roles. Fuente: El Autor	104
Figura 52. Permisos por Rol. Fuente: El Autor.....	104

RESUMEN

Este trabajo de titulación está basado en el desarrollo e implementación de una aplicación web para la creación de eventos que se desarrollen en el campus universitario con geolocalización aplicado a la facultad de ingeniería y la generación de alertas para que los usuarios estén pendientes sobre el desarrollo de las actividades que se registren en el sistema.

Luego de analizar los diversos proveedores de geolocalización se decidió utilizar el api del Google ya que posee alta disponibilidad del servicio, abundante documentación, y un gran beneficio en cuanto al costo para desarrolladores ya que da un crédito de 200USD mensuales durante 1 año. Por otra parte, se optó por una aplicación web por la compatibilidad para ejecutarse sobre cualquier dispositivo que tenga un navegador web y una conexión a internet.

Este proyecto tuvo un enfoque de investigación cualitativa, se realizó levantamiento de información observando el procedimiento para la difusión de eventos en la facultad y por medio de entrevistas al personal administrativo se identificaron las necesidades y los canales de comunicación actual.

Se realizó el diseño, desarrollo e implementación del sistema web de difusión de eventos el cual se encuentra publicado totalmente en la nube, es decir, se puede acceder desde cualquier lugar y a cualquier hora, lo cual brinda disponibilidad de la aplicación en todo momento para la constante comunicación entre la comunidad universitaria.

Palabras Clave: APLICACIÓN WEB; GEOLOCALIZACIÓN; GOOGLE API; MVC; AZURE; SENDGRID; BLOB STORAGE; C#

INTRODUCCIÓN

En la actualidad en las comunidades académicas o universitarias, converge todo tipo de información, y es precisamente en estos ámbitos donde se promueven eventos de distinta índole: culturales, charlas, conferencias, congresos, sin dejar a un lado los sociales y deportivos, por esta razón se hace necesario contar con herramientas digitales que permitan el poder comunicar a los usuarios, sobre la variedad de estos temas que se presentan día a día.

Es aquí donde se hace importante el uso de la tecnología, para con el apoyo de herramientas informáticas dotadas en todo momento de la robustez necesaria, se pueda brindar aplicaciones que cumplan los estándares en portabilidad y escalamiento; garantizando en forma permanente la tranquilidad del usuario que los datos que ingrese, siempre serán reales y oportunos. Con este proyecto investigativo de orden técnico, se logrará mostrar en cada fase de la investigación su viabilidad y grado de madurez, según la postura del investigador; donde se comenzará la propuesta, con el orden científico establecido.

Capítulo I, en él se abordará claramente el problema de investigación, se presentará el objetivo general, específicos, antecedentes, variables que puedan enmarcar el horizonte del proceso de investigación; Capítulo II, se inicia la conceptualización del tema abordando el marco de antecedentes, histórico, conceptual y legal; Capítulo III, este capítulo se explicará la metodología que se utilizó para llegar al objetivo de la ciencia, generar conocimiento; se analizarán los datos obtenidos de la técnica de investigación; para finalizar con el Capítulo V, donde se realiza la propuesta técnica del desarrollo de la aplicación en ambiente web, teniendo entre sus bondades ser una herramienta multi plataforma.

Se estima oportuno que este tipo de aplicación será de mucha ayuda para los estudiantes de la Universidad Católica Santiago de Guayaquil, brindando una mejora en la comunicación interna del recinto; así mismo se recomienda a las autoridades brindar todo el apoyo posible a estos proyectos de emprendimiento, que motivan el desarrollo socio económico a favor de los estudiantes, al fortalecer el conocimiento obtenido a lo largo de la carrera de Ingeniería en Sistemas Computacionales.

CAPÍTULO I

EL PROBLEMA

La comunicación como herramienta de interconexión en la sociedad, permite llegar y conectar a personas, que involucren algún grado de afinidad por las actividades propuestas en la sociedad, donde el poder compartir se hace justamente muy necesario al momento del desarrollo de afinidades.

1.1 Planteamiento del Problema

Son diversas las actividades que se desarrollan en la Universidad Católica de Santiago de Guayaquil (UCSG), las cuales se ven publicadas en redes sociales como Facebook o Instagram e inclusive en diarios con impacto a nivel nacional. También se pueden apreciar volantes electrónicos en el portal de servicios en línea, las cuales informan sobre actividades y festivales culturales, avisos de pago de pensiones, matriculas, exámenes, entre otros.

Existen además estudiantes que realizan actividades en sus respectivas facultades con el objetivo de recaudar fondos con fines de ayuda social, eventos académicos, culturales, deportivos, entre otros como por ejemplo la recaudación de ropa, juguetes, para realizar la entrega a quien lo necesite.

A pesar de los esfuerzos que realiza la UCSG por mantener informada a la comunidad universitaria es muy probable que a muchos no les llegue el mensaje, sea por que no usan redes sociales, no leen periódicos, no usan el portal de servicios en línea o que simplemente les pase por desapercibido.

Por otra parte, los estudiantes no tienen las herramientas necesarias para poder crear eventos que tengan la facilidad de llegar a demás alumnos sean de otras facultades o de la misma facultad. Por lo expuesto es necesaria una plataforma que permita la creación de eventos y que informe sobre los mismos de forma muy específica y de manera fácil y oportuna.

1.2 Hipótesis de Investigación

Diseñar una herramienta tecnológica de difusión informativa, para uso de la comunidad universitaria, permitirá mantener notificados sobre los eventos y actividades institucionales a la comunidad universitaria. Las variables que se derivan de esta hipótesis son:

Independiente: Herramienta tecnológica de difusión

Dependiente: Mejoramiento de la comunicación desde y hacia la universidad

1.3 Objetivos de la Investigación

En base a la propuesta de investigación, se motivan los siguientes objetivos bajo la perspectiva del investigador:

1.3.1 Objetivo General

Desarrollar e implementar un sistema basado en geolocalización, que permita alertar, notificar eventos que se generen en el campus universitario.

1.3.2 Objetivos Específicos

- Identificar cuáles son los medios de difusión institucionales, las actividades y eventos específicos que se desarrollan en la UCSG
- Analizar técnicamente las herramientas disponibles en el mercado para geolocalización, que permitan desarrollar un sistema más amplio y flexible de comunicación interna
- Desarrollar una aplicación web bajo ambiente hosting en la nube, para la creación y difusión de actividades y eventos universitarios.
- Implementar y verificar el funcionamiento de la aplicación web para creación de eventos con geolocalización, a través de pruebas en sitio.

1.4 Justificación

El desarrollo de la aplicación para registrar eventos que se generen en la universidad, que a la vez sea posible compartirlos, es esencial ya que permitirá ser portavoces proactivos de necesidades que se presenten en el día a día, ayudará a promover eventos sociales, culturales, deportivos, entre otros y creará un sistema de comunicación más efectivo, fácil y rápido.

Los principales beneficiarios de este proyecto serán todos los que conforman la comunidad universitaria: profesores, estudiantes, autoridades, etc. ya que podrán mantenerse informados sobre todas las actividades que se registren.

Este trabajo de titulación está enmarcado en las líneas de investigación de la UCSG, y de la carrera Ingeniería en Sistemas Computacionales: Artes, Letras y Conectividad Tecnológica.

1.5 Alcance

El servicio para uso de la aplicación se basa en los siguientes parámetros:

- Aplicación web para la UCSG para el semestre B-2018.
- Módulo de seguridad: creación de usuarios, roles, permisos, manejo de sesiones, recuperación de contraseña.
- Creación de eventos en ubicación del mapa clasificado de acuerdo a las necesidades que se identifiquen en el levantamiento de información.
- Los eventos permitirán adjuntar imágenes y observaciones.
- Los eventos tendrán fecha de inicio y fecha de fin.
- Notificaciones en tiempo real sobre comentarios y asistencia a los eventos creados por el administrador.

CAPÍTULO II

MARCO TEÓRICO, CONCEPTUAL Y LEGAL

En el presente capítulo se muestran todos los aspectos teóricos relacionados con la construcción de una aplicación informativa basada en geolocalización, enfocando el desarrollo a una investigación que incluya la mejor práctica en metodologías de desarrollo ágil, relacionada a los frameworks que trabajan bajo el patrón de diseño MVC (Model, View, Controller), lenguajes de programación orientado a objetos, conjuntamente con la utilización de varias herramientas web, las mismas que van a satisfacer las necesidades de navegabilidad del sitio, estimando el tiempo, esfuerzo para la implementación con varios elementos de configuración del software, de esta manera se podrá guiar el tema de estudio para su correcta implementación. Así como también se realiza una breve argumentación desde el punto de vista legal que soporta la implementación del presente proyecto.

2.1 Teorías y principios relativos a la creación y difusión de eventos relacionados a la academia, con base a herramientas tecnológicas

Dada la importancia de la difusión de ciertos eventos que sustentan la actividad académica de una universidad, y considerando que el propósito de dichos eventos es mantener actualizada la relación de estudiantes y profesores con la producción académica y científica, es necesario revisar algunos elementos teóricos que sustentan y mantienen activa la interacción de estudiantes y profesores con la vida académica, utilizando el avance acelerado de la tecnología y su influencia en la vida de las personas.

2.1.1 Sistema de información y elementos que lo conforman

Un sistema se puede catalogar como un todo integrado, (ElectronicTerms, 2008) donde convergen las funciones necesarias para un correcto desempeño, en los sistemas de transmisión de datos la parte medular está compuesta por la información, que en si es lo que se trasmite entre los usuarios del sistema.

La información se puede considerar como “un conjunto de datos acerca de algún suceso, hecho, fenómeno o situación, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo” (Castellanos, 2008, p.10), siendo de vital importancia el correcto uso que se le dé a la misma.

Entre los elementos de un sistema de información se puede mencionar a las personas, que son las encargadas de la utilización de datos; estos datos a su vez se pueden encontrar dispersos en cualquier sistema, actividades o técnicas de trabajo; es la suma de planeaciones para levantar de manera correcta un sistema y los recursos materiales en general son todos los equipos que van permitir el traspaso de la comunicación entre los diferentes elementos que componen el sistema. De otro lado, todos estos elementos interactúan entre sí para procesar los datos incluyendo la gesta manual o automática, dando lugar a información más elaborada y distribuyéndola de la manera más adecuada posible, en una determinada organización en función de sus objetivos.

Un sistema enfocado en información para datos de manera concreta en su estructura se maneja bajo cuatro componentes básicos en su pertinencia, los cuales son:

- Entrada de datos: Generalmente es propiciada por el usuario del sistema, donde las entradas serian manuales; y las automáticas que se pueden generar por configuraciones en el sistema, para la propia auto captura.
- Almacenamiento de datos: En la actualidad con el procesamiento de grandes tamaños de información, que demandan mucho espacio en disco físico, se presentan nuevas tendencias como el almacenamiento en la nube.
- Procesamiento de Información: El uso de la parte física del equipo informático para el proceso pleno de la información es muy importante, debido a que esto va a permitir poder procesar los datos que se ingresa de manera manual o automática; dando en lo posterior el atributo de usar de manera acertada toda esta información que se ha obtenido, de varias fuentes por parte del sistema.
- Salida de Información: En esta parte del sistema de información se obtiene una data procesada, que por sí ya sirve como un medio de conocimiento por

los factores que previamente fueron recibidos en la ingesta de datos. Cabe mencionar que en este paso las organizaciones pueden tomar decisiones más reales en base a la información obtenida.

2.1.2 Sistema de información geográfico

Si el objeto concreto de un sistema de información es la obtención de datos relacionados con el espacio físico, entonces se estará hablando de un Sistema de Información Geográfica o SIG, así pues, un SIG es un software específico que permite a los usuarios crear consultas interactivas, integrar, analizar y representar de una forma eficiente cualquier tipo de información geográfica referenciada asociada a un territorio, conectando mapas con bases de datos (SIG, 2017).

Este tipo de información geográfica permite tener una referencia situacional, del lugar donde físicamente se está emitiendo, transmitiendo algún tipo de información referencial a la ubicación propia del sitio.

2.1.3 Computación en la nube

Para brindar una aplicación que esté disponible dónde y cuándo sea, es necesario el uso de los servicios de computación en la nube que se ofrecen en la actualidad, tales como base de datos, servicio de aplicaciones, envío de correos, almacenamiento, etc.

Según Arias (2015) en su síntesis indica que:

“La computación en la nube se define como un sistema de computación distribuido orientado al consumidor, que consiste en una colección de ordenadores virtualizados e interconectados que son suministrados dinámicamente y presentados como uno o más recursos computacionales unificados, conforme acuerdo de nivel de servicio negociado entre el proveedor de servicios y el consumidor. Entre sus principales características está el reparto de los recursos que sirven a múltiples usuarios y que son suministrados dinámicamente según la capacidad contratada por cada uno. La entrega de los servicios por la red se lleva a cabo de forma de web servicios implementados mediante la arquitectura orientada a servicios. Esto proporciona una flexibilidad caracterizada por la

posibilidad de añadir o eliminar recursos de forma rápida y sin un gran esfuerzo administrativo por parte del proveedor”. (p. 9)

2.1.3 Aplicación Web

Las aplicaciones con tecnología basada en web son todas aquellas aplicaciones que se accede por medio de una sola interfaz, que es el navegador, lo cual resulta muy práctico ya que el navegador funciona de manera independiente al sistema operativo, es decir, puedo tener una PC con Windows, una Tablet con Android o un Smartphone con iOS, y abrir el navegador de mi preferencia para utilizar la aplicación.

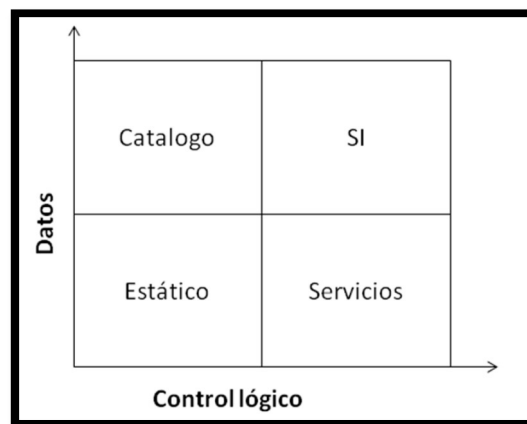


Figura 1. Taxonomía de aplicaciones web. Nota: Tomado de Manager (2016)

Las aplicaciones web han ganado terreno sobre las aplicaciones tradicionales (de escritorio, locales, etc.) por las siguientes razones:

- Las aplicaciones web son más accesibles: gracias al uso de protocolos estándar como HTTP toda aplicación web puede ser usada sin importar ubicación y plataforma.
- Las aplicaciones web son más fáciles de mantener: desde el momento en que una aplicación se despliega en el servidor, los clientes pueden acceder a ella, sin necesidad de una aplicación cliente. Esto permite que cualquier cambio solo se haga en el servidor y no en el cliente, ahorrando tiempos de desarrollo.

Una aplicación web sigue un **modelo cliente-servidor**. Esto se debe a que se requiere un componente que hospede a la aplicación y a todos sus componentes servidor Web y un componente cliente, para este caso de investigación un navegador

web. Una vez que se tiene estos dos componentes se necesita un sistema de archivos en donde se van a almacenar todos los recursos para la generación de contenido. El sistema de archivos puede almacenar una gran cantidad de archivos multimedia como fotos, videos, audio, etc., archivos XML y HTML, necesarios para el funcionamiento de la aplicación.

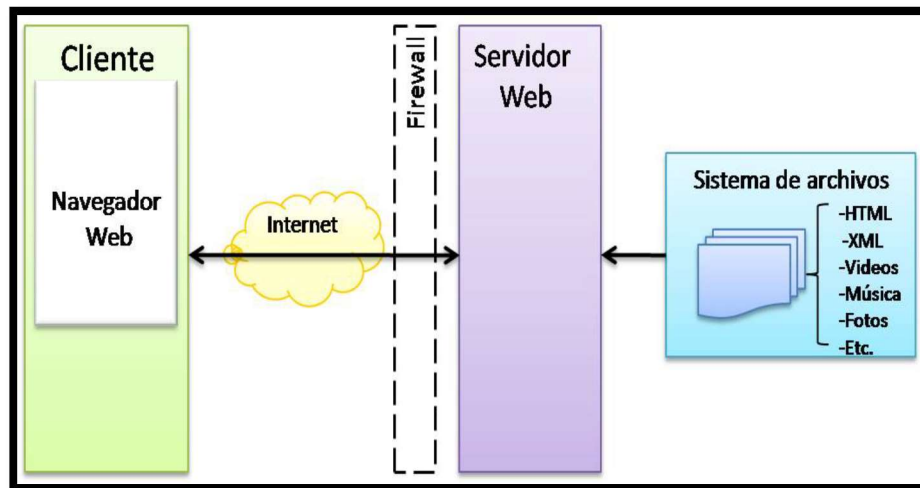


Figura 2. Arquitectura para aplicaciones Web estáticas Nota: Tomado de Manager (2016)

Los **sistemas basados en técnicas web** son una modalidad actual de mayor aplicación a partir de la nueva era de la tecnología web, ciertamente esta emplea un tipo de información donde se concentran, dentro de los servidores, todas las tareas y almacenamiento de datos brindando servicios a todos los clientes, pueden ser implementados en redes de área local LAN, redes de áreas extensas WAN o a través de Internet; siendo rápidos y dinámicos para el usuario. Actualmente el internet es una tecnología muy necesaria, útil para el desarrollo global, es la fuente de información de todas las empresas por lo que está disponible en todo momento existen varios tipos de sistemas de información web, entre ellos se tienen a:

- De información
- Orientados a transacciones, servicios
- Acceso a una base de datos
- Como almacén de datos
- Redes Sociales

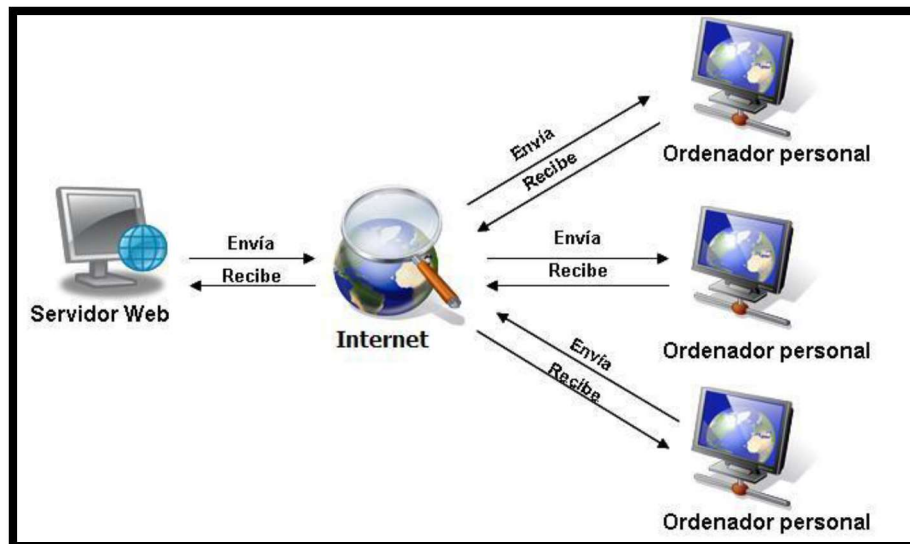


Figura 3. Arquitectura de conexión e un sitio web. Nota: Tomado de Orosco (2014)

El desarrollo de una aplicación web conlleva al uso de métodos sistemáticos para que su funcionamiento sea de forma eficiente, a esos métodos se los denomina **ingeniería web**. Esto posibilita estructuras de datos enriquecidas que se utilizan para:

- Facilitar la búsqueda de contenidos y servicios.
- Crear agentes inteligentes para la automatización de procesos.
- Extraer, procesar y almacenar conocimiento
- Integrar sistemas heterogéneos.
- Componer sistemas complejos.
- Filtrar información

La implementación de ingeniería en la web es fundamental en la actualidad, Según Soliz (2009) en su síntesis indica que:

“La ingeniería web es un proceso utilizado para crear, implantar, mantener aplicaciones y sistemas de alta calidad. La ingeniería Web está relacionada con el establecimiento y utilización de principios científicos de ingeniería y gestión, con enfoques sistemáticos, disciplinados del éxito y desarrollo, empleo, mantenimiento de sistemas y aplicaciones Web de alta calidad. Las aplicaciones web se integran cada vez más en las estrategias de negocios de las organizaciones grandes y pequeñas, es necesario que estas aplicaciones sean prácticas, confiables y adaptables”.

2.2 Marco conceptual

2.2.1 Lenguajes de programación.

Existe una gran cantidad de lenguajes de programación, muchos de ellos tienen características específicas según el uso o la afinidad al momento de desarrollar; a continuación, se citan los lenguajes que se utilizarán en el desarrollo del proyecto, se considera importante conocer de cada uno de estos lenguajes, que son, principalmente dos: lenguaje C# y lenguaje Java Script.

En cuanto al **lenguaje C#** Wagner (2015) sostiene que:

“Utiliza una sintaxis muy expresiva, pero también sencilla y fácil de aprender. Cualquier persona familiarizada con C o C++, reconocerá al instante la sintaxis de llaves de C#. Los desarrolladores que conocen cualquiera de estos lenguajes puede empezar normalmente a trabajar en C# de forma productiva en un espacio muy corto de tiempo. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces, como tipos de valor que aceptan valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java”.

Su ambiente de desarrollo orientado a objetos, ayuda a la fácil implementación de módulos, que permiten generalizar un entorno, bajo la forma de un sistema innovador que puede incluir las siguientes funciones:

- Parámetros de encapsulado, permitiendo informes, presentando seguridad en procesamiento de la información.
- Descripción de los parámetros sentenciados en las variables de registros.
- Presentación de los atributos, que forman parte de la data en los procesos de traspaso en información.

Si se tiene que interactuar con otro software de Windows se lo puede hacer usando mediante un proceso denominado Interoperabilidad. La interoperabilidad permite que los programas de C# hagan casi todo lo que puede hacer una aplicación

C++ nativa. C# permite el uso de punteros y el concepto de código no seguro en los casos en los que el acceso directo a memoria es absolutamente crítico. (Wagner, 2015)

Para el desarrollo de aplicaciones web es importante contar con un lenguaje de programación que sea compatible con cualquier navegador moderno, interactivo con HTML y mejor aún open source. Todo eso es posible con **JavaScript**, un lenguaje orientado a objetos que permite mejorar la gestión cliente servidor, dándole efectos dinámicos a la página. Este tipo de lenguaje es de manera primordial utilizado del lado del usuario, ósea del cliente, dicho de otra manera, se ejecuta en el ordenador local, mas no en un servidor, logrando ser una potente herramienta para diseñar o crear efectos, formas dinámicas en los portales web.

Entre sus principales características están:

- Lenguaje del lado del usuario (cliente).
- Compatible con la mayoría de navegadores
- Permite controlar las ventanas del navegador y el contenido que muestran.
- Se puede evitar depender del servidor Web para cálculos sencillos.
- Captura los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Obtiene los datos que el usuario introduce en un formulario antes de enviarlos.

2.2.2 Base de datos

Es el repositorio donde va estar alojada la información que se gestiona en la empresa. En su texto Tsai (2007) expresa que Base de datos es: “una colección de archivos interrelacionados y creados con un sistema de gestión de base de datos. El contenido de una base se obtiene combinando datos de todas las diferentes fuentes en una organización, de tal manera que los datos estén disponibles para todos los usuarios” (p. 5).

De manera tradicional las bases de datos son de origen transaccional como **MySQL**, encontrando también las no transaccionales, a medida de ejemplo podría ser MongoDB, Spark DB, que son las que actualmente han crecido en demanda por su pronta adaptación a la masificación de los datos.

Según Net (2012) sostiene que MySQL es:

“Un sistema de gestión de bases de datos relacional, multi hilo y multiusuario con más de seis millones de instalaciones. MySQL desde enero de 2008 pasa a una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation, la cual desde abril de 2009 desarrolla MySQL como software libre en un esquema de licenciamiento dual.”

Según el criterio de varios autores las características disponibles en las últimas versiones se pueden destacar:

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas. Conectividad segura.

MySQL es una base de datos de uso relacional, que brinda la gestión del uso de archivos dentro del sistema, permitiendo su oportuna gestión. Al ser relacional se basa su funcionamiento en el uso de tablas para su organización y almacenamiento. El código fuente de este lenguaje está desarrollado en C, C++, una de las partes más atractivas de este programa es su fácil modo de acoplamiento con otros entornos de desarrollo, y las diferentes distribuciones de la parte operativa en los sistemas.

Presenta muchas ventajas:

- La base de datos MySQL es Open Source.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

Otra base de datos transaccional es **Microsoft SQL Server** que se trata de un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact - SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas. (Rouse, 2017)

Presenta una serie de características, que motivan su uso por parte de los desarrolladores, entre ellas se tienen:

- Brinda un amplio soporte en la transmisión de transacciones.
- Presenta un alto grado en seguridad, permite escalabilidad.
- Su entorno grafico se presenta intuitivo, amigable para el desarrollador.
- La forma colaborativa de trabajo tiene su función cliente a servidor, o viceversa; logrando procesar información en los host terminales, para congregar la base de datos en un equipo servidor.
- Trabaja en modo de redundancia, permitiendo obtener información de varios servidores.

2.2.3 MVC (Modelo, Vista, Controlador)

MVC o Modelo-Vista-Controlador es un patrón de arquitectura de software que utiliza tres componentes fundamentales: vistas, modelos y controladores. Este diseño separa la lógica de la aplicación de la lógica de la vista en una aplicación. Su arquitectura es muy relevante ya que se lo puede utilizar desde desarrollos muy básico y sencillos hasta soluciones empresariales de gran importancia; Para su arquitectura los frameworks modernos en su mayoría utilizan el Modelo-Vista-Controlador, o alguna de sus variantes, entre los cuales se pueden AngularJS, Django, entre otros. (Hernández, 2015)

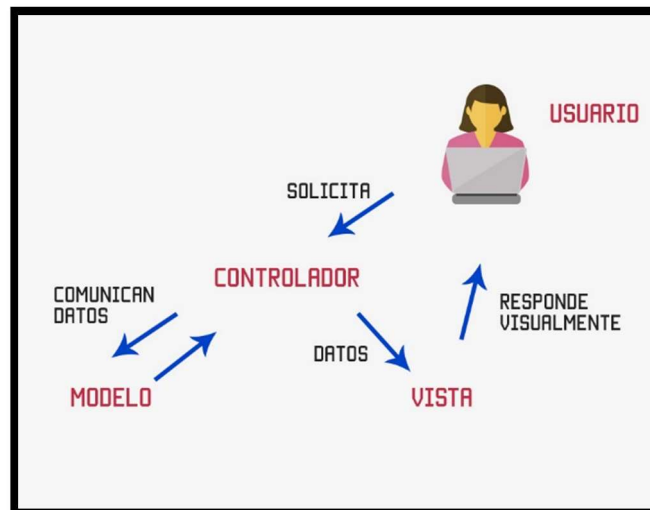


Figura 4. Modelo Vista Controlador. Tomado de Hernández (2015)

2.2.4 Microsoft Azure

Entre los proveedores de servicios de computación en la nube existentes tenemos a Microsoft Azure, el cual tiene una amplia bandeja de opciones, que van desde una base datos hasta un servidor virtualizado con todas las necesidades que pueda tener una organización de primer orden. Además de esto Azure da muchas facilidades a los desarrolladores para que prueben sus servicios y vean sus principales ventajas, por ejemplo, ofrece un mes gratis de uso en cualquiera de sus herramientas, permite la publicación gratuita de aplicaciones en ambiente de desarrollo, y da la facilidad para ajustarse a un presupuesto en base a la necesidad.

La escritora Rouse (2017) indica lo siguiente sobre Microsoft Azure:

“Microsoft Azure, antes conocido como Windows Azure, es la plataforma de computación en nube pública de Microsoft. Proporciona una gama de servicios en la nube, incluidos los de computación, analítica, almacenamiento y redes. Los usuarios pueden elegir entre estos servicios para desarrollar y escalar nuevas aplicaciones, o ejecutar aplicaciones existentes, en la nube pública. Microsoft Azure se considera ampliamente como un servicio de plataforma como servicio (PaaS) e infraestructura como servicio (IaaS)”.

Microsoft clasifica los servicios de Azure en 11 tipos principales de productos:

- **Cómputo:** Estos servicios proporcionan máquinas virtuales, contenedores, procesamiento por lotes y acceso a aplicaciones remotas.
- **Web y móvil:** Estos servicios soportan el desarrollo y despliegue de aplicaciones web y móviles, y también ofrecen funciones para la administración, notificación y generación de informes de API.
- **Almacenamiento de datos:** Esta categoría incluye las ofertas de base de datos como servicio para SQL y NoSQL, así como almacenamiento en la nube no estructurado y en caché.
- **Analítica:** Estos servicios proporcionan análisis y almacenamiento distribuidos, así como analítica en tiempo real, análisis de big data, lagos de datos, aprendizaje automático y data warehousing.
- **Redes:** Este grupo incluye redes virtuales, conexiones y pasarelas dedicadas, así como servicios para la administración del tráfico, el equilibrio de carga y el alojamiento del sistema de nombres de dominio (DNS).
- **Internet de las cosas (IoT):** Estos servicios ayudan a los usuarios a capturar, monitorear y analizar los datos de IoT, de sensores y otros dispositivos.
- **Desarrollo:** Estos servicios ayudan a los desarrolladores de aplicaciones a compartir código, probar aplicaciones y rastrear posibles problemas. Azure soporta una variedad de lenguajes de programación de aplicaciones, incluyendo JavaScript, Python, .NET y Node.js.
- **Gestión y seguridad:** Estos productos ayudan a los administradores de nube a gestionar su implementación de Azure, a programar y ejecutar trabajos, y a crear automatización. Este grupo de productos también incluye capacidades para identificar y responder a amenazas de seguridad en la nube.

2.2.5 SignalR y SQL Dependency

En la actualidad es muy importante tener información en tiempo real, esto ayuda a mantenernos informados de todo tipo de acontecimiento, por tal motivo se generaron las muy conocidas notificaciones, las podemos ver a diario en nuestros teléfonos celulares cuando recibimos un mensaje de WhatsApp o si alguien nos

comenta una foto en cualquiera de las redes sociales que usemos. Existen dos herramientas que hacen que esto sea posible: SignalR y SQL Dependency.

SignalR es una librería para desarrolladores de ASP.NET que agrega funcionalidades web en tiempo real, es decir la capacidad de hacer que el código del lado del servidor envíe contenido a los clientes conectados a medida que se produce en tiempo real mediante el uso de WebSockets, el cual habilita la comunicación entre el servidor y el navegador de forma bidireccional. (Microsoft, 2016)

SQL Dependency representa una dependencia de notificación de consulta entre una aplicación y una instancia de SQL Server. Normalmente una aplicación realiza una petición (consulta) al servidor y este devuelve una respuesta, con SQL Dependency el servidor envía una notificación automática sobre cualquier cambio que detecte en la base de datos. (Microsoft, 2017)

2.2.6 SQL Server Service Broker

El Service Broker es una característica de la base de datos SQL Server la cual interactúa con las librerías de SQL Dependency en la aplicación y permite obtener cualquier cambio que se realice en la base de datos en tiempo real.

Microsoft (2018) en su portal web dice que “SQL Server Service Broker proporciona soporte nativo para aplicaciones de mensajería y puesta en cola en el Motor de base de datos de SQL Server y facilita a los desarrolladores la creación de aplicaciones sofisticadas que utilizan los componentes del Motor de base de datos para comunicarse entre diferentes bases de datos. Los desarrolladores pueden usar Service Broker para crear fácilmente aplicaciones distribuidas y confiables”.

2.3 Sustento legal

2.3.1 Propiedad intelectual

El IEPI (2018) en su portal web sobre propiedad intelectual indica que:

“Se refiere a las creaciones de la mente, tales como obras literarias, artísticas, invenciones científicas e industriales, así como los símbolos, nombres e imágenes utilizadas en el comercio. La Propiedad Intelectual otorga al autor, creador e inventor el derecho de ser reconocido como titular de su creación o invento y, por consiguiente, ser beneficiario del mismo. Los Estados son los responsables de garantizar una legislación clara para precautelar este bien común. En Ecuador, el Servicio Nacional de Derechos Intelectuales (SENADI) es el organismo encargado de proteger, fomentar, divulgar y conducir el buen uso de la Propiedad Intelectual desde el enfoque de tres áreas distintas: la Propiedad Industrial, Derecho de Autor y las Obtenciones Vegetales”.

Se puede indicar que el grado de protección que tiene las personas sobre sus diseños, circuitos integrados, lemas, invenciones se lo puede conocer como propiedad industrial, y esto a nivel nacional se encuentra regulado por el IEPI.

2.3.2 Derecho de autor

Sobre derecho de autor el IEPI (2018) estipula que:

“Se encarga de proteger los derechos de los creadores sobre las obras, sean estas literarias o artísticas, esto incluye: libros, textos de investigación, software, folletos, discursos, conferencias, composiciones musicales, coreografías, obras de teatro, obras audiovisuales, esculturas, dibujos, grabados, litografías, historietas, comics, planos, maquetas, mapas, fotografías, videojuegos y mucho más”.

La protección por derechos de autor basa su importancia en proteger al talento humano sobre la creación de libros, ensayos, artículos, programas, entre otros; se hace imperioso registrar el producto del desarrollo de un sistema, para blindar los derechos de autor.

2.3.3 Derechos de Propiedad Intelectual

Según indica la (Asamblea Nacional, 2008):

“Los derechos de propiedad intelectual se encuentran amparados en la constitución del Ecuador, según el Art. 322.- Se reconoce la propiedad intelectual de acuerdo con las condiciones que señale la ley. Se prohíbe toda forma de apropiación de conocimientos colectivos, en el ámbito de las ciencias, tecnologías y saberes ancestrales. Se prohíbe también la apropiación sobre los recursos genéticos que contienen la diversidad biológica y la agro-biodiversidad”.

Es muy importante que las personas en general, conozcan que en la constitución se ampara el derecho de la propiedad intelectual, lo que es muy beneficioso para los desarrolladores, logrando motivar la producción a este nivel, sabiendo que se encuentran legalmente amparados.

2.4 Ámbito de aplicación

El 17 de mayo de 1962 fue creada la Universidad Católica de Santiago de Guayaquil, institución de nivel superior ubicada en la Av. Carlos Julio Arosemena, con la facultad de Ciencias Físicas y Matemáticas como una de sus primeras entidades, que posteriormente en el año 1977 fue nombrada por el consejo universitario como Facultad de Ingeniería.

La Facultad de Ingeniería tiene como misión la formación de profesionales con sólidos conocimientos que les permita comprender, evaluar y resolver los problemas que la profesión requiera, sustentados en una visión humanística y apegados a la formación ética impartida por sus docentes.

La Coordinación Académica es el área responsable de la creación y difusión de eventos y actividades que se desarrollen en la Facultad de Ingeniería, los cuales son promocionados de forma autónoma por medio de redes sociales, correo electrónico o grupos de WhatsApp.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

En el mundo de las ciencias se presentan varios paradigmas o formas de pensamiento, entre ellas tenemos el positivismo, el estructuralismo, el empirismo, entre otros; lo que ha motivado a obtener diferentes formas de búsqueda, las mismas que deberán ser abordadas desde el punto de vista del investigador, teniendo como pilar fundamental el eje marcado según la propuesta del tema a trabajar. Para poder llevar a cabo un correcto proceso investigativo se puede hacer uso de métodos, técnicas, experimentos que brinden la verificabilidad necesaria de la propuesta que es objeto de estudio.

Sin embargo, el uso de las metodologías de investigación en base a cada uno de sus alcances han sido materia de estudio cada vez más discutido desde mediados del siglo XX hasta la actualidad han centrado su enfoque en dos vertientes: la metodología cualitativa y metodología cuantitativa. Se sostiene que el uso de estos dos enfoques, aplicados de manera conjunta notablemente enriquecerá cualquier propuesta de investigación.

3.1 Metodología cualitativa

Este trabajo tiene un enfoque cualitativo, del tipo descriptivo. Según los especialistas Sampieri, Fernández-Collado, & Pilar Baptista (2006) indican lo siguiente sobre este tipo de investigación:

“Las investigaciones cualitativas se fundamentan más en un proceso inductivo (explorar y describir, y luego generar perspectivas teóricas). Van de particular lo general. Por ejemplo, en un típico estudio cualitativo, el investigador entrevista a una persona, analiza los datos que obtuvo y saca algunas conclusiones; posteriormente, entrevista a otra persona, analiza esta nueva información, revisa sus resultados y conclusiones; del mismo modo, efectiva y analiza más entrevistas para comprender que busca. Es decir, procede dato por dato, hasta llegar a una perspectiva más general. En este tipo de enfoque las observaciones de los procesos se realizan sin interrumpir o tratar de inferir algún punto propio de vista, son receptados tal como se suscitan los acontecimientos”. (p. 9)

De manera directa se entiende el investigador bajo la metodología cualitativa hace uso de técnicas que permiten la recolección de datos como entrevista de tipo abierta, la observación base no estructurada, revisión bibliográfica de la información. Donde su grado de significancia va de lo particular a lo general, obteniendo todo tipo de información que sirve en lo posterior como fuente de datos o discusión. Se va a utilizar el método cualitativo debido a que se van a estudiar casos específicos mediante observación en sitio y entrevista personalizada.

Es descriptiva debido a que motiva su uso por basarse en la descripción, en los componentes a utilizar, subjetividad de los módulos en base a la propuesta de aplicación para creación e información de alertas, eventos y actividades que se generen en el campus universitario con geolocalización aplicado a la facultad de ingeniería.

En cuanto a la población de estudio, según Levin & Rubin (2004) se trata de “un conjunto de todos los elementos que se están estudiando, acerca de los cuales se intenta sacar conclusiones” (p.68) y en base a su estudio se pueden obtener datos referenciales acerca de gustos, tendencias, entre otros.

En investigación el factor a estudiar se conoce como el “conjunto de todos los casos que concuerdan con determinadas especificaciones” (Sampieri, Fernández-Collado, & Pilar Baptista, 2006), es muy importante que esta población se la sitúe de manera correcta en base a sus características de tiempo y lugar.

3.2 Técnicas para recolección de datos

Un instrumento de medición adecuado es aquel que registra datos observables que representan verdaderamente los conceptos que tiene en mente. o las variables que el investigador tiene en mente. En términos cuantitativos: capturo verdaderamente la realidad que deseo capturar. La función de la medición es establecer una correspondencia entre el mundo real y el mundo conceptual. El primero provee evidencia empírica, el segundo proporciona modelos teóricos para encontrar sentido a ese segmento del mundo real que estamos tratando de describir. (Sampieri, Fernández-Collado, & Pilar Baptista, 2006)

Los instrumentos de medición se consideran los recursos que se utilizan llevar un registro de la información obtenida, acerca de la o las variables que han sido sujeto de estudio por parte del investigador. Este tipo de documentos deben de cumplir básicamente tres puntos muy importantes: objetividad, confiabilidad y validez.

3.2.1 Entrevista

La herramienta a utilizar en base a lo que indica el método de investigación cualitativo son las entrevistas personalizadas, ya que servirán para el levantamiento de requerimientos en base a una conversación directa con las personas involucradas en la gestión de eventos en la facultad. Los entrevistados que serán: la coordinación académica de la facultad y los profesores a tiempo completo; podrán exponer sus experiencias y necesidades a la hora de difundir un evento.

Según el investigador Bravo (2013) define entrevista de la siguiente forma:

“La entrevista es una técnica de gran utilidad en la investigación cualitativa para recabar datos; se define como una conversación que se propone un fin determinado distinto al simple hecho de conversar. Es un instrumento técnico que adopta la forma de un diálogo coloquial. Canales la define como la comunicación interpersonal establecida entre el investigador y el sujeto de estudio, a fin de obtener respuestas verbales a las interrogantes planteadas sobre el problema propuesto”.

El utilizar una entrevista permite recabar información de una fuente directa, de manera concreta, propone un fin más amplio al poder conversar de manera directa con el entrevistado, conocer su perfil, preferencias o gustos claramente será de gran aporte para el proceso de la investigación. Para este proyecto de investigación se identificó como fuente primaria de generación y difusión de eventos a la coordinación académica y a los profesores a tiempo completo de la facultad de ingeniería.

Preguntas de la entrevista

- ¿Cuáles son las necesidades al momento de organizar y difundir un evento o actividad?
- ¿Cuáles son los medios de difusión con los que cuenta la facultad de ingeniería para hacer conocer sobre los eventos a desarrollarse?
- ¿Qué tan frecuentes y efectiva es la comunicación con sus estudiantes para difundir eventos y actividades?
- Si existiera una herramienta tecnológica que ayude a la creación y difusión de eventos de forma rápida y oportuna, ¿cree usted que mejoraría la comunicación entre la administración y los estudiantes?
- ¿Cuáles son las características que a usted le gustaría que tenga la aplicación informática?
- ¿Prefiere que la gestión y administración de los eventos sea centralizado o que cualquier miembro de la comunidad universitaria pueda crear un evento?

3.3 Análisis de resultados

Las entrevistas realizadas tuvieron como objetivo fundamental identificar las necesidades en lo que respecta a la difusión autónoma de eventos organizados por la facultad, para lo cual se consideró a la Coordinadora Académica, Ing. Ana Isabel Camacho Coronel, Mgs. y al Ing. José Miguel Erazo Ayón, Mgs. como Community Manager de las redes sociales de la Carrera de Computación.

Se pudo conocer que, para la creación de un evento, dependiendo de la necesidad se apoyan en el área de marketing para el diseño de artes, campañas de difusión y notificar sobre un evento por segmento de personas o si se considera necesaria la cobertura de medios de comunicación nacionales.

Los medios de difusión de la facultad son las redes sociales como Facebook, Instagram, correo electrónico y grupos de WhatsApp. La comunicación con los estudiantes es buena y normalmente se obtienen los resultados esperados con respecto a asistencia, y se trata de realizar un evento al mes en el que participen tanto alumnos como docentes.

Las características de la aplicación es que debe ser fácil de usar y que no implique mayor carga de trabajo, sino más bien un apoyo a la labor actual, además debería ser centralizada o que cualquier miembro pueda publicar un evento bajo la aprobación de un administrador.

3.4 Metodología de desarrollo para la propuesta técnica

3.4.1 Modelos para construcción de Software

Los modelos como su palabra lo indica ayudan a diseñar la propuesta de un sistema, cada tipo de estos modelos son usados de acuerdo a la ideología o perspectiva del investigador, para este caso en especial se propone el modelo en cascada.

El **Modelo de Cascada** va en la concepción de ordenar a manera rigurosa cada una de las etapas que intervienen para el modelado de la propuesta, cada parte de las fases las considera como procesos aislados unos de los otros, dicho de otra manera, una vez que una de las fases quedo finalizada, el programador iniciara con la otra, por lo que tiene una evolución lineal y sumativa. (Morales, 1998)

Al terminar cada una de las fases se lleva a cabo pruebas, la misma que determina si se puede seguir avanzado con las fases del desarrollo, esta metodología se considera como la pionera de otras propuestas que tienen que ver con la vida del software, siendo a la actualidad aun de mucha implementación, en base a su fuerte proceso de rendimiento.

Este tipo de modelo está conformado por las siguientes etapas:

1. Se procede con el análisis de los requerimientos: se debe de cubrir o informarse sobre las necesidades básicas, de los usuarios que utilicen el sistema; es muy importante levantar una memoria técnica donde se guarde todos los requisitos establecidos previamente; también se deberán de integrar los componentes, no permitiendo algún tipo de nuevo requerimiento una vez comenzado el desarrollo.

2. Propuesta en base al diseño: Se hace el diseño del código o de los algoritmos que serán necesarios según el levantamiento de información hecho a los usuarios, también se definirá el tipo de herramientas por utilizar.

3. Desarrollo: En esta etapa es donde se comenzará a escribir el código fuente, realizando algún tipo de prueba que permitan corregir los errores en la fase de desarrollo, si el lenguaje lo permite se crearan y reutilizaran bibliotecas, tratando que el desarrollo se efectuó de forma más ágil.

4. Acoplamiento y fase de pruebas: Los módulos que se encuentren los comienzan la etapa de ensamblaje para comenzar a darle forma al sistema, realizando pruebas en caliente, antes de proceder con su entrega al usuario.

5. Soporte o Mantenimiento: Es una de las partes más importantes y la de mayor tiempo en ejecución, básicamente consiste en corregir algún tipo de problema que presente el programa en su uso diario, tratando de mantener siempre la operación continua del sistema.

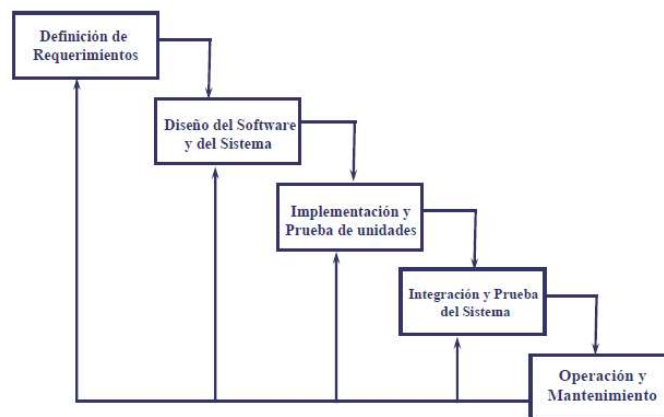


Figura 5. Tomado de Modelo en Cascada por Gutierrez (2011)

3.4.2 Entorno de desarrollo Microsoft Visual Studio 2017 Community

Para el desarrollo de este proyecto se ha optado por la utilización del IDE de Microsoft: Visual Studio 2017 Community el cual entre sus principales ventajas está la de tener un acuerdo de licencia gratuita para programadores, además de que tiene

incorporado una gran cantidad de tipos de proyectos como API, WCF, Soluciones para Smartphone, entre otros.

Visual Studio es una potente herramienta de desarrollo que se integra fácilmente con el diseñador, para poder crear imponentes sitios web, que se puedan integrar con otros tipos de tecnología, de manera amigable.

“Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todo el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML.” (Microsoft, 2017)

Estas potentes herramientas comparten un entorno de desarrollo, logrando motivar soluciones en diferentes tipos de lenguajes, pilar fundamental para motivar a un desarrollador a interactuar con este tipo de lenguaje.

3.4.3 Factores para escoger un proveedor de servicios de geolocalización

Según el análisis de (SmartyStreets, 2018) hay que evaluar los siguientes factores al momento de escoger un proveedor de servicios de geolocalización: exactitud, disponibilidad, velocidad, soporte técnico y documentación, precio.

Los proveedores a evaluados fueron: Google, TAMU, Loqate, Geocodio y SmartyStreets.

Exactitud: La primera y más obvia métrica por la cual un servicio de geocodificación puede ser juzgado es la precisión de los mismos geo códigos. Es importante tener en cuenta que los códigos geográficos menos exactos no significan que sea inútiles. En este ámbito "incorrecto" e "inexacto" son dos cosas completamente diferentes.

Cuando se trata de generar un código geográfico solo hay dos opciones: ir físicamente a la dirección en sí y escribir lo que indica el GPS, o usar las matemáticas

para interpolar la posición. El primero es obviamente el método más preciso, no puede ser más preciso que pintar una gran "X" roja en el lugar, a ese método se lo denomina precisión de "**rooftop level**".

El problema con el "**rooftop level**" es que es difícil de generar. Requiere que alguien vaya físicamente al sitio y eso tiene su costo. Entonces, lo que se gana en precisión, usualmente se paga en precio, es un intercambio.

El segundo método, llamado interpolación, consiste en tomar dos puntos conocidos y luego dividir el espacio entre ellos para obtener las ubicaciones estimadas. Brinda una precisión menos exacta, pero requiere menos esfuerzo y es más barato.

- Google: "**rooftop level**".
- Loqate: "**rooftop level**" en algunas partes, "interpolación" en otros.
- Geocodio: "**rooftop level**" en algunas partes, "interpolación" en otros.
- SmartyStreets: Otros métodos de precisión. (ZIP9)

Disponibilidad: es una métrica utilizada para medir cuánto tiempo y con qué consistencia ha estado un sistema en funcionamiento. No siempre se puede planificar cuándo vamos a necesitar el servicio, por lo que es necesario que esté listo en todo momento.

Supongamos que se llama al 911 porque el gato está atascado en un árbol y nadie responde, de hecho, aparece el mensaje "Su llamada no se puede completar en este momento". Esta sería una emergencia menor, por lo que no quedaría más que tomar una escalera y rescatar al gato uno mismo. Pero ¿Y si eso sucede una emergencia real?

Los usuarios ocasionales pueden no sentir esa necesidad, dependiendo de un servicio que no cuenta con tiempos de actividad de clase mundial. Sin embargo, los clientes recurrentes necesitarán cierta garantía de confiabilidad. Esa garantía generalmente viene en forma de un Acuerdo de Nivel de Servicio (o SLA); que, entre otras cosas, promete un tiempo de actividad que cumple o supera un cierto umbral. Sin embargo, algunos proveedores no hacen estos compromisos y otros solo ofrecerán mostrar su SLA a los clientes registrados y que pagan. En este último caso, realmente

no hay manera de saber qué tipo de confiabilidad está obteniendo antes de comenzar a usar el servicio.

Un usuario habitual de las API de geo codificación necesitará un sistema que garantice la disponibilidad del servicio los 365 días del año.

- **Google:** 99.9% Disponibilidad con SLA
- **TAMU:** No hay información sin registrarse.
- **Loqate:** No hay información sin registrarse.
- **Geocodio:** No posee SLA
- **SmartyStreets:** 99.98% Disponibilidad con SLA

Velocidad: métrica para medir el tiempo de respuesta en procesar 10 millones de direcciones.

- **Google:** 33 minutos (gratis), 17 minutos (pago)
- **Geocodio:** “Un par de minutos”
- **SmartyStreets** 5 segundos
- **TAMU:** Desconocido

Soporte técnico y documentación: Algo que podría haber pasado por alto, pero que es de importancia crítica, lo fácil que es lidiar con una API cuando algo sale mal. Un buen soporte técnico ayuda a resolver los problemas a medida que surgen. Una buena documentación ayuda tanto a prevenir problemas como a ayudar a los usuarios a resolverlos. Algunos proveedores son de primera categoría en este sentido, mientras que otros han dedicado sus esfuerzos a otras áreas.

- **Google:** documentación extensa, existen varias páginas y foros con soluciones a problemas comunes y proyectos de ejemplo para descargar.
- **TAMU:** No se ofrece documentación a menos que esté registrado; Contacto telefónico, chat y correo electrónico, pero no cuenta de Twitter

- Geocodio: Documentación extensa, incluye bibliotecas de múltiples idiomas; correo electrónico, chat, teléfono y Twitter ofrecidos para soporte técnico
- SmartyStreets: documentación extensa, bibliotecas de idiomas alojadas en Github; Teléfono, chat, correo electrónico y Twitter, todo ofrecido para soporte técnico

Precio: este es el punto que hace girar al mundo, el precio es a menudo el factor decisivo en la elección de un proveedor, y hay una gran variedad en los precios ofrecidos por estas compañías. Después de decidir qué servicios se necesitan, así como las funciones que se desean, el costo es el punto crítico en el proceso de elección de un proveedor de geo codificación.

- Loqate: \$\$\$\$
- Google: Desarrolladores: regalan \$200 mensuales y hay que vincular una forma de pago. Premium \$\$\$ /? (el costo inicial es por excedentes; debe solicitar una cotización de los contratos)
- Geocodio: \$
- SmartyStreets: \$\$
- TAMU: \$

3.4.4 Google Maps Platform

Google Maps Platform son servicios basados en geolocalización que sirven para visualizar mapas, trazar rutas o ver información acerca de un lugar específico. Con Google Maps se puede ver todo el mundo con mapas estáticos o dinámicos, Google Routes ayuda a los usuarios a encontrar la mejor manera de ir desde un punto A hasta X con datos y actualizaciones de tráfico en tiempo real; y Google Place ayuda al usuario a descubrir el mundo con contenido rico en detalles sobre más de 100 millones de lugares. Además, permite encontrar lugares específicos mediante números de teléfono, indicadores en tiempo real y direcciones. (Google, 2018)

CAPÍTULO IV

PROPUESTA TECNOLÓGICA

4.1 Introducción

Dentro de las funciones que desempeñan los profesores a tiempo completo y la coordinación académica de la facultad de ingeniería, y con el fin de atender las necesidades relacionadas con la creación y difusión de eventos de tipo académicos, culturales, educativos y sociales, se ha diseñado y desarrollado un sistema web, que busca optimizar los recursos existentes para proveer resultados en el ámbito de la comunicación entre la facultad y los estudiantes de manera eficaz, como requerimiento específico para agilizar los siguientes procesos:

Crear y difundir eventos: actualmente la facultad se apoya en redes sociales, como Facebook o Instagram, para la difusión de eventos. Esto sin duda ayuda en esa labor, sin embargo, se dependería de tener “agregados” o “seguir” a quienes se requiera hacer llegar el mensaje. Por tanto, es necesaria una herramienta tecnológica autónoma que, primero sea fácil, sencilla y dinámica para la creación de eventos y segundo que difunda la actividad ya que se cuenta con los datos de contacto de los estudiantes que vendrían a ser los principales usuarios.

Notificar y alertar sobre los eventos: luego de difundir el evento viene la tarea de notificar, y por otra parte mantener en alerta cuando la actividad esté por comenzar. Cabe recalcar que esta tarea se la realiza vía correo electrónico, lo cual resulta un poco tedioso para quienes son los encargados de hacerlo. Por ese motivo se desarrolló la funcionalidad de que dicho proceso sea automático en base a los datos que se ingresan al crear un evento en la aplicación web, haciendo de eso un trabajo más sencillo y dinámico.

4.2 Objetivo

Facilitar a las personas encargadas de la gestión de eventos, la creación y difusión de actividades de índole académico, cultural, educativo, etc., con el fin de que, por un lado, la labor de comunicar sea de forma automatizada, rápida y exacta en cuanto a su ubicación gracias a la geolocalización; y por otro lado que los receptores

tengan la opción de interactuar con los eventos por medio de comentarios, e indicar y si les interesa y/o asistirán al evento para que puedan ser notificados y alertados.

4.3 Descripción del aplicativo web

Este aplicativo está compuesto por 3 opciones principales que son: bandeja principal, la gestión de eventos y la gestión de usuarios.

4.3.1 Módulos y componentes de la solución

La bandeja principal o dashboard es la primera pantalla que el usuario, sea administrador o visitante, verá después de haber iniciado sesión. En esta pantalla se podrán visualizar todos los eventos creados e interactuar con los mismos por medio de comentarios, indicar si le interesa o si asistirá. Por otra parte, al administrador le servirá para visualizar cuantas personas han comentado sobre un evento, o para ver la cantidad de asistentes.

La gestión de eventos es módulo más importante de la aplicación ya que en él se van a crear los eventos con sus respectivas propiedades como el nombre, fecha de inicio, fecha de fin, observaciones y se especifica mediante geolocalización la ubicación del mismo.

La gestión de usuarios es un módulo disponible para el administrador del sistema, ya que le permitirá crear y resetear contraseñas de usuario, además de asignarle el rol de administrador si es necesario.

4.3.2 Herramientas tecnológicas

La aplicación web para creación y difusión de eventos con geolocalización se desarrolló con el IDE de Microsoft: Visual Studio 2017 Community, empleando la arquitectura de diseño ASP.NET MVC y utilizando los lenguajes de programación C# y JavaScript; como servidor de base de datos SQL Server 2016 habilitando la característica de Server Broker para el envío de notificaciones a la aplicación sobre los cambios que se realicen a la base, cuyo hosting se encuentra con el proveedor SmaterASP.net con un periodo de evaluación de 3 meses, después de eso el valor mensual es de 2.95 USD.

La aplicación web se encuentra publicada en la nube con el servicio de aplicaciones de Microsoft Azure, tiene habilitado el certificado digital de comunicación segura, y además hace uso de WebSockets para la comunicación bidireccional por protocolo TCP entre el servidor y el navegador. El servicio para hostear la aplicación en la nube con Azure se encuentra en un plan gratuito para 10 conexiones simultáneas y 1GB de espacio para los archivos de la aplicación.

Y para seleccionar el proveedor de geolocalización se consideró que el método más exacto de geolocalización es el “rooftop level”, que la documentación para soporte y desarrollo mientras más extensa y diversa es, facilita el uso del servicio, y que mientras se den beneficios a desarrolladores en lo que respecta a costos, el API de Google es la mejor opción por lo cual se usará para el desarrollo de esta aplicación web.

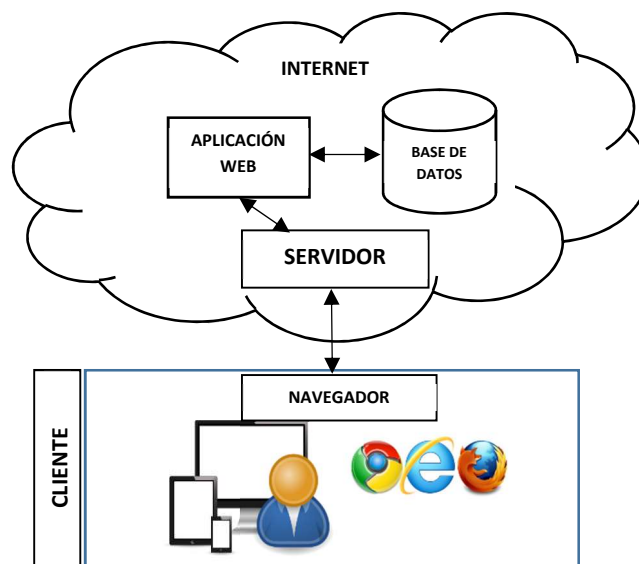


Figura 6. Arquitectura de la Aplicación. Fuente: el autor.

4.3.3 Procesos

Creación de usuario: como punto de partida para comenzar a utilizar la aplicación es necesario la creación del usuario, que, con tan solo especificar un nombre y un correo electrónico, se lo registra en el sistema y se autogenera una contraseña que será enviada a la dirección ingresada.

Creación de eventos: este proceso consiste en que el administrador del sistema tenga la funcionalidad de crear eventos asignándole un título, clasificarlo en base a su

tipo (social, académico, educativo, etc.), especificar cuando inicia y cuando termina, agregar una observación, indicar exactamente dentro del mapa donde se va a llevar a cabo, y especificar a quienes se va a realizar la notificación del mismo cumpliendo con el objetivo fundamental que es la comunicación entre la universidad y los estudiantes apoyado en la tecnología.

Notificar y alertar sobre la ejecución de los eventos: los usuarios, tanto alumnos como maestros, pueden interactuar con los eventos por medio de comentarios y dos acciones específicas que consisten en indicar si se está interesado y/o si asistirá a la gestión organizada por el administrador. Esto le servirá al aplicativo para que notifique y alerte sobre el comienzo de la actividad, con lo cual se mantiene informado al usuario y lo ayuda a planificar con anticipación su participación en los mismos.

4.3.4 Diseño de la aplicación web

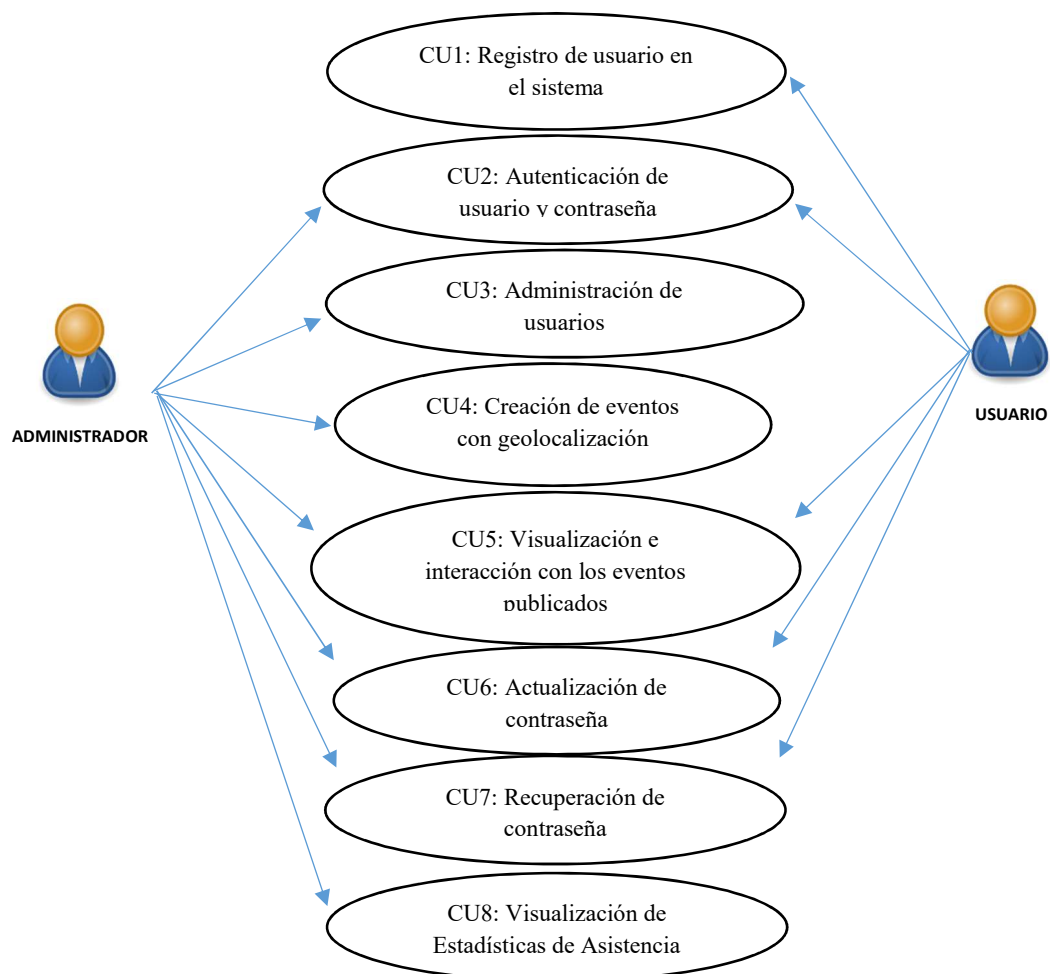


Figura 7. Casos de uso de la aplicación web. Fuente: el autor

Descripción de los actores: a continuación, se describen los actores de los casos de uso con su respectivo nombre y limitaciones.

N°	ACTOR	NOMBRE	DESCRIPCIÓN	LIMITACIONES
1	PRIMARIO	ADMINISTRADOR	Persona encargada de crear los eventos en la aplicación con su respectiva geolocalización y el envío de la notificación automática. También será responsable de administrar los usuarios.	<ul style="list-style-type: none"> - Crear eventos asignando la geolocalización. - Administrar usuarios. - Comentar eventos.
2	SECUNDARIO	USUARIO	Visualiza e interactúa con los diferentes eventos creados en la aplicación.	<ul style="list-style-type: none"> - Comentar eventos. - Indicar si le interesa el evento. - Indicar si asistirá al evento.

Tabla 1. Descripción de los actores. Fuente: el autor

Descripción de los casos de uso

CASO DE USO N°	NOMBRE	DESCRIPCIÓN	ACTORES
CU1	Registro de usuario en el sistema	El usuario ingresa al portal web en la opción de crear cuenta. Digita un usuario y un correo electrónico y click en registrar. El sistema valida datos registrados y genera una contraseña que se envía al correo electrónico.	Usuario
CU2	Autenticación de usuario y contraseña	Usuario ingresa al portal web, digita su usuario y contraseña. El sistema autentica las credenciales ingresadas.	Usuario, Administrador
CU3	Administración de usuarios	Se ingresa a la opción del menú de administrar usuarios. Se permite crear o modificar usuarios y los roles asignados.	Administrador
CU4	Creación de eventos con geolocalización	Mediante el aplicativo web en la bandeja de entrada estará habilitado el botón de crear evento. Se registran los datos principales, se especifica geolocalización y se graba el formulario. Se genera una notificación a los invitados al evento y cuando el evento este por comenzar se genera una alerta.	Administrador
CU5	Visualización e interacción con los eventos publicados	Mediante el aplicativo web se podrá visualizar la bandeja principal, la cual contiene todos los eventos creados por el administrador. Se podrá comentar, indicar si se está interesado o si asistirá al evento.	Usuario, Administrador
CU6	Actualización de contraseña	Mediante el aplicativo web en la barra superior derecha se permitirá la actualización de la contraseña.	Usuario, Administrador
CU7	Recuperación de contraseña	Mediante el aplicativo web al ingresar a la página, se podrá acceder a la opción de recuperar contraseña, se ingresará una dirección de correo electrónico válida y el sistema generará una nueva clave que será notificada.	Usuario, Administrador
CU8	Visualización de Estadísticas de Asistencia	Mediante el aplicativo web en el menú de opciones se podrá acceder a las estadísticas de asistencia por evento.	Administrador

Tabla 2. Descripción de los casos de uso. Fuente: el autor

Modelo Entidad Relación

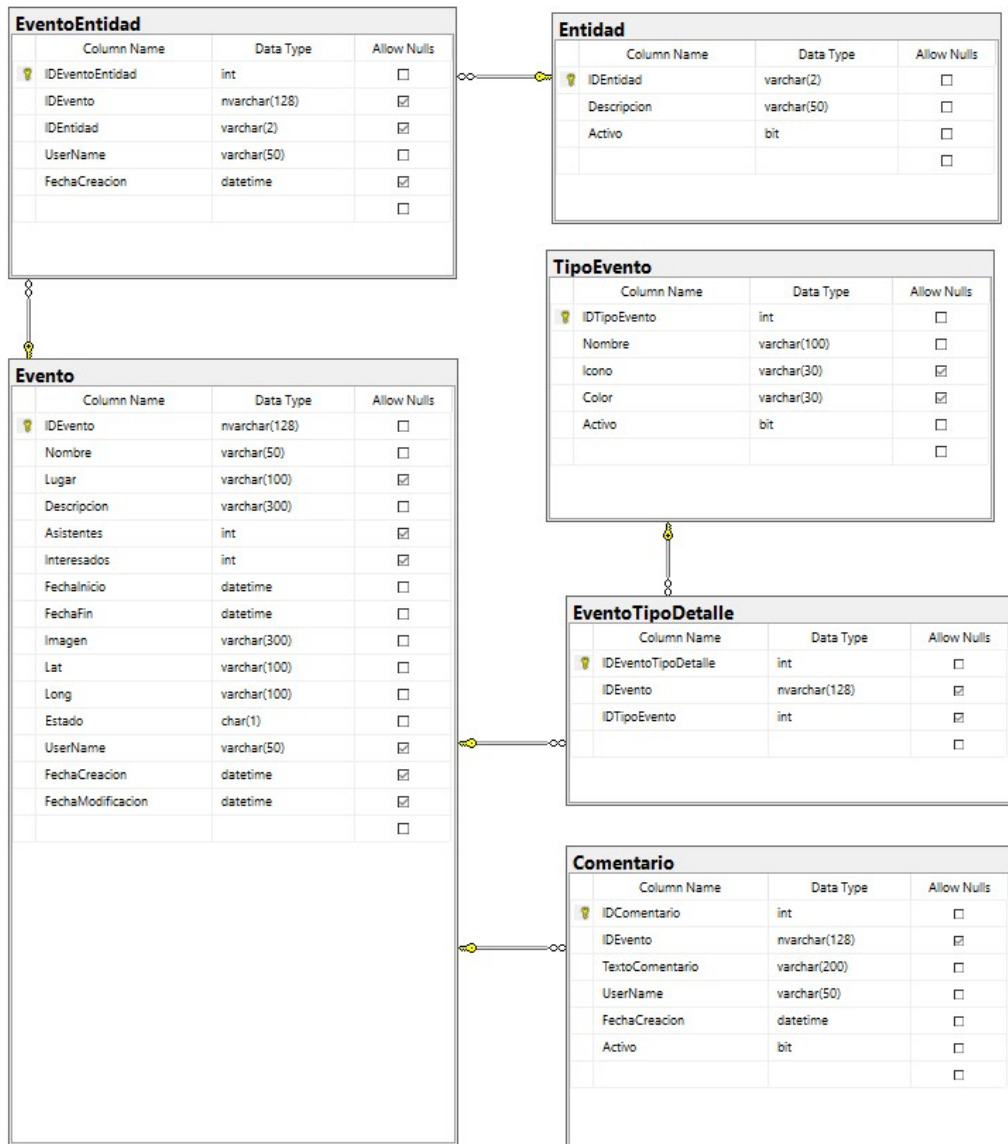


Figura 8. Tablas relacionadas al core de la aplicación. Fuente: el autor

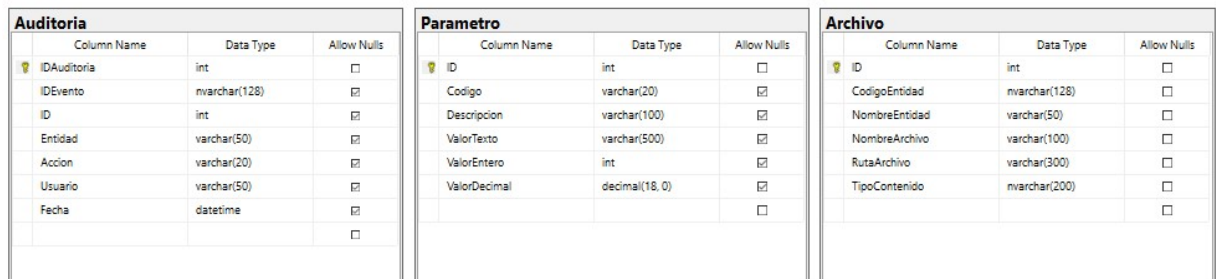


Figura 9. Tablas aisladas. Fuente: el autor

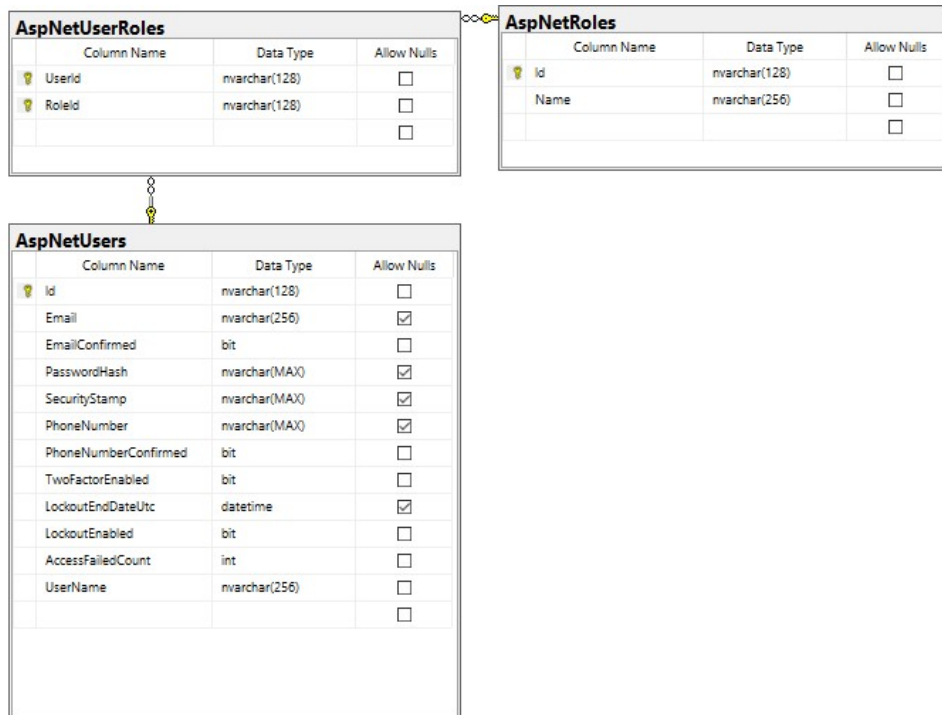


Figura 10. Tablas relacionadas a la autenticación de la aplicación. Fuente: el autor

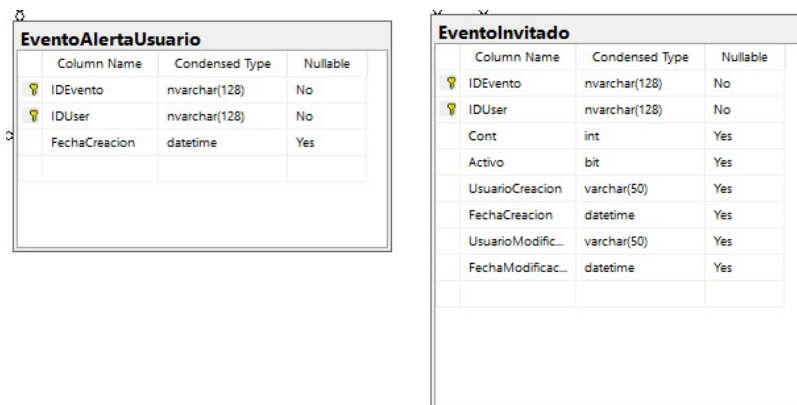


Figura 11. Tablas relacionadas a la notificación de eventos. Fuente: el autor

Diccionario de Datos

Tabla: Evento

Objetivo: estructura que contiene los datos principales del evento.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDEvento	Texto	256	clave primaria de eventos auto incrementable
	Nombre	Texto	50	nombre del evento
	Lugar	Texto	100	lugar del evento
	Descripcion	Texto	300	descripción del evento
	Asistentes	Numérico	4	asistentes al evento
	Interesados	Numérico	4	interesados al evento
	FechaInicio	Fecha y Hora	8	fecha de inicio del evento
	FechaFin	Fecha y Hora	8	fecha de fin del evento
	Imagen	Texto	300	ruta de la imagen
	Lat	Texto	100	latitud de la ubicación del evento
	Long	Texto	100	longitud de la ubicación del evento
	Estado	Texto	1	estado del evento
	UserName	Texto	50	usuario que creó el evento
	FechaCreacion	Fecha y Hora	8	fecha de creación del evento
	FechaModificacion	Fecha y Hora	8	fecha de modificación del evento

Tabla 3. Tabla Eventos. Fuente: el autor

Tabla: Entidad

Objetivo: almacena las entidades de la aplicación, identificadas como Me Interesa y Asistiré.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDEntidad	Texto	2	clave primaria
	Descripcion	Texto	50	descripción de la entidad
	Activo	Boleano	1	indica si está activo o no

Tabla 4. Tabla Entidad. Fuente: el autor

Tabla: EventoEntidad

Objetivo: relaciona uno o varios eventos o una o varias entidades.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDEventoEntidad	Numérico	4	clave primaria auto incrementable
FK	IDEvento	Texto	256	clave foránea del evento
FK	IDEntidad	Texto	2	clave foránea de la entidad
	UserName	Texto	50	usuario que registra la relación
	FechaCreacion	Fecha y Hora	8	fecha de creación del registro

Tabla 5. Tabla EventoEntidad. Fuente: el autor

Tabla: TipoEvento

Objetivo: guarda los diferentes tipos eventos que se puedan realizar, p. e. académicos, cultural, social, etc.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDTipoEvento	Numérico	4	clave primaria auto incrementable
	Nombre	Texto	100	nombre del tipo de evento
	Icono	Texto	30	icono para mostrar en pantalla
	Color	Texto	30	campo opcional para personalizar el color de visualización
	Activo	Boleano	1	indica si está activo o no

Tabla 6. Tabla TipoEvento. Fuente: el autor

Tabla: EventoTipoDetalle

Objetivo: almacena la relación del evento con uno o varios tipos de eventos.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDEventoTipoDetalle	Numérico	4	clave primaria auto incrementable
FK	IDEvento	Texto	256	clave foránea del evento
FK	IDTipoEvento	Numérico	4	clave foránea del tipo de evento

Tabla 7. Tabla EventoTipoDetalle. Fuente: el autor

Tabla: Comentario

Objetivo: guarda los comentarios que se realicen sobre cada evento.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDComentario	Numerico	4	clave primaria auto incrementable
FK	IDEvento	Texto	256	clave foránea del evento
	TextoComentario	Texto	200	texto del comentario registrado
	UserName	Texto	50	nombre del usuario que ingresa el comentario
	FechaCreacion	Fecha y Hora	8	fecha de creación del registro
	Activo	Boleano	1	indica si está activo o no

Tabla 8. Tabla Comentario. Fuente: el autor

Tabla: AspNetRoles

Objetivo: almacenar los distintos roles que se manejan en la aplicación.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	Id	Texto	256	clave primaria auto incrementable
	Name	Texto	512	nombre del rol

Tabla 9. Tabla AspNetRoles. Fuente: el autor

Tabla: AspNetUserRoles

Objetivo: relacionar los usuarios con uno o varios roles.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
FK	UserId	Texto	256	clave foranea del usuario
FK	RoleId	Texto	256	clave foranea del rol

Tabla 10. Tabla AspNetUserRoles. Fuente: el autor

Tabla: AspNetUsers

Objetivo: esta estructura contiene los datos principales del usuario, además la contraseña se mantiene encriptada.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	Id	Texto	256	clave primaria de usuarios
	Email	Texto	512	correo electrónico
	EmailConfirmed	Boleano	1	indica si el correo ha sido confirmado o no
	PasswordHash	Texto		para el almacenamiento, y para verificar que una contraseña proporcionada coincida con un hash almacenado
	SecurityStamp	Texto		se utiliza para realizar un seguimiento de los cambios realizados en el perfil de usuario
	PhoneNumber	Texto		número de teléfono del usuario
	PhoneNumberConfirmed	Boleano	1	indica si el número de teléfono ha sido confirmado o no
	TwoFactorEnabled	Boleano	1	para autenticación usando SMS o Email
	LockoutEndDateUtc	Fecha y Hora	8	especifica la fecha de bloqueo
	LockoutEnabled	Boleano	1	significa que el usuario puede ser bloqueado, no que el usuario esté bloqueado.
	AccessFailedCount	Numérico	4	contador de accesos fallidos
	UserName	Texto	512	nombre de usuario

Tabla 11. Tabla AspNetUsers. Fuente: el autor

Tabla: Auditoria

Objetivo: registra auditoría cuando se ejecuten los eventos de inserción, actualización o eliminación de las tablas de Eventos, Comentarios y EventoEntidad.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDAuditoria	Numérico	4	clave primaria auto incrementable
	IDEvento	Texto	256	clave del evento
	ID	Numérico	4	clave de la entidad
	Entidad	Texto	50	nombre de la entidad
	Accion	Texto	20	acción que se realizó
	Usuario	Texto	50	usuario que realizó la acción
	Fecha	Fecha y Hora	8	fecha que se genera la acción

Tabla 12. Tabla Auditoria. Fuente: el autor

Tabla: Parametro

Objetivo: guarda los parámetros de la aplicación.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	ID	Numérico	4	clave primaria auto incrementable
	Codigo	Texto	20	código del parámetro, campo único
	Descripcion	Texto	100	descripción del parámetro
	ValorTexto	Texto	500	valor tipo texto del parámetro
	ValorEntero	Numérico	4	valor entero del parámetro
	ValorDecimal	Decimal	9	valor decimal del parámetro

Tabla 13. Tabla Parametro. Fuente: el autor

Tabla: Archivo

Objetivo: contiene el listado de rutas de las imágenes que se cargan al momento de generar un evento, además de las imágenes de perfil de cada usuario.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	ID	Numérico	4	clave primaria auto incrementable
	CodigoEntidad	Texto	256	código de la entidad relacionada al archivo
	NombreEntidad	Texto	50	nombre de la entidad relacionada al archivo
	NombreArchivo	Texto	100	nombre del archivo
	RutaArchivo	Texto	300	ruta del archivo
	TipoContenido	Texto	400	tipo de contenido del archivo

Tabla 14. Tabla Archivo. Fuente: el autor

Tabla: EventoAlertaUsuario

Objetivo: contiene los registros de las alertas generadas automáticamente por evento y usuario.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDEvento	Numérico	4	Clave primaria compuesta por el ID del evento y el ID del usuario.
PK	IDUser	Numérico	4	Clave primaria compuesta por el ID del evento y el ID del usuario.
	FechaCreacion	Fecha y Hora	8	Fecha y hora en que se generó la alerta.

Tabla 15. Tabla EventoAlertaUsuario. Fuente: el autor

Tabla: EventoInvitado

Objetivo: contiene los registros de los invitados por evento.

Llave	Nombre	Tipo de Dato	Tamaño	Descripción
PK	IDEvento	Numérico	4	Clave primaria compuesta por el ID del evento y el ID del usuario.
PK	IDUser	Numérico	4	Clave primaria compuesta por el ID del evento y el ID del usuario.
	Cont	Numérico	4	Contador
	Activo	Booleano	1	Estado del registro
	UsuarioCreacion	Texto	100	Usuario creador del registro.
	FechaCreacion	Fecha y Hora	8	Fecha y hora en que se creó la notificación.
	UsuarioModificacion	Texto	100	Usuario modificador del registro.
	FechaModificacion	Fecha y Hora	8	Fecha y hora de modificación.

Tabla 16. Tabla EventoInvitado. Fuente: el autor

Diseño de interfaz gráfica de la aplicación web

Pantalla de Inicio de Sesión: el usuario digitará su usuario y contraseña, el sistema valida los datos ingresados y muestra mensaje si son incorrectos.



Figura 12. Vista de inicio de sesión. Fuente: el autor

Pantalla de Registro de Usuario: el usuario digitará un nombre válido para iniciar sesión y un correo electrónico al cual se enviará la contraseña autogenerada.

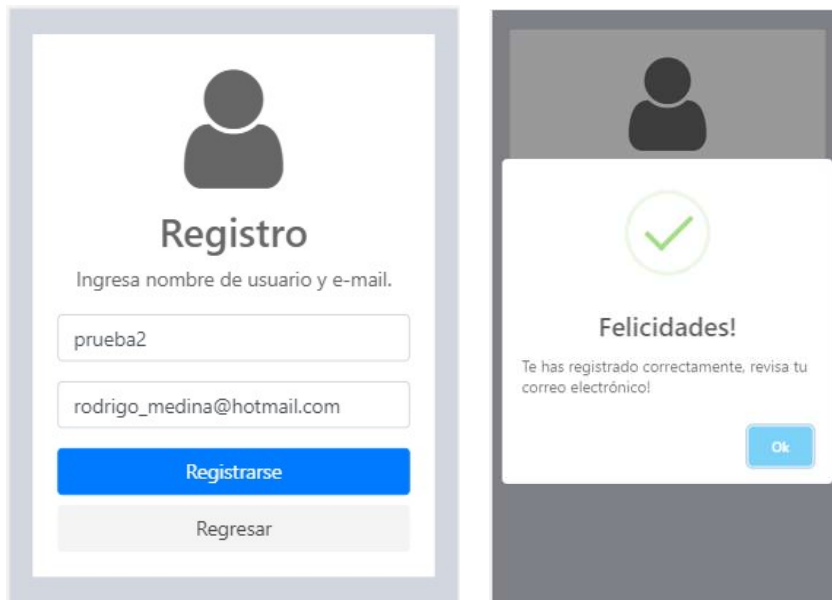


Figura 13. Vista de registro de usuario. Fuente: el autor

Pantalla de Recuperación de Contraseña: el usuario digitará el correo electrónico con el cual se registró, el sistema lo valida y envía una nueva contraseña a esa dirección.

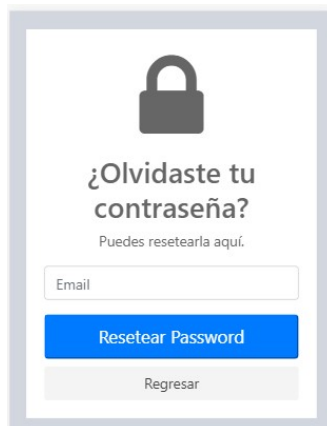


Figura 14. Vista de recuperación de contraseña. Fuente: el autor

Pantalla de Creación de Eventos: el usuario digitará los datos principales del evento como nombre, tipo de evento, fecha de inicio y fin, observaciones y además le permitirá seleccionar una imagen y la ubicación geográfica en donde se realizará la actividad.

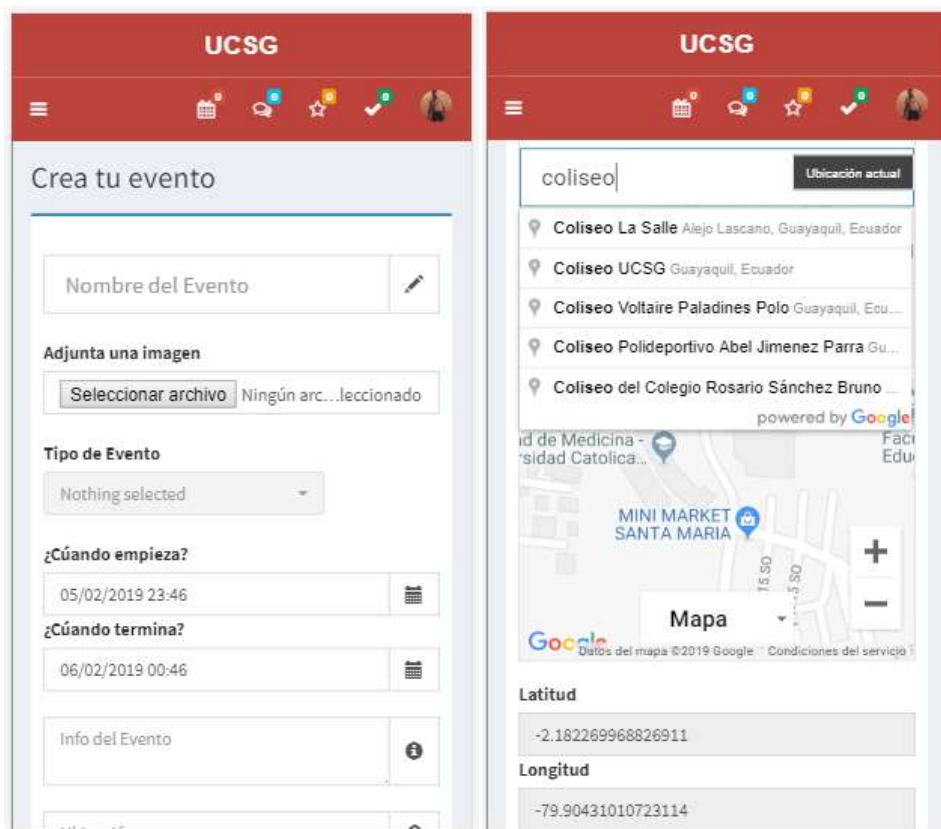


Figura 15. Vista de creación de evento. Fuente: el autor



Figura 16. Evento creado satisfactoriamente. Fuente: el autor

Pantalla de Bandeja Principal: esta pantalla es la que visualizarán todos los usuarios para interactuar con los eventos escribiendo comentarios o indicando que les interesa o asistirán al evento.

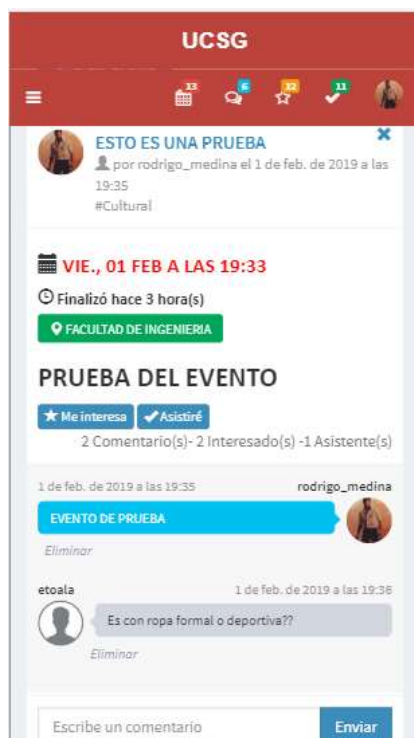


Figura 17. Vista de la bandeja principal de la aplicación desde un dispositivo móvil. Fuente: el autor

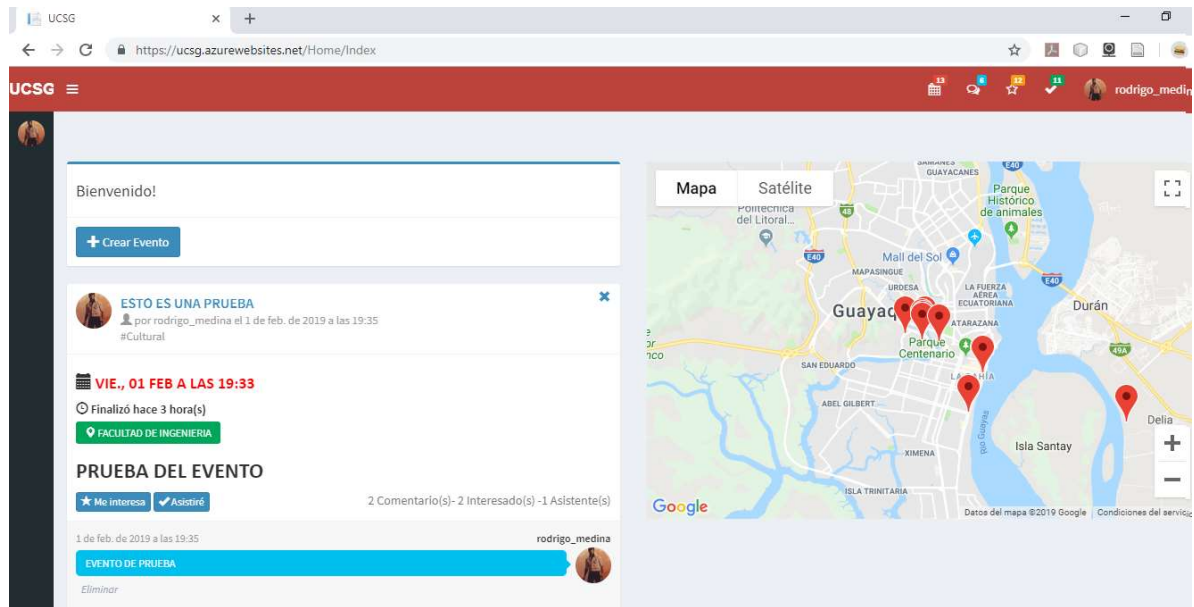


Figura 18. Vista de la bandeja principal de la aplicación desde un PC. Fuente: el autor



Figura 19. Sección para interactuar con los eventos. Fuente: el autor

Pantalla de Estadísticas de Asistencia: esta pantalla permite visualizar las estadísticas de asistencia de los eventos por medio de dos gráficos de tipos lineal y dona.



Figura 20. Pantalla de Estadísticas gráfico tipo Lineal. Fuente: el autor



Figura 21. Pantalla de Estadísticas gráfico tipo Dona. Fuente: el autor

Pantalla para Administración de Usuarios: esta pantalla permite administrar los usuarios registrados en el sistema, entre sus opciones están las de modificar el correo electrónico, asignarle otros roles y bloquearlo del sistema.

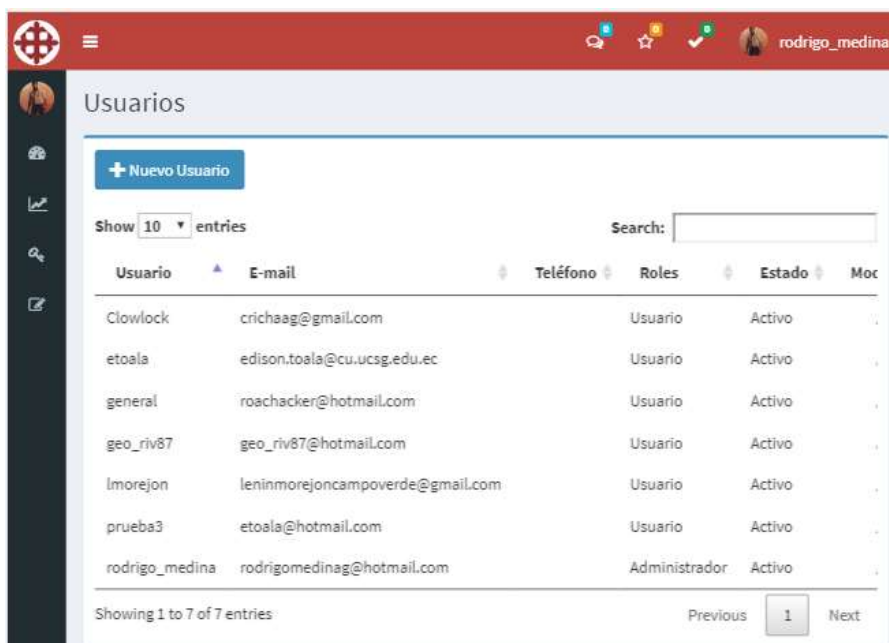


Figura 22. Pantalla de Administración de Usuarios. Fuente: el autor

4.4 Plan de pruebas

4.4.1 Objetivos

El siguiente plan de pruebas tiene como objetivo especificar los elementos del aplicativo que se van a realizar validaciones y verificaciones, además de obtener información sobre posibles errores o fallas que se presenten para realizar las correcciones del caso.

4.4.2 Alcance

Con el siguiente plan de pruebas se busca dar la mayor confiabilidad de que la aplicación se va a ejecutar de forma correcta con el resultado esperado el cual consiste en identificar todos los errores que sean posibles ejecutando pruebas sobre características funcionales del aplicativo.

Test	Objetivo	Participantes	Ambiente	Método
Funcional	Detectar errores en la implementación de requerimientos	Tester, Analista	Desarrollo	Funcional

Tabla 17. Prueba funcional de aplicación. Fuente: el autor

Las opciones del sistema a probar son:

- CU1: Registro de usuario en el sistema
- CU4: Creación de eventos con geolocalización
- CU8: Visualización de estadísticas de asistencia

4.4.2 Resultados de las pruebas

Las pruebas se desarrollaron con normalidad, en la mayoría de casos los resultados fueron los esperados, salvo en la creación de eventos que se tuvieron que agregar validaciones ya que no todos los dispositivos poseen habilitada la opción de geolocalización. Las matrices de pruebas se encuentran en el Anexo D y las imágenes con los resultados en el Anexo E.

4.5 Requerimientos para la implementación

4.5.1 Requisitos de hardware

- Cualquier dispositivo como Tablet, Smartphone o PC que tenga instalado un navegador. Se recomienda Google Chrome.
- Acceso a internet.
- Opcional: habilitar opción de ubicación.

4.5.2 Servicios en la nube

Para el servicio de **base de datos** se realizó la suscripción por periodo de pruebas con el proveedor SmarterASP.NET el cual brinda 60 días para pruebas y posterior a eso tiene un costo de 2.95 USD mensuales. La ventaja de este proveedor es que su motor de base de datos permite habilitar la característica de Server Broker, fundamental para obtener los cambios que se realicen en tiempo real y mostrar las notificaciones en la bandeja principal de la aplicación web.

The screenshot displays the SmarterASP.NET website with four hosting plans. The ASPNET BASIC plan is highlighted with a blue border. The plans are as follows:

Plan Name	Price	Key Features
ASPNET BASIC	\$2.95 a month	Host 1 Website, Unlimited Domain Pointers, Unlimited Space/Transfers, US/Europe Datacenter, FREE Activation!
ADVANCE	\$4.95 a month	Host 6 Websites, Unlimited Domain Pointers, Unlimited Space/Transfers, US/Europe Datacenter, FREE Activation!
ASPNET PREMIUM	\$7.95 a month	Unlimited Websites, Unlimited Domain Pointers, Unlimited Space/Transfers, US/Europe Datacenter, FREE Activation!
AMAZING AFFILIATE PLAN	FREE	Huge Payouts!, \$5 USD for Free Trial Signup, 40% Commission on New Sales, 20% Commission on Renewals, Withdraw Anytime, Recurring Commission, Real-Time Affiliate Panel

Figura 23. Planes de servicios de SmarterASP.NET. Fuente: SmarterASP.NET

Para el servicio de **hosting de la aplicación** se realizó la suscripción en Microsoft Azure, el cual da una gran ventaja a los desarrolladores de hacer uso de sus servicios con un plan gratuito sin límite de tiempo y que permite tener la aplicación en internet, con certificado de comunicación segura, 1GB para los archivos de la aplicación y hasta 10 conexiones simultaneas.

The screenshot shows the Microsoft Azure pricing page for application services. The 'FREE Try for free' plan is highlighted with a blue border. The table below summarizes the pricing and features:

Feature	FREE Try for free	SHARED Environment for dev/test	BASIC Dedicated environment for dev/test	STANDARD Run production workloads	PREMIUM Enhanced performance and scale	ISOLATED High-Performance, Security and Isolation
Web, mobile, or API apps	10	100	Unlimited	Unlimited	Unlimited	Unlimited
Disk space	1 GB	1 GB	10 GB	50 GB	250 GB	1 TB
Maximum instances	-	-	Up to 3	Up to 10	Up to 20	Up to 100*
Custom domain	-	Supported	Supported	Supported	Supported	Supported
Auto Scale	-	-	-	Supported	Supported	Supported
VPN hybrid connectivity	-	-	-	Supported	Supported	Supported
Network Isolation	-	-	-	-	-	Supported
Price per hour	Free	\$0.013	\$0.075	\$0.10	\$0.20	\$0.40

Figura 24. Planes de servicios de aplicaciones de Microsoft Azure. Fuente: Microsoft Azure

Para la funcionalidad de **geolocalización** se requirió del API de Google para lo cual se realizó la suscripción en la plataforma de servicios Google Maps Platform, el cual brinda un cupo de 200USD gratis mensualmente por 12 meses sin ningún tipo de restricción.

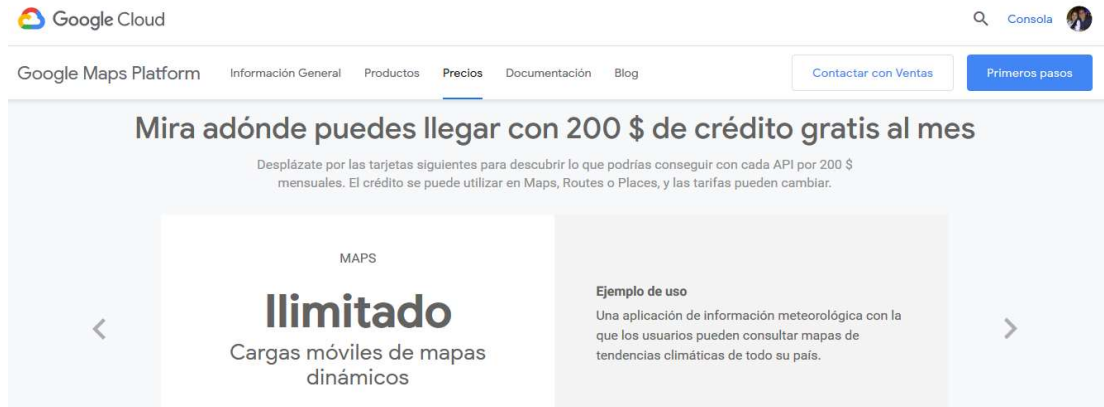


Figura 25. Planes de servicios de Google Maps Platform. Fuente: Google

Google™ Factura		Número de factura: 3544914759		
Fuente: My Project [my-project-1539917859906]				
Producto	Tipo de recurso	Intervalo	Uso	Importe (USD)
Maps API	Dynamic Maps	1 ene. - 30 ene.	1237 recuentos	8.66
Places API	Query Autocomplete - Per Request	1 ene. - 25 ene.	687 recuentos	1.94
Places API	Places Details	1 ene. - 25 ene.	53 recuentos	0.90
Directions API	Directions	1 ene. - 22 ene.	143 recuentos	0.72
Places API	Places - Text Search	1 ene. - 24 ene.	21 recuentos	0.67
Places API	Atmosphere Data	1 ene. - 24 ene.	73 recuentos	0.37
Places API	Contact Data	1 ene. - 24 ene.	73 recuentos	0.22
Credit	Maps Free Tier	1 ene. - 30 ene.		-13.48
Subtotal en USD				USD 0.00
Impuesto (0%)				USD 0.00
Total en USD				USD 0.00

Figura 26. Detalle de factura generada por uso de API de Google. Fuente: el autor.

Para el envío de **notificaciones por correo** se hace uso de SendGrid, uno de los servicios que ofrece Azure, la cual permite el envío de hasta 25mil correos al mes de forma gratuita.

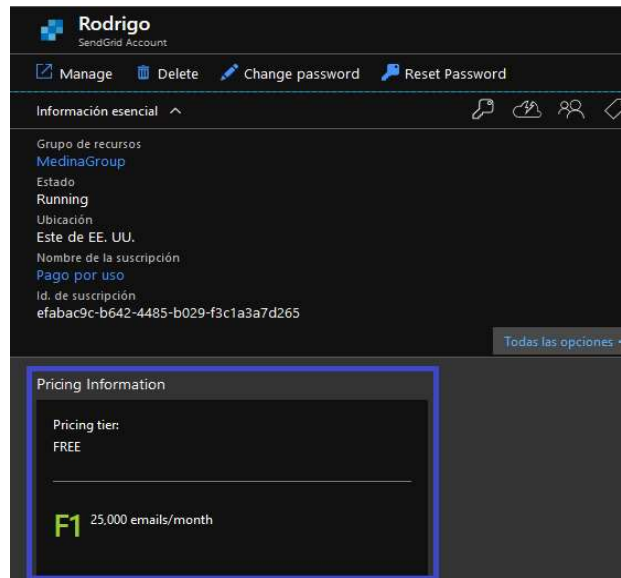


Figura 27. Detalle de plan de envío de correos por SendGrid. Fuente: Microsoft Azure

Uno de los servicios en la nube más importantes para el desarrollo de este proyecto es el que permite el envío automático de las notificaciones, se lo realizó utilizando **Azure Schedule** que funciona censando en la base de datos la fecha de inicio del evento con una frecuencia de 1 minuto por medio de la llamada a un api creado en la controladora de alertas de la aplicación.

Se seleccionó el plan de tarifa más económico con un costo de 13.99USD al mes y posee redundancia global, frecuencia máxima de llamadas cada minuto, ejecuciones ilimitadas de trabajos entre otras características.

	STANDARD	P10 PREMIUM	P20 PREMIUM
Price per month ¹	\$13.99 per unit	\$139.99 per unit	\$1,399.99 per unit
Job executions	Unlimited	Unlimited	Unlimited
Maximum execution frequency	Every minute	Every minute	Every minute
Job collections	10/unit	10,000/unit	5,000/unit
Jobs per collection	50	50	1,000
Scaling	10	1 premium unit. Contact us for more.	1 premium unit. Contact us for more.
Outbound authentication	✓	✓	✓

¹ Billing is prorated hourly. Price above is based on 730 hours per month. One standard unit is billed for every 10 job collections (or fraction) created, prorated hourly. Similarly, one premium unit is billed for every 10,000 job collections (or fraction) created, prorated hourly. Jobs are aggregated across geographies.

Figura 28. Planes de servicios de Azure Schedule. Fuente: Azure

4.6 Resultados esperados del aplicativo

Mediante la implementación de la aplicación web con geolocalización se logrará lo siguiente:

- Registrarse en el sistema de forma autónoma para poder visualizar e interactuar con los eventos.
- Autenticarse en la aplicación web con los datos ingresados en el formulario de registro.
- Restaurar y restablecer contraseña.
- Crear eventos asignándoles una ubicación geográfica del lugar donde se va a desarrollar la actividad.
- Difundir los eventos creados mediante envíos automáticos de correo electrónico.
- Alertar sobre los eventos mediante envíos automáticos de correo electrónico.
- Permitir al usuario interactuar con los eventos indicando si le interesa o si asistirá al evento además de poder comentarlo.

CONCLUSIONES

Durante el proceso de desarrollo del presente trabajo de titulación, se pudo identificar que los medios de difusión institucionales que se utiliza en la UCSG son: redes sociales como Facebook e Instagram, correo electrónico y grupos de WhatsApp, a través de entrevistas a profesores a tiempo completo, coordinador y director de la Facultad de Ingeniería.

Se consideraron 3 aspectos fundamentales para la selección de la herramienta de geolocalización que fueron disponibilidad del servicio, documentación para desarrolladores y el costo, que sin duda fue lo más importante. Luego de evaluar estos aspectos se decidió por utilizar el api de Google por disponibilidad del servicio el cuál se encuentra 24/7, por documentación que hay tanto en la página oficial como en foros y conversatorios para desarrolladores y en costo que da un beneficio de consumo de hasta 200USD gratis al mes, lo cual permitió el desarrollo de un sistema más amplio y flexible para la comunicación interna en la UCSG.

Se hizo factible el desarrollo e implementación del aplicativo web que permite la creación y difusión de eventos con ubicación geográfica y que al ser tipo web y hosteado en la nube permite ser accedido desde cualquier dispositivo que cuente con un navegador web y una conexión a internet ya que el navegador funciona de forma independiente al sistema operativo.

Para la implementación de la herramienta se requirieron de 6 servicios de computación en la nube de forma fundamental que son: servicio de aplicaciones, servicio de base de datos, servicio de geolocalización, servicio de almacenamiento de archivos, servicio de envío de correos y servicio de tareas programadas.

Por medio de la utilización de estos servicios se pudieron realizar las notificaciones y las alertas de forma automática que funcionan como un web job que está censando la base de datos y verificando cuando un evento está por comenzar para generar el aviso por medio de un correo electrónico manteniendo informados a todos los involucrados en el desarrollo del evento.

Se pudo verificar el funcionamiento de la aplicación web a través de la ejecución de un plan de pruebas en las que se realizaron validaciones en las opciones de creación de usuario verificando que se validen tanto nombre de usuario como correo electrónico, se probó la pantalla de creación de eventos en la cual se presentó la novedad de que no todos los navegadores tienen activada la opción de ubicación, por lo que fue necesario agregar validaciones para mostrar la pantalla y que no se muestre como un error sino más bien solicitar activar esta característica, además también se validó la recepción de los correos electrónicos tanto por notificación como por alerta. Por último, se verificó la funcionalidad de la pantalla de estadísticas de asistencia y se constató que se mostraba la información solicitada.

RECOMENDACIONES

- Crear grupos de usuarios para que puedan publicar por ejemplo los docentes.
- Generación de reportes de asistentes a eventos.
- Que se implemente y se use la aplicación web.

BIBLIOGRAFÍA

- Alicante, U. d. (s.f.). *Servicio de Informática ASP.NET MVC 3 Framework*. Obtenido de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Alvarado, M. T. (2011). *Protocolo y organización de eventos*. Editorial UOC.
- Alvarez, M. (02 de Enero de 2016). *Desarrollo Web*. Obtenido de <https://desarrolloweb.com>: <https://desarrolloweb.com/articulos/392.php>
- Arias, Á. (2015). *Computación en la Nube: 2ª Edición*. IT Campus Academy.
- Arteaga, L. (12 de Junio de 2018). *Software libre*. Obtenido de Gnu: <https://www.gnu.org/philosophy/free-sw.es.html>
- Asamblea Nacional. (2008). *Asamblea Nacional*. Quito: Constitución del Ecuador.
- Bravo, L. (19 de 07 de 2013). *SCIELO*. Obtenido de <http://www.scielo.org.mx>: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-50572013000300009
- Camacho, E., Cardeso, F., & Nuñez, G. (2004). *Arquitectura de Software*. Obtenido de <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>
- Castellanos, R. (2008, p.10). *Enciclopedia de Informática Activa*. Grupo libro.
- Cruz, E. (25 de Septiembre de 2012). *Slideshare*. Obtenido de <https://es.slideshare.net/edwardcruz9638/ingenieria-web-14457826>
- Duarte, E. (04 de Marzo de 2015). *JavaScript*. Obtenido de Capacity Academy: <http://blog.capacityacademy.com/2012/10/19/que-es-javascript-ventajas-y-desventajas/>
- Eguiluz, J. (02 de Febrero de 2014). *Introducción a XHTML*. Obtenido de LibrosWeb: <https://librosweb.es/libro/xhtml/>
- ElectronicTerms. (2008). *IEEE Standard Dictionary of Electrical and Electronic Terms*. Grupo libro.
- Fernández, E. (2012). *Proceso de agregación para estudios experimentales en Ingeniería*.
- Google. (2018). Obtenido de <https://cloud.google.com/maps-platform/products/?hl=es>

- Gutierrez, D. (Julio de 2011). *Metodologías y procesos de análisis de software*.
Obtenido de <http://alexanderlemus.blogspot.mx/2011/12/documentos-que-se-entregan-en-el-modelo.html>
- Hernández, U. (25 de Octubre de 2015). *MVC*. Obtenido de Código Facilito:
<https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- Holguera, J. (22 de Noviembre de 2010). *SintaxisRazor*. Obtenido de Desarrollo Web:
<https://desarrolloweb.com/articulos/la-sintaxis-razor.html>
- IEPI. (05 de 10 de 2018). *Servicio Nacional de Derechos Intelectuales*. Obtenido de <https://www.propiedadintelectual.gob.ec>:
<https://www.propiedadintelectual.gob.ec/propiedad-intelectual/>
- Lerner, P. (2009). *La Arquitectura de la Información como estrategia comunicacional*. Colombia: Editorial CASEWISE.
- Levin, R., & Rubin, D. (2004). *Estadística para Administración y Economía*. Mexico: Person Education, Inc.
- Manager, S. (02 de Febrero de 2016). Obtenido de <https://scrummanager.com/index.php/es/?title=Artefactos>
- Microsoft. (2016). *ASP.NET SignalR*. Obtenido de <http://signalr.net/>
- Microsoft. (2017). *Microsoft*. Obtenido de <https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAAnLZL9>
- Microsoft. (2018). *SQL Server Service Broker*. Obtenido de <https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-service-broker?view=sql-server-2017>
- Morales, R. C. (1998). *Introducción Al Análisis de Sistemas Y la Ingeniería de Software*. EUNED.
- Net, P. (15 de Enero de 2012). Obtenido de <http://php.net/manual/es/book.mysql.php>
- Orosco, B. (16 de Abril de 2014). Obtenido de <http://bonillajessica.blogspot.com/>
- Pérez, D. (26 de Octubre de 2007). *Base de Datos* . Obtenido de <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Pérez, D. (02 de Noviembre de 2007). *Los diferentes lenguajes de programación para la web*. Obtenido de Maestros del Web:
<http://www.maestrosdelweb.com/los-diferentes-lenguajes-de-programacion-para-la-web/>

- Pérez, M. D. (2000). *Ingeniería del software y bases de datos*. Univ de Castilla La Mancha.
- Ponce, A. (16 de 11 de 2017). *Digital Book*. Obtenido de <http://reader.digitalbooks.pro>:
http://reader.digitalbooks.pro/book/preview/43053/id_ch_4
- Rouse, M. (Marzo de 2017). Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/Microsoft-Azure-Windows-Azure>
- Rouse, M. (29 de Noviembre de 2017). *SQL Server*. Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/SQL-Server>
- Sampieri, R., Fernández-Collado, C., & Pilar Baptista, L. (2006). *Metología de la Investigacion*. Mexico: McGramHill.
- Sanchez, C. (02 de Febrero de 2014). *Sitio Web*. Obtenido de <http://www.diferencia-entre.com>: <http://www.diferencia-entre.com/diferencia-entre-sitio-web-y-pagina-web/>
- SIG. (09 de 10 de 2017). *Sistemas de Informacion Geografica, tipos y Aplicaciones Informaticas*. Obtenido de <http://sig.cea.es>: <http://sig.cea.es/SIG>
- SmartyStreets. (2018). *SmartyStreets*. Retrieved from SmartyStreets:
<https://smartystreets.com/articles/is-there-a-geocoding-api-as-good-as-google>
- Soliz, A. M. (15 de Junio de 2009). *Blogspot*. Obtenido de <http://anamaigua.blogspot.com/2009/06/ingenieria-web.html>
- Suárez, R. (2008). *Programando en Java Script*. Editorial RitisGraff.
- Taranto, G., & Baquerizo, G. (05 de Marzo de 2009). *Diseño y elaboración de un sistema de información para el manejo de historias clínicas de pacientes con càncer en los hospitales de la ciudad de Guayaquil*. Obtenido de Dspace Espol:
<https://www.dspace.espol.edu.ec/bitstream/123456789/2101/1/4128.pdf>
- Tsai, A. (2007). *Sistemas de base de datos*. Editorial Prentice-Hall hispanoamerica.
- Wagner, B. (19 de Julio de 2015). *Lenguaje C#*. Obtenido de Microsoft:
<https://docs.microsoft.com/es-es/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>

ANEXOS

ANEXO A: SCRIPT DE CREACIÓN DE MODELO E-R DE LA BASE DE DATOS

Tablas

Entidades del modelo ER para la gestión de la aplicación.

```
/****** Object: Table [dbo].[Archivo]    Script Date: 25/02/2019 9:18:08
*****/
```

```
CREATE TABLE [dbo].[Archivo](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [CodigoEntidad] [nvarchar](128) NOT NULL,
    [NombreEntidad] [varchar](50) NOT NULL,
    [NombreArchivo] [varchar](100) NOT NULL,
    [RutaArchivo] [varchar](300) NOT NULL,
    [TipoContenido] [nvarchar](200) NOT NULL,
    CONSTRAINT [PK_Files] PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
/****** Object: Table [dbo].[AspNetRoles]    Script Date: 25/02/2019 9:18:09
*****/
```

```
CREATE TABLE [dbo].[AspNetRoles](
    [Id] [nvarchar](128) NOT NULL,
    [Name] [nvarchar](256) NOT NULL,
    CONSTRAINT [PK_dbo.AspNetRoles] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
/****** Object: Table [dbo].[AspNetUserClaims]    Script Date: 25/02/2019
9:18:09 *****/
```

```
CREATE TABLE [dbo].[AspNetUserClaims](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UserId] [nvarchar](128) NOT NULL,
    [ClaimType] [nvarchar](max) NULL,
    [ClaimValue] [nvarchar](max) NULL,
    CONSTRAINT [PK_dbo.AspNetUserClaims] PRIMARY KEY CLUSTERED
(
    [Id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```
***** Object: Table [dbo].[AspNetUserRoles] Script Date: 25/02/2019
9:18:09 *****/
```

```
CREATE TABLE [dbo].[AspNetUserRoles](
    [UserId] [nvarchar](128) NOT NULL,
    [RoleId] [nvarchar](128) NOT NULL,
    CONSTRAINT [PK_dbo.AspNetUserRoles] PRIMARY KEY CLUSTERED
(
    [UserId] ASC,
    [RoleId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
***** Object: Table [dbo].[AspNetUsers] Script Date: 25/02/2019 9:18:09
*****/
```

```
CREATE TABLE [dbo].[AspNetUsers](
    [Id] [nvarchar](128) NOT NULL,
    [Email] [nvarchar](256) NULL,
    [EmailConfirmed] [bit] NOT NULL,
    [PasswordHash] [nvarchar](max) NULL,
    [SecurityStamp] [nvarchar](max) NULL,
    [PhoneNumber] [nvarchar](max) NULL,
    [PhoneNumberConfirmed] [bit] NOT NULL,
    [TwoFactorEnabled] [bit] NOT NULL,
    [LockoutEndDateUtc] [datetime] NULL,
    [LockoutEnabled] [bit] NOT NULL,
    [AccessFailedCount] [int] NOT NULL,
    [UserName] [nvarchar](256) NOT NULL,
    CONSTRAINT [PK_dbo.AspNetUsers] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```
***** Object: Table [dbo].[Auditoria] Script Date: 25/02/2019 9:18:09
*****/
```

```
CREATE TABLE [dbo].[Auditoria](
    [IDAuditoria] [int] IDENTITY(1,1) NOT NULL,
    [IDEvento] [nvarchar](128) NULL,
    [ID] [int] NULL,
    [Entidad] [varchar](50) NULL,
    [Accion] [varchar](20) NULL,
    [Usuario] [varchar](50) NULL,
    [Fecha] [datetime] NULL,
    PRIMARY KEY CLUSTERED
(
    [IDAuditoria] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
***** Object: Table [dbo].[Comentario] Script Date: 25/02/2019 9:18:09
*****/
```

```
SET ANSI_NULLS ON
CREATE TABLE [dbo].[Comentario](
    [IDComentario] [int] IDENTITY(1,1) NOT NULL,
```

```

        [IDEvento] [nvarchar](128) NULL,
        [TextoComentario] [varchar](200) NOT NULL,
        [UserName] [varchar](50) NOT NULL,
        [FechaCreacion] [datetime] NOT NULL,
        [Activo] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
        [IDComentario] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

/***** Object: Table [dbo].[Entidad]    Script Date: 25/02/2019 9:18:09
*****/
CREATE TABLE [dbo].[Entidad](
        [IDEntidad] [varchar](2) NOT NULL,
        [Descripcion] [varchar](50) NOT NULL,
        [Activo] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
        [IDEntidad] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

/***** Object: Table [dbo].[Evento]    Script Date: 25/02/2019 9:18:09
*****/
CREATE TABLE [dbo].[Evento](
        [IDEvento] [nvarchar](128) NOT NULL,
        [Nombre] [varchar](50) NOT NULL,
        [Lugar] [varchar](100) NULL,
        [Descripcion] [varchar](300) NOT NULL,
        [Asistentes] [int] NULL,
        [Interesados] [int] NULL,
        [FechaInicio] [datetime] NOT NULL,
        [FechaFin] [datetime] NOT NULL,
        [Imagen] [varchar](300) NOT NULL,
        [Lat] [varchar](100) NOT NULL,
        [Long] [varchar](100) NOT NULL,
        [Estado] [char](1) NOT NULL,
        [UserName] [varchar](50) NULL,
        [FechaCreacion] [datetime] NULL,
        [FechaModificacion] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
        [IDEvento] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

/***** Object: Table [dbo].[EventoAlertaUsuario]    Script Date: 25/02/2019
9:18:09 *****/
CREATE TABLE [dbo].[EventoAlertaUsuario](
        [IDEvento] [nvarchar](128) NOT NULL,
        [IDUser] [nvarchar](128) NOT NULL,
        [FechaCreacion] [datetime] NULL,
        CONSTRAINT [pk_EventoAlertaUsuario] PRIMARY KEY CLUSTERED
(
        [IDEvento] ASC,
        [IDUser] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
***** Object: Table [dbo].[EventoEntidad] Script Date: 25/02/2019
9:18:09 *****/
```

```
CREATE TABLE [dbo].[EventoEntidad](
    [IDEventoEntidad] [int] IDENTITY(1,1) NOT NULL,
    [IDEvento] [nvarchar](128) NULL,
    [IDEntidad] [varchar](2) NULL,
    [UserName] [varchar](50) NOT NULL,
    [FechaCreacion] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
    [IDEventoEntidad] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
***** Object: Table [dbo].[EventoInvitado] Script Date: 25/02/2019
9:18:09 *****/
```

```
CREATE TABLE [dbo].[EventoInvitado](
    [IDEvento] [nvarchar](128) NOT NULL,
    [IDUser] [nvarchar](128) NOT NULL,
    [Cont] [int] NULL,
    [Activo] [bit] NULL,
    [UsuarioCreacion] [varchar](50) NULL,
    [FechaCreacion] [datetime] NULL,
    [UsuarioModificacion] [varchar](50) NULL,
    [FechaModificacion] [datetime] NULL,
CONSTRAINT [pk_EventoInvitado] PRIMARY KEY CLUSTERED
(
    [IDEvento] ASC,
    [IDUser] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
***** Object: Table [dbo].[EventoTipoDetalle] Script Date: 25/02/2019
9:18:09 *****/
```

```
CREATE TABLE [dbo].[EventoTipoDetalle](
    [IDEventoTipoDetalle] [int] IDENTITY(1,1) NOT NULL,
    [IDEvento] [nvarchar](128) NULL,
    [IDTipoEvento] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [IDEventoTipoDetalle] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [uq_EventoTipoDetalle] UNIQUE NONCLUSTERED
(
    [IDEventoTipoDetalle] ASC,
    [IDTipoEvento] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
***** Object: Table [dbo].[Menu] Script Date: 25/02/2019 9:18:09 *****/
CREATE TABLE [dbo].[Menu](
```

```

        [ID] [int] IDENTITY(1,1) NOT NULL,
        [Nombre] [varchar](100) NOT NULL,
        [Icono] [varchar](100) NULL,
PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
    CONSTRAINT [UC_Menu] UNIQUE NONCLUSTERED
(
    [Nombre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

/***** Object: Table [dbo].[Parametro]    Script Date: 25/02/2019 9:18:09
*****/

```

```

CREATE TABLE [dbo].[Parametro](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Codigo] [varchar](20) NULL,
    [Descripcion] [varchar](100) NULL,
    [ValorTexto] [varchar](500) NULL,
    [ValorEntero] [int] NULL,
    [ValorDecimal] [decimal](18, 0) NULL,
PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

/***** Object: Table [dbo].[PermisoRol]    Script Date: 25/02/2019 9:18:09
*****/

```

```

CREATE TABLE [dbo].[PermisoRol](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [RolID] [nvarchar](128) NOT NULL,
    [SubMenuID] [int] NOT NULL,
    [Controlador] [varchar](100) NULL,
    [Accion] [varchar](100) NULL,
    [Opcion] [varchar](100) NULL,
    [Visible] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
    [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

/***** Object: Table [dbo].[SubMenu]    Script Date: 25/02/2019 9:18:09
*****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SubMenu](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [MenuID] [int] NOT NULL,
    [Nombre] [varchar](100) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [ID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [UC_SubMenu] UNIQUE NONCLUSTERED
(
[MenuID] ASC,
[Nombre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
/****** Object: Table [dbo].[TipoEvento] Script Date: 25/02/2019 9:18:09
*****/
```

```
CREATE TABLE [dbo].[TipoEvento](
[IDTipoEvento] [int] IDENTITY(1,1) NOT NULL,
[Nombre] [varchar](100) NOT NULL,
[Icono] [varchar](30) NULL,
[Color] [varchar](30) NULL,
[Activo] [bit] NOT NULL,
PRIMARY KEY CLUSTERED
(
[IDTipoEvento] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
UNIQUE NONCLUSTERED
(
[Nombre] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[AspNetUserClaims] WITH CHECK ADD CONSTRAINT
[FK_dbo.AspNetUserClaims_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserClaims] CHECK CONSTRAINT
[FK_dbo.AspNetUserClaims_dbo.AspNetUsers_UserId]
GO
ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetRoles_RoleId] FOREIGN KEY([RoleId])
REFERENCES [dbo].[AspNetRoles] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetRoles_RoleId]
GO
ALTER TABLE [dbo].[AspNetUserRoles] WITH CHECK ADD CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetUsers_UserId] FOREIGN KEY([UserId])
REFERENCES [dbo].[AspNetUsers] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_dbo.AspNetUserRoles_dbo.AspNetUsers_UserId]
GO
ALTER TABLE [dbo].[Comentario] WITH CHECK ADD FOREIGN KEY([IDEvento])
REFERENCES [dbo].[Evento] ([IDEvento])
GO
ALTER TABLE [dbo].[EventoAlertaUsuario] WITH CHECK ADD FOREIGN
KEY([IDEvento])
REFERENCES [dbo].[Evento] ([IDEvento])
GO
```

```

ALTER TABLE [dbo].[EventoAlertaUsuario] WITH CHECK ADD FOREIGN KEY([IDUser])
REFERENCES [dbo].[AspNetUsers] ([Id])
GO
ALTER TABLE [dbo].[EventoEntidad] WITH CHECK ADD FOREIGN KEY([IDEntidad])
REFERENCES [dbo].[Entidad] ([IDEntidad])
GO
ALTER TABLE [dbo].[EventoEntidad] WITH CHECK ADD FOREIGN KEY([IDEvento])
REFERENCES [dbo].[Evento] ([IDEvento])
GO
ALTER TABLE [dbo].[EventoInvitado] WITH CHECK ADD FOREIGN KEY([IDEvento])
REFERENCES [dbo].[Evento] ([IDEvento])
GO
ALTER TABLE [dbo].[EventoInvitado] WITH CHECK ADD FOREIGN KEY([IDUser])
REFERENCES [dbo].[AspNetUsers] ([Id])
GO
ALTER TABLE [dbo].[EventoTipoDetalle] WITH CHECK ADD FOREIGN KEY([IDEvento])
REFERENCES [dbo].[Evento] ([IDEvento])
GO
ALTER TABLE [dbo].[EventoTipoDetalle] WITH CHECK ADD FOREIGN
KEY([IDTipoEvento])
REFERENCES [dbo].[TipoEvento] ([IDTipoEvento])

```

Triggers

Escriben en la tabla Auditoria al realizar operaciones CUD (Create, Update, Delete) sobre las tablas Comentario, Evento y EventoEntidad. La tabla Auditoria es leída constantemente por el SQL Dependency para verificar si existen cambios y mostrarlos en tiempo real del lado del cliente.

```

create trigger [dbo].[Comentario_trigger]
on [dbo].[Comentario]
after UPDATE, INSERT, DELETE
as
declare @IDEvento [nvarchar](128), @user varchar(20), @activity varchar(20),
@fecha datetime
if exists(SELECT * from inserted) and exists (SELECT * from deleted)
begin
    SET @activity = 'UPDATE';
    SELECT @IDEvento = i.IDEvento, @user = i.UserName, @fecha = getdate() from
inserted i;
    INSERT into Auditoria(IDEvento, ID, Entidad, Accion, Usuario, Fecha) values
(isnull(@IDEvento, ''), 0, 'Comentario', isnull(@activity, ''), isnull(@user, ''), is
null(@fecha, getdate()));
end

If exists (Select * from inserted) and not exists(Select * from deleted)
begin
    SET @activity = 'INSERT';
    SELECT @IDEvento = i.IDEvento, @user = i.UserName, @fecha =
i.FechaCreacion from inserted i;
    INSERT into Auditoria(IDEvento, ID, Entidad, Accion, Usuario, Fecha)
values(isnull(@IDEvento, ''), 0, 'Comentario', isnull(@activity, ''), isnull(@user, '
'), isnull(@fecha, getdate()));
end

```



```

If exists(select * from deleted) and not exists(Select * from inserted)
begin
    SET @activity = 'DELETE';
    SELECT @IDEvento = i.IDEvento, @user = i.UserName, @fecha = getdate() from
deleted i;
    INSERT into Auditoria(IDEvento, ID, Entidad, Accion, Usuario, Fecha)
values(isnull(@IDEvento, ''), 0, 'Comentario', isnull(@activity, ''), isnull(@user, '
'), isnull(@fecha, getdate()));
end
GO
ALTER TABLE [dbo].[Comentario] ENABLE TRIGGER [Comentario_trigger]
GO
/***** Object: Trigger [dbo].[Evento_trigger]    Script Date: 25/02/2019
9:18:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

create trigger [dbo].[Evento_trigger]
on [dbo].[Evento]
after UPDATE, INSERT, DELETE
as
declare @IDEvento [nvarchar](128), @user varchar(20), @activity varchar(20),
@fecha datetime
if exists(SELECT * from inserted) and exists (SELECT * from deleted)
begin
    SET @activity = 'UPDATE';
    SELECT @IDEvento = i.IDEvento, @user = i.UserName, @fecha =
i.FechaModificacion from inserted i;
    INSERT into Auditoria(IDEvento, ID, Entidad, Accion, Usuario, Fecha) values
(isnull(@IDEvento, ''), 0, 'Evento', isnull(@activity, ''), isnull(@user, ''), isnull(
@fecha, getdate()));
end

If exists (Select * from inserted) and not exists(Select * from deleted)
begin
    SET @activity = 'INSERT';
    SELECT @IDEvento = i.IDEvento, @user = i.UserName, @fecha =
i.FechaCreacion from inserted i;
    INSERT into Auditoria(IDEvento, ID, Entidad, Accion, Usuario, Fecha)
values(isnull(@IDEvento, ''), 0, 'Evento', isnull(@activity, ''), isnull(@user, ''), i
snull(@fecha, getdate()));
end

If exists(select * from deleted) and not exists(Select * from inserted)
begin
    SET @activity = 'DELETE';
    SELECT @IDEvento = i.IDEvento, @user = i.UserName, @fecha = getdate() from
deleted i;
    INSERT into Auditoria(IDEvento, ID, Entidad, Accion, Usuario, Fecha)
values(isnull(@IDEvento, ''), 0, 'Evento', isnull(@activity, ''), isnull(@user, ''), i
snull(@fecha, getdate()));
end
GO
ALTER TABLE [dbo].[Evento] ENABLE TRIGGER [Evento_trigger]
GO
/***** Object: Trigger [dbo].[EventoEntidad_trigger]    Script Date:
25/02/2019 9:18:11 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

create trigger [dbo].[EventoEntidad_trigger]
on [dbo].[EventoEntidad]
after UPDATE, INSERT, DELETE
as
declare @id int, @IDEvento [nvarchar](128), @IDEntidad varchar(2), @user
varchar(20), @activity varchar(20), @fecha datetime
if exists(SELECT * from inserted) and exists (SELECT * from deleted)
begin
    SET @activity = 'UPDATE';
    SELECT @id = IDEventoEntidad, @IDEvento = i.IDEvento, @user = i.UserName,
@fecha = getdate(), @IDEntidad = i.IDEntidad from inserted i;
    INSERT into Auditoria(IDEvento,ID,Entidad,Accion,Usuario,Fecha) values
(isnull(@IDEvento, ''),isnull(@id,0),isnull(@IDEntidad, ''),isnull(@activity, ''))
,isnull(@user, ''),isnull(@fecha,getdate()));
end

If exists (Select * from inserted) and not exists(Select * from deleted)
begin
    SET @activity = 'INSERT';
    SELECT @id = IDEventoEntidad, @IDEvento = i.IDEvento, @user = i.UserName,
@fecha = i.FechaCreacion, @IDEntidad = i.IDEntidad from inserted i;
    INSERT into Auditoria(IDEvento,ID,Entidad,Accion,Usuario,Fecha)
values(isnull(@IDEvento, ''),isnull(@id,0),isnull(@IDEntidad, ''),isnull(@activi
ty, ''),isnull(@user, ''),isnull(@fecha,getdate()));
end

If exists(select * from deleted) and not exists(Select * from inserted)
begin
    SET @activity = 'DELETE';
    SELECT @id = IDEventoEntidad, @IDEvento = i.IDEvento, @user = i.UserName,
@fecha = getdate(), @IDEntidad = i.IDEntidad from deleted i;
    INSERT into Auditoria(IDEvento,ID,Entidad,Accion,Usuario,Fecha)
values(isnull(@IDEvento, ''),isnull(@id,0),isnull(@IDEntidad, ''),isnull(@activi
ty, ''),isnull(@user, ''),isnull(@fecha,getdate()));
end
GO
ALTER TABLE [dbo].[EventoEntidad] ENABLE TRIGGER [EventoEntidad_trigger]
GO

```

Funciones

Tienen como objetivo recibir parámetros y devolver datos en otro formato dependiendo de las necesidades que se presentaron en el desarrollo de la aplicación.

```

/***** Object: UserDefinedFunction [dbo].[Comenzado]    Script Date:
25/02/2019 9:26:01 *****/
/***** Devuelve verdadero o falso si un evento ya ha comenzado *****/
CREATE FUNCTION [dbo].[Comenzado]
    (@FechaActual datetime, @FechaInicio datetime)
RETURNS bit

```

```

AS
BEGIN
    DECLARE @result bit = 0;;
    set @result = convert(bit, case when
DATEDIFF(MINUTE,@FechaActual,@FechaInicio) > 0 then 0 else 1 end )

    return @result;
END

/***** Object: UserDefinedFunction [dbo].[Descripcion]    Script Date:
25/02/2019 9:26:01 *****/
/***** Obtiene una descripción de texto sobre el inicio o fin del evento, si
está por comenzar o si está por terminar *****/

CREATE FUNCTION [dbo].[Descripcion]
    (@FechaActual datetime, @FechaInicio datetime,@FechaFin datetime )
RETURNS varchar(100)
AS
BEGIN
    DECLARE @result varchar(100) = '';
    declare @Prioridad int = 0;
    declare @DiaInicia int = 0;
    declare @DiaTermina int = 0;
    declare @HoraInicia int = 0;
    declare @HoraTermina int = 0;
    declare @MinutoInicia int = 0;
    declare @MinutoTermina int = 0;

    set @Prioridad = dbo.Prioridad(@FechaActual,@FechaInicio,@FechaFin);

    set @DiaInicia = DATEDIFF(DAY,@FechaActual,@FechaInicio) ;
    set @DiaTermina = DATEDIFF(DAY,@FechaActual,@FechaFin);

    set @HoraInicia = DATEDIFF(HOUR,@FechaActual,@FechaInicio) ;
    set @HoraTermina = DATEDIFF(HOUR,@FechaActual,@FechaFin);

    set @MinutoInicia = DATEDIFF(MINUTE,@FechaActual,@FechaInicio) ;
    set @MinutoTermina = DATEDIFF(MINUTE,@FechaActual,@FechaFin);

    if (@prioridad = 1)
    begin

        if (@DiaTermina)>0
            SET @result = 'Termina en ' +
convert(varchar(10),@DiaTermina) + ' día(s)';
            else if (@HoraTermina>0)
                SET @result = 'Termina en ' +
convert(varchar(10),@HoraTermina) + ' hora(s)';
            else
                SET @result = 'Termina en ' +
convert(varchar(10),@MinutoTermina) + ' minuto(s)';

        end
    else if (@prioridad = 2)
    begin

        if (@DiaInicia)>0

```

```

        SET @result = 'Comienza en ' +
convert(varchar(10),@DiaInicia) + ' día(s)';
        else if (@HoraInicia>0)
        SET @result = 'Comienza en ' +
convert(varchar(10),@HoraInicia) + ' hora(s)';
        else
        SET @result = 'Comienza en ' +
convert(varchar(10),@MinutoInicia) + ' minuto(s)';

    end
    else if (@prioridad = 3)
    begin

        if (abs(@DiaTermina))>0
        SET @result = 'Finalizó hace ' +
convert(varchar(10),abs(@DiaTermina)) + ' día(s)';
        else if (abs(@HoraTermina)>0)
        SET @result = 'Finalizó hace ' +
convert(varchar(10),abs(@HoraTermina)) + ' hora(s)';
        else
        SET @result = 'Finalizó hace ' +
convert(varchar(10),abs(@MinutoTermina)) + ' minuto(s)';

    end

    return @result;
END

```

```

/***** Object: UserDefinedFunction [dbo].[FechaAuditoria]    Script Date:
25/02/2019 9:26:01 *****/
/***** Obtiene la fecha de auditoria en formato texto *****/

```

```

CREATE FUNCTION [dbo].[FechaAuditoria]
    (@Fecha datetime)
RETURNS varchar(100)
AS
BEGIN
    DECLARE @result varchar(100) = '';

    set @result = convert(varchar(2), day(@Fecha)) + ' de ' +
LOWER(dbo.[MesChar](month(@Fecha))) + '. de ' +
convert(varchar(4),year(@Fecha)) +
        N' a las ' + RIGHT('00'+
convert(varchar(2),DATEPART(HOUR,@Fecha)),2) + ':' + RIGHT('00'+
convert(varchar(2),DATEPART(MINUTE,@Fecha)),2);

    set @result = isnull(@result, '');

    return @result;
END

```

```

/***** Object: UserDefinedFunction [dbo].[FechaHoraTexto]    Script Date:
25/02/2019 9:26:01 *****/
/***** Recibe como parámetro una fecha y la devuelve la fecha y hora en
formato texto *****/

```

```

create FUNCTION [dbo].[FechaHoraTexto]
    (@Fecha datetime)
RETURNS varchar(100)
AS

```

```

BEGIN
    DECLARE @result varchar(100) = '';

    set @result = convert(varchar(2), day(@Fecha)) + ' de ' +
LOWER(dbo.[MesChar](month(@Fecha))) + '. de ' +
convert(varchar(4),year(@Fecha)) +
        N' a las ' + RIGHT('00'+
convert(varchar(2),DATEPART(HOUR,@Fecha)) ,2) + ':' + RIGHT('00'+
convert(varchar(2),DATEPART(MINUTE,@Fecha)),2);

    set @result = isnull(@result, '');

    return @result;
END

/***** Object: UserDefinedFunction [dbo].[FechaTexto]    Script Date:
25/02/2019 9:26:01 *****/
/***** Recibe como parámetro una fecha y la devuelve la fecha en formato
texto *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create FUNCTION [dbo].[FechaTexto]
    (@Fecha datetime)
RETURNS varchar(100)
AS
BEGIN
    DECLARE @result varchar(100) = '';

    set @result = upper(SUBSTRING( FORMAT(@Fecha, 'dddd', 'es-es') , 1,3))
+ '., ' +
        RIGHT('00'+ convert(varchar(2), day(@Fecha)),2) + ' ' +
        dbo.[MesChar](month(@Fecha)) + ' ';

    set @result = isnull(@result, '');

    return @result;
END

```

```

/***** Object: UserDefinedFunction [dbo].[HoraTexto]    Script Date:
25/02/2019 9:26:01 *****/
/***** Recibe como parámetro una fecha y la devuelve la hora en formato texto
*****/

```

```

create FUNCTION [dbo].[HoraTexto]
    (@Fecha datetime)
RETURNS varchar(100)
AS
BEGIN
    DECLARE @result varchar(100) = '';

    set @result = RIGHT('00'+ convert(varchar(2),DATEPART(HOUR,@Fecha)),2)
+ ':' + RIGHT('00'+ convert(varchar(2),DATEPART(MINUTE,@Fecha)) ,2);
    set @result = isnull(@result, '');

    return @result;
END

```

```

/***** Object: UserDefinedFunction [dbo].[MesChar]    Script Date:
25/02/2019 9:26:01 *****/
/***** Recibe como parámetro entero y devuelve el mes en formato texto
*****/

```

```

CREATE FUNCTION [dbo].[MesChar]
    (@Mes int )
RETURNS varchar(100)
AS
BEGIN
    DECLARE @result varchar(100) = '';

    if @Mes = 1
        set @result = 'ENE'
    else if @Mes = 2
        set @result = 'FEB'
    else if @Mes = 3
        set @result = 'MAR'
    else if @Mes = 4
        set @result = 'ABR'
    else if @Mes = 5
        set @result = 'MAY'
    else if @Mes = 6
        set @result = 'JUN'
    else if @Mes = 7
        set @result = 'JUL'
    else if @Mes = 8
        set @result = 'AGO'
    else if @Mes = 9
        set @result = 'SEP'
    else if @Mes = 10
        set @result = 'OCT'
    else if @Mes = 11
        set @result = 'NOV'
    else if @Mes = 12
        set @result = 'DIC'

    return @result;
END

```

```

/***** Object: UserDefinedFunction [dbo].[Prioridad]    Script Date:
25/02/2019 9:26:01 *****/
/***** Devuelve un entero con la prioridad para mostrar el evento de acuerdo
a la fecha inicio y fecha de fin *****/

```

```

CREATE FUNCTION [dbo].[Prioridad]
    (@FechaActual datetime, @FechaInicio datetime, @FechaFin datetime)
RETURNS int
AS
BEGIN
    DECLARE @result int = 0;
    set @result = case when dbo.Comenzado(@FechaActual,@FechaInicio) = 1
and dbo.Terminado(@FechaActual,@FechaFin) = 0 then 1
                    when
dbo.Comenzado(@FechaActual,@FechaInicio) = 0 and
dbo.Terminado(@FechaActual,@FechaFin) = 0 then 2

```

```

                                when
dbo.Comenzado(@FechaActual,@FechaInicio) = 1 and
dbo.Terminado(@FechaActual,@FechaFin) = 1 then 3 end

        return @result;
END

/***** Object: UserDefinedFunction [dbo].[Terminado]    Script Date:
25/02/2019 9:26:01 *****/

/***** Devuelve verdadero o falso si el evento ya ha finalizado *****/

CREATE FUNCTION [dbo].[Terminado]
    (@FechaActual datetime, @FechaFin datetime)
RETURNS bit
AS
BEGIN
    DECLARE @result bit = 0;;
    set @result = convert( bit, Case when
DATEDIFF(MINUTE,@FechaActual,@FechaFin) > 0 then 0 else 1 end )

        return @result;
END

/***** Object: UserDefinedFunction [dbo].[UserProfileImage]    Script Date:
25/02/2019 9:26:01 *****/
/***** Devuelve la URL de la imagen del evento, si no posee devuelve una ruta
por defecto *****/

CREATE FUNCTION [dbo].[UserProfileImage]
    (@UserName varchar(50) )
RETURNS varchar(100)
AS
BEGIN
    DECLARE @result varchar(200) = '';

    Select @result = RutaArchivo
    from Archivo
    where NombreEntidad = 'FotoPerfil'
        and CodigoEntidad = @UserName;

    if isnull(@result, '') = ''
        set @result = '../Content/img/user_new.jpg';

    return @result;
END

```

Procedimientos Almacenados

Procesos que contienen lógica de la aplicación que son utilizados desde las diversas acciones de los controladores.

```

/***** Object: StoredProcedure [dbo].[AlertaEvento]    Script Date:
25/02/2019 9:35:13 *****/
/*Genera la estructura para el envio del correo de alerta*/

CREATE PROCEDURE [dbo].[AlertaEvento]
    @Fecha varchar(14) = ''
as

if @Fecha = '' return;

Declare @FechaActual as datetime;
Declare @html as varchar(max) = '';
declare @FechaInicio varchar(200) = '';
declare @Nombre varchar(50) = '';
declare @Descripcion varchar(300) = '';
declare @LocationURL varchar(300) = '';
declare @ImagenURL varchar(300) = '';
declare @Lugar varchar(100) = '';
declare @Invitados varchar(max) = '';
declare @SENDGRID_APIKEY varchar(200) = '';
declare @ADMINNAME varchar(150) = '';
declare @ADMINMAIL varchar(150) = '';
declare @IDEvento [nvarchar](128) = '';
declare @ALMININI int = 0;

Select @ALMININI = ValorEntero
from Parametro WITH (NOLOCK)
where Codigo = 'ALMININI';

Select @FechaActual = convert(datetime,stuff(stuff(stuff(@Fecha, 9, 0, ' '),
12, 0, ':'),15, 0, ':'));

create table #tmpAlerta
(
    IDEvento [nvarchar](128)
    ,De varchar(200)
    ,Nombre varchar(150)
    ,Asunto varchar(50)
    ,Contenido varchar(max)
    ,ApiKey varchar(200)
);

create table #tmpDestinatario
(
    IDEvento [nvarchar](128)
    ,IDUser [nvarchar](128)
    ,Email varchar(150)
    ,UserName varchar(200)
);

select IDEvento , FechaInicio, FechaCreacion, FechaModificacion
into #tmp
from Evento WITH (NOLOCK)
where FechaInicio between @FechaActual and
dateadd(minute,@ALMININI,@FechaActual)
and Estado = 'G';

insert into #tmpDestinatario
Select distinct a.*
from ( Select a.IDEvento IDEvento, c.Id as IDUser, c.Email, c.UserName

```



```

        from #tmp a
        join EventoEntidad b WITH (NOLOCK) on a.IDEvento = b.IDEvento
        join aspnetusers c WITH (NOLOCK) on b.UserName = c.UserName
        union all
        Select a.IDEvento, c.Id, c.Email, c.UserName
        from #tmp a
        join EventoInvitado b WITH (NOLOCK) on a.IDEvento =
b.IDEvento
        join aspnetusers c WITH (NOLOCK) on b.IDUser = c.Id ) as a
left join EventoAlertaUsuario b on a.IDEvento = b.IDEvento
                                and a.IDUser = b.IDUser
where b.IDEvento is null;

```

```

Select @SENDGRID_APIKEY = ValorTexto
from Parametro WITH (NOLOCK)
where Codigo = 'SENDGRID_APIKEY';

```

```

Select @ADMINNAME = ValorTexto
from Parametro WITH (NOLOCK)
where Codigo = 'ADMINNAME';

```

```

while exists (select 1 from #tmp)
begin
    select top 1 @IDEvento = IDEvento
    from #tmp ;

```

```

        Select @Nombre = 'Recordatorio de ' + Nombre
        ,@FechaInicio = CASE WHEN
CONVERT(date, FechaInicio)=CONVERT(date, @FechaActual) then 'HOY'
                                WHEN
CONVERT(date, FechaInicio)=CONVERT(date, DATEADD(DAY, 1, @FechaActual)) then
'MAÑANA'
                                else
dbo.[FechaTexto](FechaInicio) end + '.', a las ' + dbo.[HoraTexto](a.FechaInicio)
        ,@Descripcion = Descripcion
        ,@Lugar = Lugar
        ,@LocationURL =
'https://www.google.com/maps/search/?api=1&query=' + Lat + ',' + Long + ''
        ,@ImagenURL = Imagen
        ,@ADMINMAIL = b.Email
        from Evento a WITH (NOLOCK)
        left join aspnetUsers b WITH (NOLOCK) on a.UserName = b.UserName
        where IDEvento = @IDEvento;

```

```

        set @html = '<head></head><body><h1 style="color: #5e9ca0;">' + @Nombre
+ '</h1>
                                <h2 style="color: #5e9ca0;">El evento está por
comenzar! </h2>
                                <h2 style="color: #2e6c80;"><span style="color:
#ff0000;">' + @FechaInicio + '</span></h2>
                                <p>' + @Descripcion + '&nbsp;</p>
                                <p>El evento se llevará a cabo en <span
style="background-color: #008000; color: #ffffff; display: inline-block;
padding: 3px 10px; font-weight: bold; border-radius: 5px;"><a href=""
+ @LocationURL + '' style="color:white" >' +
@Lugar + '</a></span></p>
                                <p>&nbsp;</p>

```

```

        <p><br /><strong>Te esperamos!</strong></p>';

        if @ImagenURL<>'
        begin
            set @html = @html + '<table><tr><td
valign="top" align="left"></td></tr>
                                <tr><td valign="top"
align="left">
                                <img src="" + @ImagenURL + "" width="800" height="500" border="0"
alt="">
                                </td></tr>
                                </table>';

            end;

            set @html = @html +
'<p><strong>&nbsp;</strong></p></body>';

        insert into #tmpAlerta
        select @IDEvento AS IDEvento
            ,@ADMINMAIL as De
            ,@ADMINNAME as Nombre
            ,@Nombre as Asunto
            ,@html as Contenido
            ,@SENDGRID_APIKEY as ApiKey;

        delete from #tmp where IDEvento = @IDEvento;
end

select * from #tmpAlerta;
select * from #tmpDestinatario;

```

```

/***** Object: StoredProcedure [dbo].[Dashboard]    Script Date: 25/02/2019
9:35:13 *****/
/*Procedimiento que devuelve la bandeja principal en base a la fecha actual y
el usuario*/

```

```

CREATE PROCEDURE [dbo].[Dashboard]
    @Fecha varchar(14) = ''
    ,@Usuario varchar(50) = ''
as

Declare @EVMIN int=30;

select @EVMIN = ValorEntero
from Parametro
where Codigo = 'EVMIN';

Declare @FechaActual as datetime;

if isnull(@Fecha, '') = ''
    select @FechaActual = getdate();
else
    select @FechaActual = convert(datetime,stuff(stuff(stuff(@Fecha, 9, 0,
' '), 12, 0, ':'),15, 0, ':'));

/*obtengo los eventos activos*/
select ROW_NUMBER() OVER(ORDER BY a.IDEvento ASC) AS ID
    ,a.IDEvento as IDEvento

```

```

, a.Nombre
, a.Lugar
, a.Descripcion
, '' as Organizador
, convert(FLOAT, a.Lat) as Lat
, convert(FLOAT, a.Long) as Long
, a.Lat as MarkerLat
, a.Long as MarkerLong
, 0 as IDTipoEvento
, convert(varchar(max), '') as TipoEventoNombre
, CASE WHEN
CONVERT(date, a.FechaInicio)=CONVERT(date, @FechaActual) then 'HOY'
      WHEN
CONVERT(date, a.FechaInicio)=CONVERT(date, DATEADD(DAY, 1, @FechaActual)) then
'MAÑANA'
      else dbo.[FechaTexto](a.FechaInicio) end as Inicio
, dbo.[HoraTexto](a.FechaInicio) as HoraIni
, dbo.[FechaTexto](a.FechaFin) as Fin
, dbo.[HoraTexto](a.FechaFin) as HoraFin
, 0 as Seguidores
, a.FechaInicio
, a.FechaFin
, dbo.Comenzado(@FechaActual, a.FechaInicio) as Comenzado
, dbo.Terminado(@FechaActual, a.FechaFin) as Terminado
, dbo.Prioridad(@FechaActual, a.FechaInicio, a.FechaFin) as
Prioridad
, dbo.Descripcion(@FechaActual, a.FechaInicio, a.FechaFin) as
Tiempo
, a.UserName
, case when a.FechaCreacion = a.FechaModificacion then
      dbo.[FechaHoraTexto](a.FechaCreacion) else
dbo.[FechaHoraTexto](a.FechaModificacion) + ' (Modificado)' end as Auditoria
, a.FechaModificacion as UltimaFecha
, convert(varchar(300), '') UserProfileImage
, a.Imagen as ImagenEvento
, convert(int, 0) as Comentarios
, convert(bit, 0) as Interesado
, convert(bit, 0) as Asistente
, convert(bit, case when @Usuario = a.UserName then 1 else 0 end)
as PermiteEliminar
, convert(varchar(50), '0 Comentario(s)') as TotalComentarios
, convert(varchar(50), '0 Interesado(s)') as TotalInteresados
, convert(varchar(50), '0 Asistente(s)') as TotalAsistentes
into #tmpEvento
from Evento as a WITH (NOLOCK)
where DATEADD(MINUTE, @EVMIN, a.FechaFin) >= @FechaActual
and a.Estado <> 'N';

/*actualizo campo de foto de perfil, si es nulo seteo la imagen para usuarios
por defecto*/
Update a
set a.UserProfileImage =
isnull(RutaArchivo, '../Content/img/user_new.jpg')
from #tmpEvento as a WITH (NOLOCK)
left join Archivo as b WITH (NOLOCK) on a.UserName = b.CodigoEntidad
and b.NombreEntidad = 'FotoPerfil';

/*actualizo el tipo de evento, se concatena los tipos de eventos
seleccionados*/
Update a
set a.TipoEventoNombre = b.TipoEventoNombre

```

```

from #tmpEvento a WITH (NOLOCK)
join (SELECT IDEvento = o.IDEvento,
      TipoEventoNombre = STUFF(
        (SELECT N' #' + d.Nombre
         FROM EventoTipoDetalle AS c
         join TipoEvento d WITH
         (NOLOCK) on c.IDTipoEvento = d.IDTipoEvento
         WHERE c.IDEvento =
         o.IDEvento
         FOR XML PATH, TYPE
        ).value(N'.[1]',N'nvarchar(max)'),1,1,N''
      FROM #tmpEvento AS o WITH (NOLOCK) ) as b on a.IDEvento =
b.IDEvento;

/*busco los comentarios*/
Select a.IDComentario
      ,a.IDEvento
      ,TextoComentario
      ,a.UserName
      ,dbo.[FechaHoraTexto](FechaCreacion) as FechaCreacion
      ,dbo.[UserProfileImage](a.UserName) as UserProfileImage
      ,convert(bit,case when @Usuario = a.UserName then 1 else 0 end)
as PermiteEliminar
into #tmpComentario
from Comentario as a WITH (NOLOCK)
join #tmpEvento as b WITH (NOLOCK) on a.IDEvento = b.IDEvento
Where a.Activo = 1;

/*actualizo la cantidad de comentarios*/
Update a
set a.Comentarios = b.Comentarios
    ,a.TotalComentarios = convert(varchar(10),b.Comentarios) + '
Comentario(s)'
from #tmpEvento a
join ( Select a.IDEvento as IDEvento , count(b.IDEvento) as Comentarios
      from #tmpEvento as a WITH (NOLOCK)
      join #tmpComentario as b WITH (NOLOCK) on a.IDEvento =
b.IDEvento
      Group by a.IDEvento ) as b on a.IDEvento = b.IDEvento;

/*obtengo la informacion de las entidades, me interesa y asistire*/
Select a.IDEvento
      ,sum(TotalInteresados) as TotalInteresados
      ,sum(TotalAsistentes) as TotalAsistentes
      ,sum(TotalOtrosInteresados) as TotalOtrosInteresados
      ,sum(TotalOtrosAsistentes) as TotalOtrosAsistentes
      ,sum(Interesado) as Interesado
      ,sum(Asistente) as Asistente
into #tmpEntidad
from (select a.IDEvento
      ,case when b.IDEntidad = 'mi' then 1 else 0 end as
TotalInteresados
      ,case when b.IDEntidad = 'as' then 1 else 0 end as
TotalAsistentes
      ,case when b.IDEntidad = 'mi' and b.UserName <>
@Usuario then 1 else 0 end as TotalOtrosInteresados

```

```

        ,case when b.IDEntidad = 'as' and b.UserName <>
@Usuario then 1 else 0 end as TotalOtrosAsistentes
        ,case when b.IDEntidad = 'mi' and b.UserName =
@Usuario then 1 else 0 end as Interesado
        ,case when b.IDEntidad = 'as' and b.UserName =
@Usuario then 1 else 0 end as Asistente
    from #tmpEvento as a WITH (NOLOCK)
    join EventoEntidad as b WITH (NOLOCK) on a.IDEvento =
b.IDEvento) as a
group by a.IDEvento;

/*actualizo la cantidad total de interesados y asistentes por evento*/
update a
set a.TotalInteresados = convert(varchar(15),b.TotalInteresados)+ '
Interesado(s)'
    ,a.TotalAsistentes = convert(varchar(15),b.TotalAsistentes)+ '
Asistente(s)'
    ,a.Interesado = b.Interesado
    ,a.Asistente = b.Asistente
from #tmpEvento as a WITH (NOLOCK)
join #tmpEntidad as b WITH (NOLOCK) on a.IDEvento = b.IDEvento;

/*si el usuario es administrador o maestro habilito para que le permita
eliminar el evento o el comentario*/
if (Select count(1)
    from AspNetUserRoles a WITH (NOLOCK)
    join AspNetRoles b WITH (NOLOCK) on a.RoleId = b.Id
    join AspNetUsers c WITH (NOLOCK) on a.UserId = c.Id
    Where c.UserName = @Usuario
    and b.Name in ('Administrador','Maestro')) > 0
BEGIN
    update #tmpEvento set PermiteEliminar = 1;
    update #tmpComentario set PermiteEliminar = 1;
end;

/*seccion para indicadores*/
create table #tmpDatoIndicador
(
    IDIndicador int primary key
    ,Valor int
);

/*inserto la cantidad total de eventos activos del usuario*/
Insert into #tmpDatoIndicador
Select 1, count(a.UserName) as ContTotalComentario
from #tmpEvento a WITH (NOLOCK)
Where a.UserName = @Usuario
Group by a.UserName;

/*inserto la cantidad de comentarios total de los eventos del usuario*/
Insert into #tmpDatoIndicador
Select 2, count(a.UserName) as ContTotalComentario
from #tmpEvento a WITH (NOLOCK)
join #tmpComentario b WITH (NOLOCK) on a.IDEvento = b.IDEvento
Where a.UserName = @Usuario
Group by a.UserName;

/*inserto la cantidad total de interesados en los eventos del usuario*/
Insert into #tmpDatoIndicador

```

```

Select 3,sum(b.TotalInteresados) as ContTotalInteresados
from #tmpEvento a WITH (NOLOCK)
join #tmpEntidad b WITH (NOLOCK) on a.IDEvento = b.IDEvento
Where a.UserName = @Usuario
Group by a.UserName;

/*inserto la cantidad total de asistentes en los eventos del usuario*/
Insert into #tmpDatoIndicador
Select 4,sum(b.TotalAsistentes) as ContTotalAsistentes
from #tmpEvento a WITH (NOLOCK)
join #tmpEntidad b WITH (NOLOCK) on a.IDEvento = b.IDEvento
Where a.UserName = @Usuario
Group by a.UserName;

select a.IDIndicador
      ,a.Nombre
      ,convert(varchar(50), isnull(b.Valor,0)) Valor
      ,a.Icono
      ,a.Clase
      ,a.Controlador
      ,a.Accion
      ,a.Orden
      ,a.Visible
      ,a.Activo
into #tmpIndicador
from Indicador a
left join #tmpDatoIndicador b on a.IDIndicador = b.IDIndicador;

/*obtengo consulta*/
select * from #tmpEvento order by UltimaFecha desc, Prioridad;
select * from #tmpComentario;
select * from #tmpIndicador order by Orden;

/***** Object: StoredProcedure [dbo].[GetAsistentes] Script Date:
25/02/2019 9:35:13 *****/
/*Obtiene listado de asistentes a los eventos*/
CREATE procedure [dbo].[GetAsistentes]
@IDEvento [nvarchar](128) = ''

AS

Select a.IDEvento
      ,a.Nombre
      ,b.UserName
      ,dbo.[UserProfileImage](b.UserName) as UserProfileImage
from Evento a
join EventoAsistente b on a.IDEvento = b.IDEvento
where a.IDEvento = @IDEvento;

GO

/***** Object: StoredProcedure [dbo].[GetChanges] Script Date: 25/02/2019
9:35:13 *****/
/*Obtiene los cambios realizados a la base haciendo un conteo de los registros
de la tabla de auditoria*/

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE procedure [dbo].[GetChanges]
as

select COUNT_BIG(*) as Cont
from [dbo].[Auditoria]

/***** Object: StoredProcedure [dbo].[GetComentarioEventoID]    Script Date:
25/02/2019 9:35:13 *****/
/*Obtiene los comentarios según el ID del Evento*/

CREATE procedure [dbo].[GetComentarioEventoID]
@IDEvento [nvarchar](128)
AS

Select  a.IDComentario
        ,a.IDEvento
        ,TextoComentario
        ,a.UserName
        ,dbo.[FechaHoraTexto](FechaCreacion) FechaCreacion
        ,dbo.[UserProfileImage](a.UserName) as UserProfileImage
from    Comentario a
Where  a.Activo = 1 and a.IDEvento = @IDEvento;
GO
/***** Object: StoredProcedure [dbo].[GetEstadisticaAsistencia]    Script
Date: 25/02/2019 9:35:13 *****/
/*Obtiene las estadísticas de asistencia a los eventos*/

create procedure [dbo].[GetEstadisticaAsistencia]
as

select b.Nombre
        ,count(a.IDEventoEntidad) as Cont
from  EventoEntidad a
join  Evento b on a.IDEvento = b.IDEvento
Where IDEntidad = 'as'
group by b.Nombre
order by Cont desc;
GO
/***** Object: StoredProcedure [dbo].[GetEventoEntidad]    Script Date:
25/02/2019 9:35:13 *****/
/*Obtiene las entidades relacionadas al evento*/

CREATE procedure [dbo].[GetEventoEntidad]
@IDEvento [nvarchar](128) = ''
,@IDEntidad varchar(2) = ''
AS

Select  a.IDEvento
        ,a.Nombre
        ,b.UserName
        ,dbo.[UserProfileImage](b.UserName) as UserProfileImage
from    Evento a
join    EventoEntidad b on a.IDEvento = b.IDEvento
where  a.IDEvento = @IDEvento
and    b.IDEntidad = @IDEntidad;

/***** Object: StoredProcedure [dbo].[GetInteresados]    Script Date:
25/02/2019 9:35:13 *****/

```

```

/**Obtiene el listado de interesados al evento */

create procedure [dbo].[GetInteresados]
@IDEvento [nvarchar](128) = ''

AS

Select  a.IDEvento
        ,a.Nombre
        ,b.UserName
        ,dbo.[UserProfileImage](b.UserName) as UserProfileImage
from    Evento a
join    EventoInteresado b on a.IDEvento = b.IDEvento
where  a.IDEvento = @IDEvento;

/***** Object: StoredProcedure [dbo].[GetListaIndicadores]    Script Date:
25/02/2019 9:35:13 *****/
/**Obtiene el listado de indicadores por evento */

CREATE PROCEDURE [dbo].[GetListaIndicadores]
    @Fecha varchar(14) = ''
    ,@Usuario varchar(50) = ''
    ,@Opcion varchar(50) = ''
as
Declare @CalendarDefault as varchar(100) = '../Content/img/calendar.png';
Declare @FechaActual as datetime;

if isnull(@Fecha, '') = ''
    select @FechaActual = getdate();
else
    select @FechaActual = convert(datetime,stuff(stuff(stuff(@Fecha, 9, 0,
' '), 12, 0, ':'),15, 0, ':'));

if (@opcion = 'ListaActivos')
Begin

    select a.IDEvento
           --,dbo.[UserProfileImage](a.UserName) as ImagenUrl
           ,case when a.Imagen = '' then @CalendarDefault else a.Imagen
end as ImagenUrl
        ,a.UserName
        ,a.Nombre
        ,a.FechaCreacion
        ,0 as Cont
from    Evento as a WITH (NOLOCK)
where  DATEADD(HOUR,5, a.FechaFin) > @FechaActual
and a.Estado <> 'N'
and a.UserName = @Usuario;

end;
else if (@opcion = 'ListaComentarios')
Begin
    select a.IDEvento
           --,dbo.[UserProfileImage](a.UserName) as ImagenUrl
           ,case when a.Imagen = '' then @CalendarDefault else a.Imagen
end as ImagenUrl
        ,a.UserName
        ,a.Nombre
        ,a.FechaCreacion

```



```

        ,count(a.IDEvento) as Cont
    from Evento as a WITH (NOLOCK)
    join Comentario b WITH (NOLOCK) on a.IDEvento = b.IDEvento
    where DATEADD(HOUR,5, a.FechaFin) > @FechaActual
    and a.Estado <> 'N'
    and b.Activo = 1
    and a.UserName = @Usuario
    group by a.IDEvento,a.UserName,a.Nombre,a.FechaCreacion, a.Imagen;
end
else if (@opcion = 'ListaInteresados')
Begin
    select a.IDEvento
           --,dbo.[UserProfileImage](a.UserName) as ImagenUrl
           ,case when a.Imagen = '' then @CalendarDefault else a.Imagen
    end as ImagenUrl
           ,a.UserName
           ,a.Nombre
           ,a.FechaCreacion
           ,count(a.IDEvento) as Cont
    from Evento as a WITH (NOLOCK)
    join EventoEntidad b WITH (NOLOCK) on a.IDEvento = b.IDEvento
    where DATEADD(HOUR,5, a.FechaFin) > @FechaActual
    and a.Estado <> 'N'
    and b.IDEntidad = 'mi'
    and a.UserName = @Usuario
    group by a.IDEvento,a.UserName,a.Nombre,a.FechaCreacion,a.Imagen;
end
else if (@opcion = 'ListaAsistentes')
Begin
    select a.IDEvento
           --,dbo.[UserProfileImage](a.UserName) as ImagenUrl
           ,case when a.Imagen = '' then @CalendarDefault else a.Imagen
    end as ImagenUrl
           ,a.UserName
           ,a.Nombre
           ,a.FechaCreacion
           ,count(a.IDEvento) as Cont
    from Evento as a WITH (NOLOCK)
    join EventoEntidad b WITH (NOLOCK) on a.IDEvento = b.IDEvento
    where DATEADD(HOUR,5, a.FechaFin) > @FechaActual
    and a.Estado <> 'N'
    and b.IDEntidad = 'as'
    and a.UserName = @Usuario
    group by a.IDEvento,a.UserName,a.Nombre,a.FechaCreacion,a.Imagen;
end
GO
/***** Object: StoredProcedure [dbo].[GetMenu]    Script Date: 25/02/2019
9:35:13 *****/
/**Obtiene el menú de opciones en base a los permisos asignados **/

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure [dbo].[GetMenu]
    @UserName varchar(50) = ''
as

--declare @UserName varchar(50) = 'rodrigo_medina'

```

```

Select c.Nombre as Menu
      ,b.Nombre as SubMenu
      ,a.Controlador as Controlador
      ,a.Accion as Accion
      ,c.Icono as Icono
      ,a.Visible as Visible
from PermisoRol a
join SubMenu b on a.SubMenuID = b.ID
join Menu c on b.MenuID = c.ID
join AspNetUserRoles d on a.RolID = d.RoleId
join AspNetUsers e on d.UserId = e.Id
where a.Controlador is not null
      and a.Accion is not null
      and e.UserName = @UserName
group by c.Nombre
        ,b.Nombre
        ,a.Controlador
        ,a.Accion
        ,c.Icono
        ,a.Visible;

/***** Object: StoredProcedure [dbo].[NotificacionEventoInvitado] Script
Date: 25/02/2019 9:35:13 *****/
/*Procedimiento que genera la estructura para el envío de la notificación al
evento recibido por parametro*/
CREATE PROCEDURE [dbo].[NotificacionEventoInvitado]
    @IDEvento [nvarchar](128) = ''
    ,@Fecha varchar(14) = ''
as

Declare @html as varchar(max) = '';
declare @FechaInicio varchar(200) = '';
Declare @FechaActual as datetime;
declare @Nombre varchar(50) = '';
declare @Descripcion varchar(300) = '';
declare @LocationURL varchar(300) = '';
declare @ImagenURL varchar(300) = '';
declare @Lugar varchar(100) = '';
declare @Invitados varchar(max) = '';
declare @SENDGRID_APIKEY varchar(200) = '';
declare @ADMINNAME varchar(150) = '';
declare @ADMINMAIL varchar(150) = '';

if isnull(@Fecha, '') = ''
    select @FechaActual = getdate();
else
    select @FechaActual = convert(datetime,stuff(stuff(stuff(@Fecha, 9, 0,
' '), 12, 0, ':'),15, 0, ':'));

Select @SENDGRID_APIKEY = ValorTexto
from Parametro
where Codigo = 'SENDGRID_APIKEY';

Select @ADMINNAME = ValorTexto
from Parametro
where Codigo = 'ADMINNAME';

Select @Nombre = Nombre
      ,@FechaInicio = CASE WHEN
CONVERT(date, FechaInicio)=CONVERT(date, @FechaActual) then 'HOY'

```

```

                                WHEN
CONVERT(date, FechaInicio)=CONVERT(date, DATEADD(DAY,1,@FechaActual)) then
'MAÑANA'
                                else dbo.[FechaTexto](FechaInicio)
end + ', a las ' + dbo.[HoraTexto](a.FechaInicio)
    ,@Descripcion = Descripcion
    ,@Lugar = Lugar
    ,@LocationURL = 'https://www.google.com/maps/search/?api=1&query=' +
Lat + ', ' + Long + ' '
    ,@ImagenURL = Imagen
    ,@ADMINMAIL = b.Email
from Evento a
left join aspnetUsers b on a.UserName = b.UserName
where IDEvento = @IDEvento;

set @html = '<head></head><body><h1 style="color: #5e9ca0;">' + @Nombre +
'</h1>
                <h2 style="color: #2e6c80;"><span style="color:
#ff0000;">' + @FechaInicio + '</span></h2>
                <p>' + @Descripcion + '&nbsp;</p>
                <p>El evento se llevará a cabo en <span style="background-
color: #008000; color: #ffffff; display: inline-block; padding: 3px 10px;
font-weight: bold; border-radius: 5px;"><a href="'
                + @LocationURL + ' ' style="color:white" >' + @Lugar +
'</a></span></p>
                <p>&nbsp;</p>
                <p><br /><strong>Te esperamos!</strong></p>';

                if @ImagenURL<>' '
                begin
                set @html = @html + '<table><tr><td valign="top"
align="left"></td></tr>
                                <tr><td
                                valign="top" align="left">
                                
                                </td></tr>
                                </table>';
                end;

                set @html = @html +
'<p><strong>&nbsp;</strong></p></body>';

select @IDEvento as IDEvento
    ,@ADMINMAIL as De
    ,@ADMINNAME as Nombre
    ,@Nombre as Asunto
    ,@html as Contenido
    ,@SENDGRID_APIKEY as ApiKey;

/***** Object: StoredProcedure [dbo].[ProcEventoEntidad] Script Date:
25/02/2019 9:35:13 *****/
/*inserta o elimina las entidades relacionadas a un evento*/

CREATE PROCEDURE [dbo].[ProcEventoEntidad]
    @IDEvento [nvarchar](128) = ''
    ,@IDEntidad varchar(2) = ''
    ,@UserName varchar(50) = ''
    ,@FechaCreacion varchar(14) = ''
as

```

Begin

```
if (select count(1)
    from EventoEntidad WITH (NOLOCK)
    where IDEvento = @IDEvento
    and IDEntidad = @IDEntidad
    and UserName = @UserName) > 0
begin
    delete from EventoEntidad
    where IDEvento = @IDEvento
    and IDEntidad = @IDEntidad
    and UserName = @UserName;
end
ELSE
BEGIN
    Declare @FechaActual as datetime;

    if isnull(@FechaCreacion, '') = ''
        select @FechaActual = getdate();
    else
        select @FechaActual =
convert(datetime,stuff(stuff(stuff(@FechaCreacion, 9, 0, ' '), 12, 0, ':'),15,
0, ':'));

        insert into EventoEntidad values
(@IDEvento,@IDEntidad,@UserName,@FechaActual);
end
```

End

ANEXO B: MATRICES DE PRUEBA

Registro de usuario en el sistema

Código Caso de uso/Requerimiento:	CUI Registro de usuario en el sistema
Descripción del Caso/Requerimiento:	El usuario ingresa al portal web en la opción de crear cuenta. Digita un usuario y un correo electrónico y click en registrar. El sistema valida datos registrados y genera una contraseña que se envía al correo electrónico.
Usuarios participantes:	Usuario Tester
Ambiente	Desarrollo

Nº	Escenarios	[MENU OPCION]	PARÁMETROS	RESULTADOS		VALORACIÓN			OBS	ANEXOS	
				Esperados	Obtenidos	Tipo Error	Nivel Error	Estado Prueba			
1	Ingreso de formulario incompleto	Pantalla de Login	<Parámetro 1: UserName> <Parámetro 2: EMail>	null null	Mensaje que indique que los campos son obligatorios	Se muestra mensaje indicando que tanto usuario como correo son obligatorios	Funcional		Exitosa		Figura 29
2	Ingreso de formulario con usuario ya existente	Pantalla de Login	<Parámetro 1: UserName> <Parámetro 2: EMail>	Clowlock null	Mensaje que indique que usuario no está disponible	Se muestra mensaje indicando que el usuario no se encuentra disponible	Funcional		Exitosa		Figura 30
3	Ingreso de formulario con correo electrónico ya existente	Pantalla de Login	<Parámetro 1: UserName> <Parámetro 2: EMail>	prueba crichaag@gmail.com	Mensaje que indique que e-mail no está disponible	Se muestra mensaje indicando que el e-mail no se encuentra disponible	Funcional		Exitosa		Figura 31
4	Ingreso de formulario con correo electrónico no válido	Pantalla de Login	<Parámetro 1: UserName> <Parámetro 2: EMail>	prueba dddddd@gdsfsd.com	Registro ok	Registro ok	Funcional		Exitosa	El servicio de envío de correo no puede validar si una dirección es válida.	

Tabla 18. Matriz de Prueba de Registro de Usuario en el Sistema. Fuente: el autor

Creación de eventos con geolocalización

Código Caso de uso/Requerimiento:	CU4 Creación de eventos con geolocalización
Descripción del Caso/Requerimiento:	Mediante el aplicativo web en la bandeja de entrada estará habilitado el botón de crear evento. Se registran los datos principales, se especifica geolocalización y se graba el formulario. Se genera una notificación a los invitados al evento y cuando el evento este por comenzar se genera una alerta.
Usuarios participantes:	Usuario Tester
Ambiente	Desarrollo

N°	Escenarios	[MENU OPCION]	PARÁMETROS	RESULTADOS		VALORACIÓN			OBS	ANEXOS
				Esperados	Obtenidos	Tipo Error	Nivel Error	Estado Prueba		
1	Visualización de formulario de creación de evento con mapa cargado de api de Google	Transacciones> Evento> Crear Nuevo	<Parámetro 1:> null <Parámetro 2: > null	Pantalla cargada correctamente	Error	Técnico	Menor	Fallida	Hay que agregar validación para verificar si está habilitada la opción de ubicación del navegador.	Figura 32
2	Ingreso de formulario incompleto	Transacciones> Evento> Crear Nuevo	<Parámetro 1: Nombre del Evento> null <Parámetro 2: Tipo de Evento> null <Parámetro 3: FechaInicio> null <Parámetro 4: FechaFin> null <Parámetro 5: InfoEvento> null <Parámetro 6: Ubicación> null	Mensaje que indique que los campos son obligatorios	Se muestra mensaje indicando que todos los parámetros son obligatorios	Funcional		Exitosa		Figura 33
3	Fecha de inicio menor a la actual	Transacciones> Evento> Crear Nuevo	Fecha Actual 20/2/2019 23:24 <Parámetro 1: FechaInicio> 20/2/2019 22:00	Mensaje que indique que la fecha de inicio no puede ser menor a la fecha actual menos un intervalo de 15 minutos	Se muestra mensaje indicando que la fecha de inicio ser hasta 15 minutos antes a la actual	Funcional		Exitosa		Figura 34
4	Fecha de fin menor a la actual	Transacciones> Evento> Crear Nuevo	Fecha Actual 20/2/2019 23:30 <Parámetro 1: FechaFin> 20/2/2019 22:00	Mensaje que indique que la fecha de fin no puede ser menor a la fecha actual	Se muestra mensaje indicando que la fecha de fin debe ser mayor o igual a la actual	Funcional		Exitosa		Figura 35
5	Fecha de inicio mayor a la fecha fin	Transacciones> Evento> Crear Nuevo	Fecha Actual 20/2/2019 23:50 <Parámetro 1: FechaInicio> 20/2/2019 23:50 <Parámetro 2: FechaFin> 20/2/2019 23:40	Mensaje que indique que la fecha de fin no puede ser menor a la fecha inicio	Se muestra mensaje indicando que la fecha de fin debe ser mayor o igual a la fecha inicio	Funcional		Exitosa		Figura 35

Tabla 19. Matriz de Prueba de Creación de Evento con Geolocalización. Fuente: el autor

N°	Escenarios	[MENU OPCION]	PARÁMETROS	RESULTADOS		VALORACIÓN			OBS	ANEXOS	
				Esperados	Obtenidos	Tipo Error	Nivel Error	Estado Prueba			
6	Generación de notificación por creación de evento	Transacciones> Evento> Crear Nuevo	<Parámetro 1: Nombre del Evento> <Parámetro 2: Tipo de Evento> <Parámetro 3: FechaInicio> <Parámetro 4: FechaFin> <Parámetro 5: InfoEvento> <Parámetro 6: Ubicación> <Parámetro 1: Usuario>	Evento de prueba Académico 11/2/2019 23:00 12/2/2019 0:00 Este es una evento de prueba Malecon 2000 maestro	Evento creado Ok, notificación enviada correctamente	Se recibió notificación al correo	Funcional		Exitosa		Figuras 36,37
7	Generación automática de alertas sobre eventos	API>Alerta>Generar	<Parámetro 2: Contraseña>	maestro	Correo con alerta	Se recibió notificación al correo	Funcional		Exitosa	Ejecuta un api por medio de un web job que recibe usuario y contraseña como parámetro	Figuras 38

Tabla 20. Matriz de Prueba de Notificaciones y Alertas de Eventos con Geolocalización. Fuente: el autor

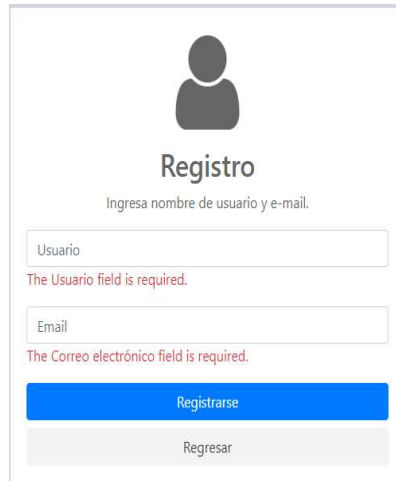
Visualización de Estadísticas de Asistencia

Código Caso de uso/Requerimiento:	CU8 Visualización de Estadísticas de Asistencia
Descripción del Caso/Requerimiento:	Mediante el aplicativo web en el menú de opciones se podrá acceder a las estadísticas de asistencia por evento.
Usuarios participantes:	Usuario Tester
Ambiente	Desarrollo

N°	Escenarios	[MENU OPCION]	PARÁMETROS	RESULTADOS		VALORACIÓN			OBS	ANEXOS	
				Esperados	Obtenidos	Tipo Error	Nivel Error	Estado Prueba			
1	Visualización de formulario con estadísticas de asistencia	Estadísticas> Asistencia	<Parámetro 1>	No recibe parámetros	Visualización de graficos estadísticos de asistencia	Graficos tipo dona con ditribución de asistencias y gráfico tipo lineal con asistencia por evento de mayor a menor	Funcional		Exitosa		Figura 39,40

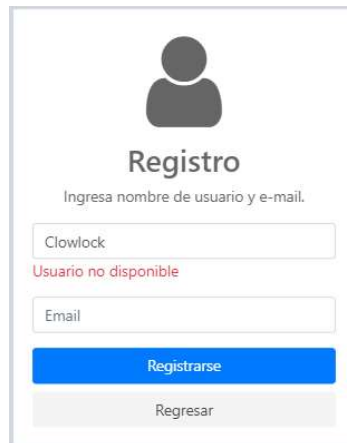
Tabla 21. Matriz de Prueba de Visualización de Estadísticas. Fuente: el autor

ANEXO C: RESULTADOS DE PRUEBAS



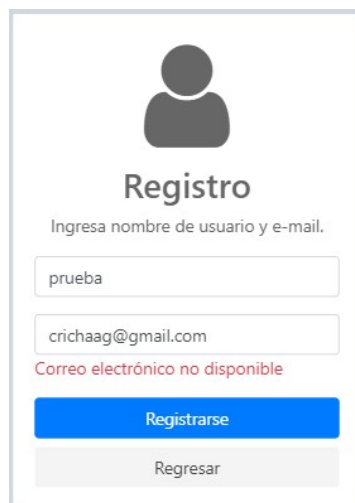
The screenshot shows a registration page titled "Registro" with the instruction "Ingresa nombre de usuario y e-mail." There are two input fields: "Usuario" and "Email". Below the "Usuario" field, a red error message reads "The Usuario field is required." Below the "Email" field, a red error message reads "The Correo electrónico field is required." At the bottom, there are two buttons: a blue "Registrarse" button and a grey "Regresar" button.

Figura 29. Ingreso de formulario incompleto. Fuente: El Autor



The screenshot shows the same registration page. The "Clowlock" field is now filled with the text "Clowlock". A red error message below it reads "Usuario no disponible". The "Email" field remains empty. The "Registrarse" and "Regresar" buttons are still present at the bottom.

Figura 30. Ingreso de formulario con usuario ya existente. Fuente: El Autor



The screenshot shows the registration page with both fields filled: "prueba" in the "Usuario" field and "crichaag@gmail.com" in the "Email" field. A red error message below the email field reads "Correo electrónico no disponible". The "Registrarse" and "Regresar" buttons are at the bottom.

Figura 31. Ingreso de formulario con correo ya existente. Fuente: El Autor

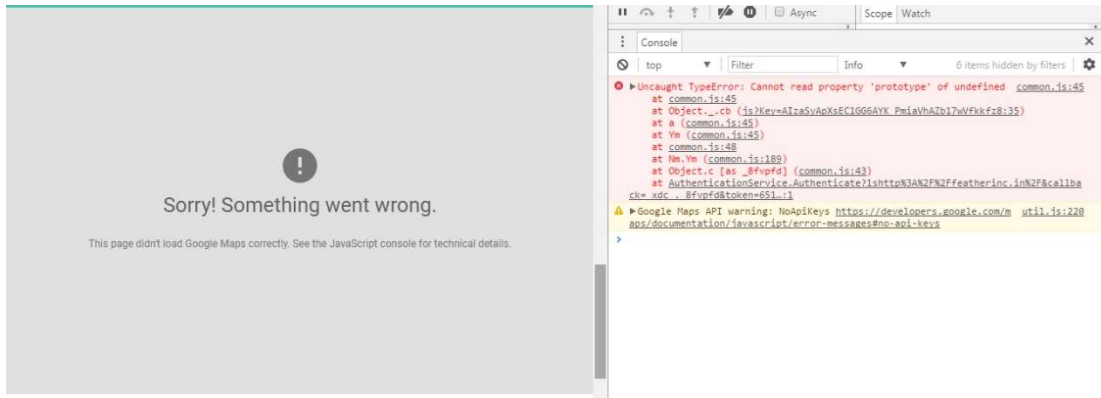


Figura 32. Error al cargar pantalla para creación de evento sin validar geolocalización. Fuente: El Autor

Figura 33. Prueba de creación de evento: validaciones sobre campos obligatorios. Fuente: El Autor

Figura 34. Prueba de creación de evento: validación sobre fecha de inicio. Fuente: El Autor

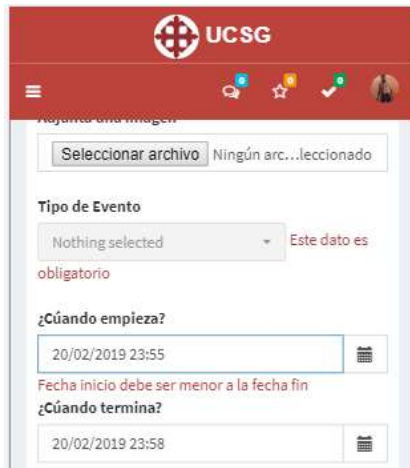


Figura 35. Prueba de creación de evento: validación sobre fecha de fin. Fuente: El Autor

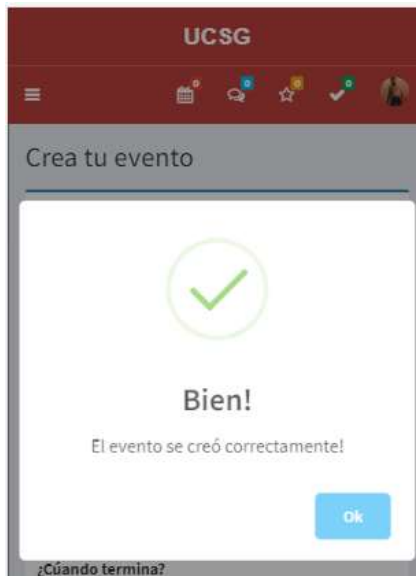


Figura 36. Prueba de creación de evento generación exitosa. Fuente: El Autor

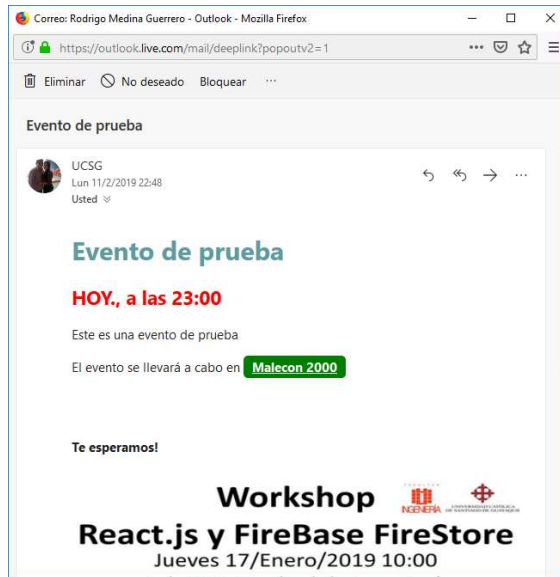


Figura 37. Notificación sobre creación de evento. Fuente: El Autor

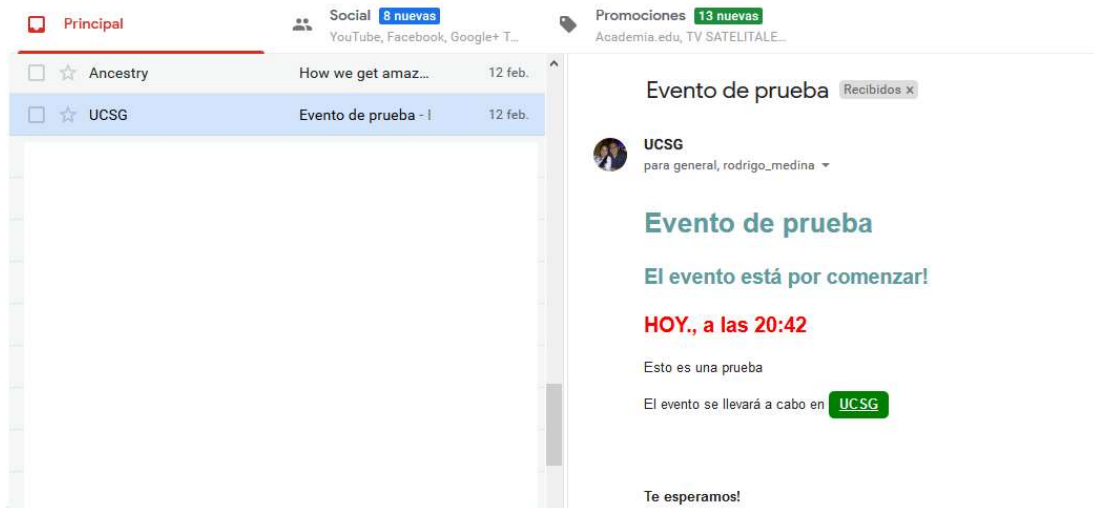


Figura 38. Notificación sobre alerta de evento. Fuente: El Autor



Figura 39. Prueba de Estadísticas gráfico tipo Lineal. Fuente: el autor



Figura 40. Prueba de Estadísticas gráfico tipo Dona. Fuente: el autor

ANEXO D: MANUAL DE USUARIO

El siguiente manual tiene como objetivo describir los pasos a realizar para cada una de las opciones disponibles en el sistema, con el fin de facilitar su uso y manejo.

Acceso al portal web

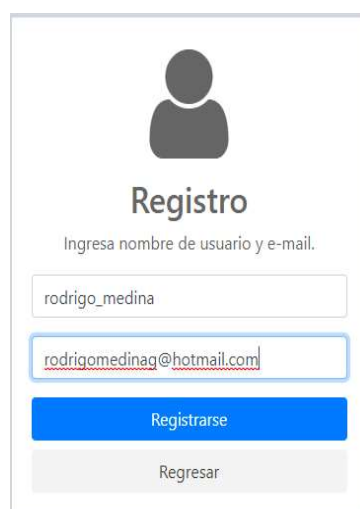
Para acceder al portal donde está alojada la aplicación hay que escribir la dirección <https://ucsg.azurewebsites.net/> en el navegador web de su preferencia.

Ingreso al sistema

Para poder comenzar a utilizar la aplicación es necesario contar con una cuenta de usuario debidamente registrada, a continuación, veremos cómo crear una cuenta y cómo ingresar al sistema si ya se poseen credenciales de acceso.

Creación de usuario

- Acceder al portal <https://ucsg.azurewebsites.net/>
- Dar click en el botón “Registrarse”
- Llenar el formulario con un nombre de usuario y una cuenta de correo electrónico válida.
- Abrir el correo con el cual se registró y verificar que se haya recibido la notificación con su usuario y contraseña para acceder al sistema.



The image shows a registration form titled "Registro". At the top, there is a grey silhouette of a person. Below the title, the text "Ingresa nombre de usuario y e-mail." is displayed. There are two input fields: the first contains "rodrigo_medina" and the second contains "rodrigomedinag@hotmail.com". Below the input fields are two buttons: a blue "Registrarse" button and a grey "Regresar" button.

Figura 41. Pantalla para creación de usuario. Fuente: El Autor

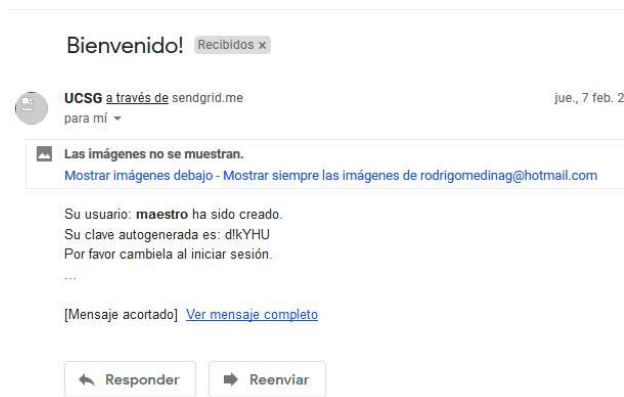


Figura 42. Notificación recibida con datos de acceso al sistema. Fuente: El Autor

Inicio de sesión

- Acceder al portal <https://ucsg.azurewebsites.net/>
- Llenar el formulario de inicio de sesión con usuario y contraseña.
- Una vez autenticado se cargará la bandeja de entrada.

Olvido de contraseña

- Acceder a la dirección <https://ucsg.azurewebsites.net/Seguridad/OlvidoPassword>
- Ingresar el correo electrónico con el cuál se registró en el sistema.
- Se recibirá un correo con una nueva contraseña autogenerada.

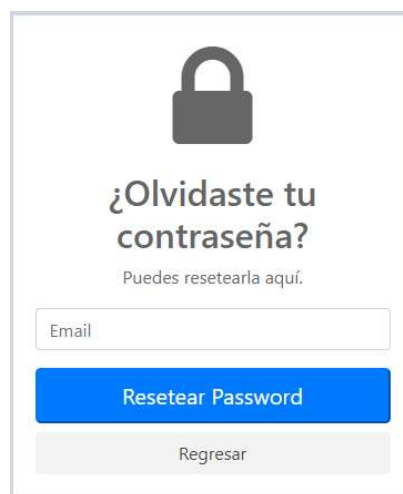


Figura 43. Pantalla para restaurar contraseña. Fuente: El Autor

Dashboard

El Dashboard o bandeja principal es la primera pantalla que se muestra al iniciar sesión, contiene los eventos creados y que aún se encuentran vigentes. Se permite la interacción por medio de los botones de “Me Interesa” y “Asistiré”. Además, permite ingresar comentarios y poder visualizarlos en tiempo real.

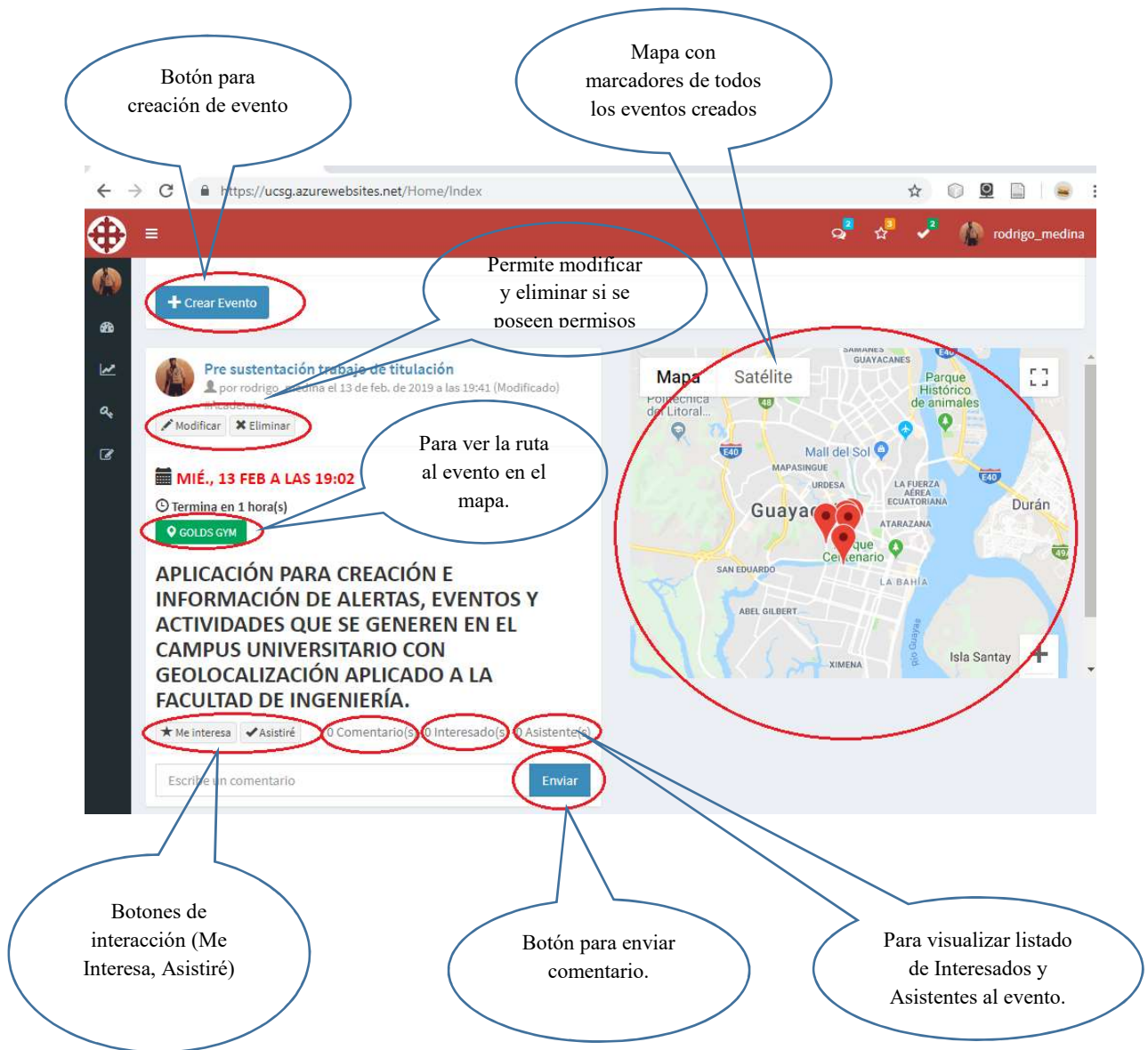


Figura 44. Dashboard. Fuente: El Autor

Menú de Opciones

Es la barra de opciones ubicada al lado izquierda de la pantalla la cual contiene las opciones disponibles para el usuario dependiendo de los permisos asignados. Todo usuario que se registra al sistema tiene asignado por defecto el rol Usuario, el cual le permite interactuar con los eventos y recibir notificaciones y alertas sobre los mismos.



Figura 45. Menú de Opciones. Fuente: El Autor

Barra Superior

Esta barra está ubicada en la parte superior de la aplicación, contiene 3 iconos, el primero (icono azul) es el de comentarios recibidos en los eventos que se hayan creado, el segundo (icono naranja) muestra el listado de interesados por evento creado, y el tercero (icono verde) muestra el listado de asistentes por evento. Junto a estos iconos se encuentran disponibles las opciones de cambio de foto de perfil, cambio de contraseña y salir del sistema.

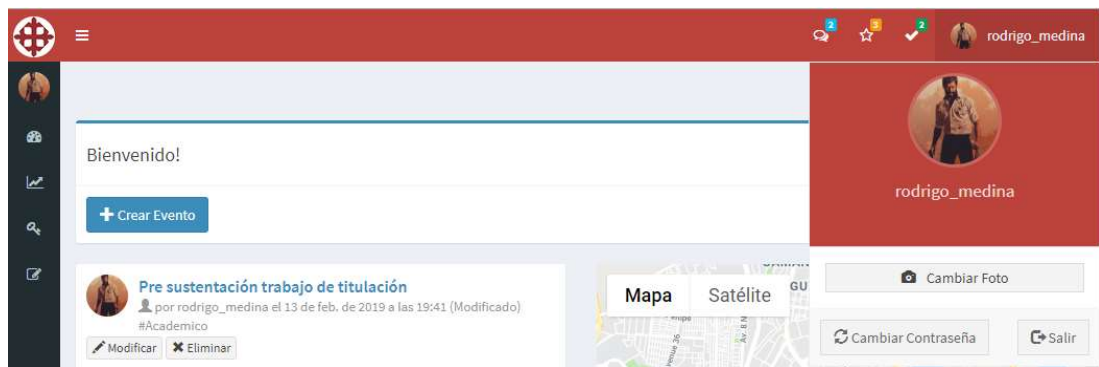


Figura 46. Barra Superior. Fuente: El Autor

Creación de Eventos

Esta opción está disponible por defecto para el administrador del sistema y puede ser asignada a otro rol dependiendo de la necesidad. Permite la creación de un evento ingresando sus datos principales como nombre, descripción, tipo de evento, fecha de inicio y fecha de fin, y la ubicación donde se llevará a cabo el mismo con geolocalización.

Además, se puede adjuntar una imagen sobre el evento para que pueda ser visualizada por los demás usuarios. Se debe elegir del listado de usuarios a quién se desea invitar al mismo para que le llegue la notificación y alerta correspondiente.

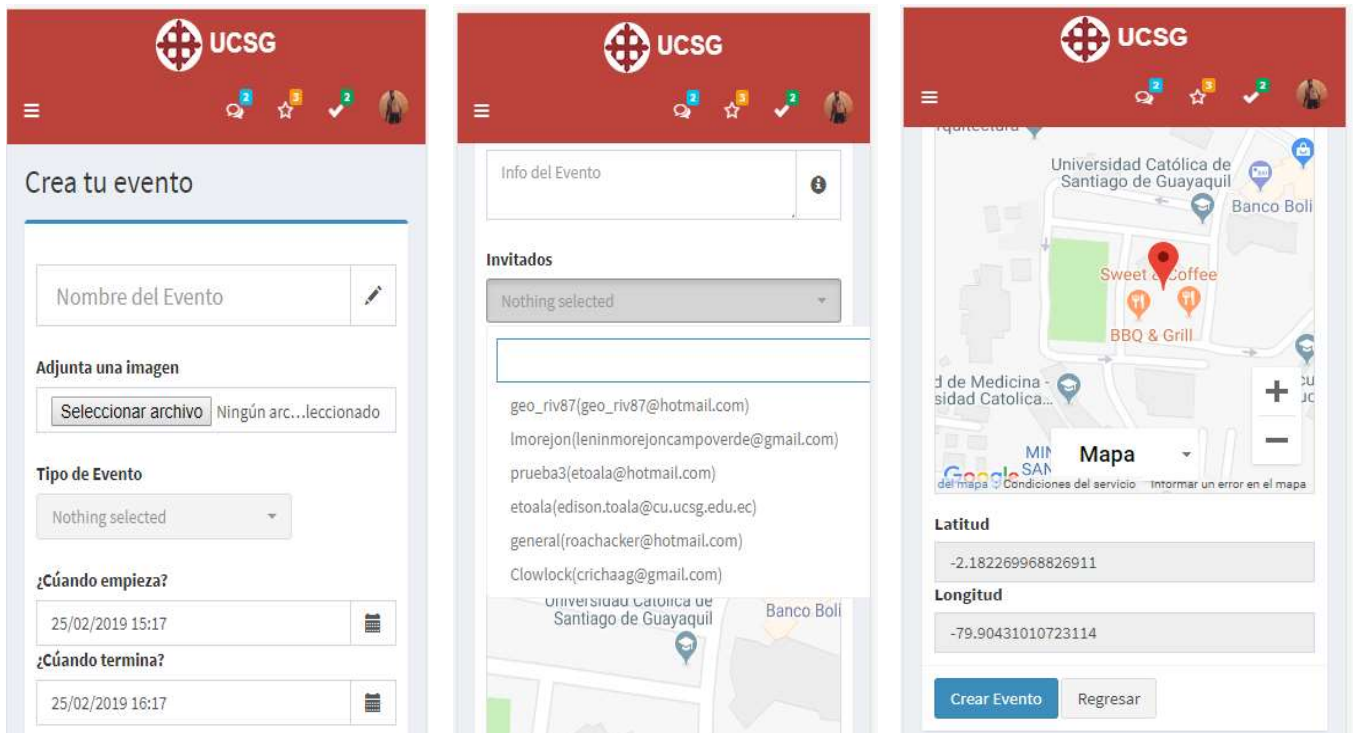


Figura 47. Creación de Eventos. Fuente: El Autor

Estadísticas de Asistentes

Esta opción permite visualizar las estadísticas de asistencia de los eventos creados por medio de dos gráficos, el uno tipo lineal que muestra en orden descendente la cantidad de asistencia por evento y el otro tipo dona que muestra la distribución de asistencia por evento. Esta opción se encuentra disponible en la dirección <https://ucsg.azurewebsites.net/Estadistica/Asistente> o por medio del menú de opciones en Estadísticas->Asistentes.

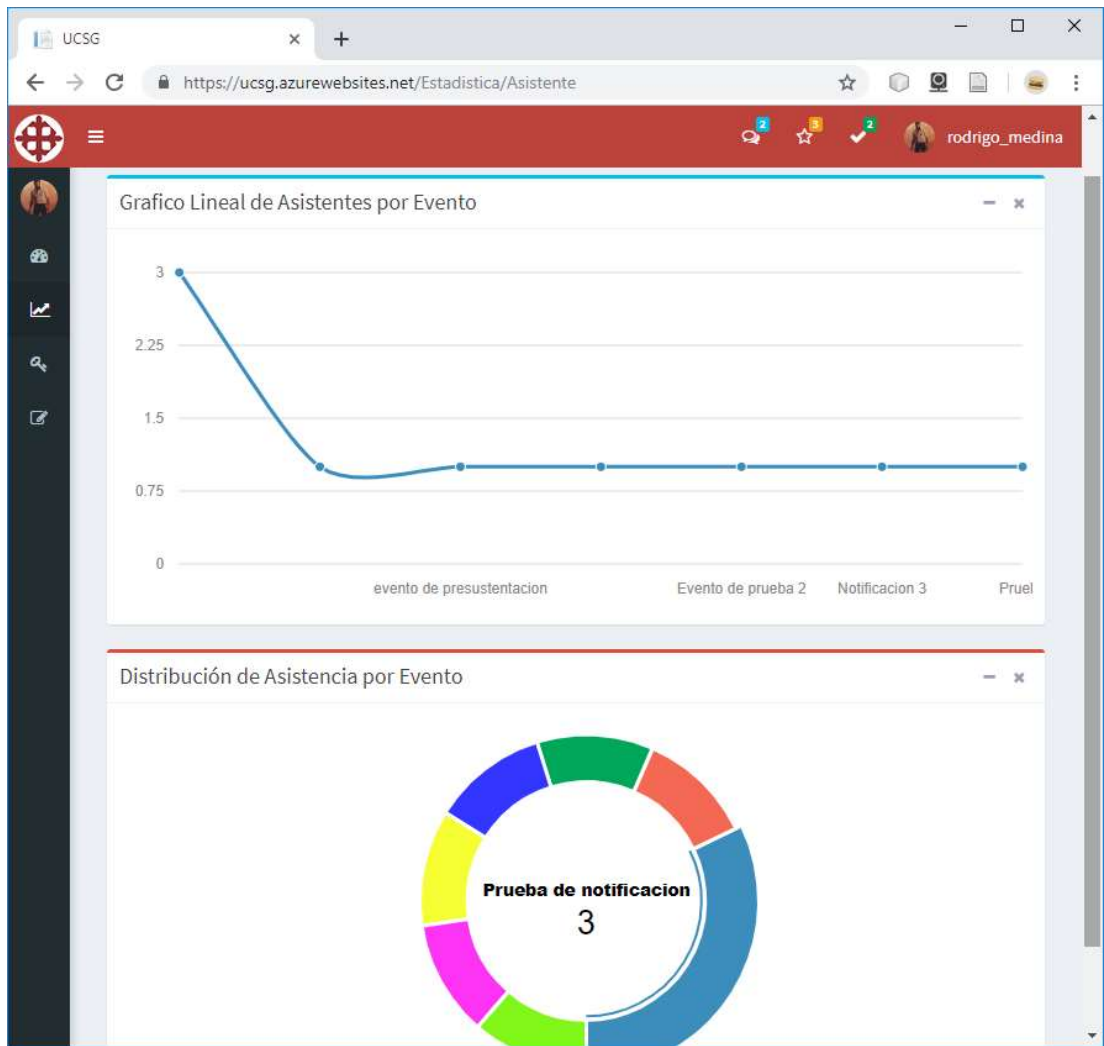
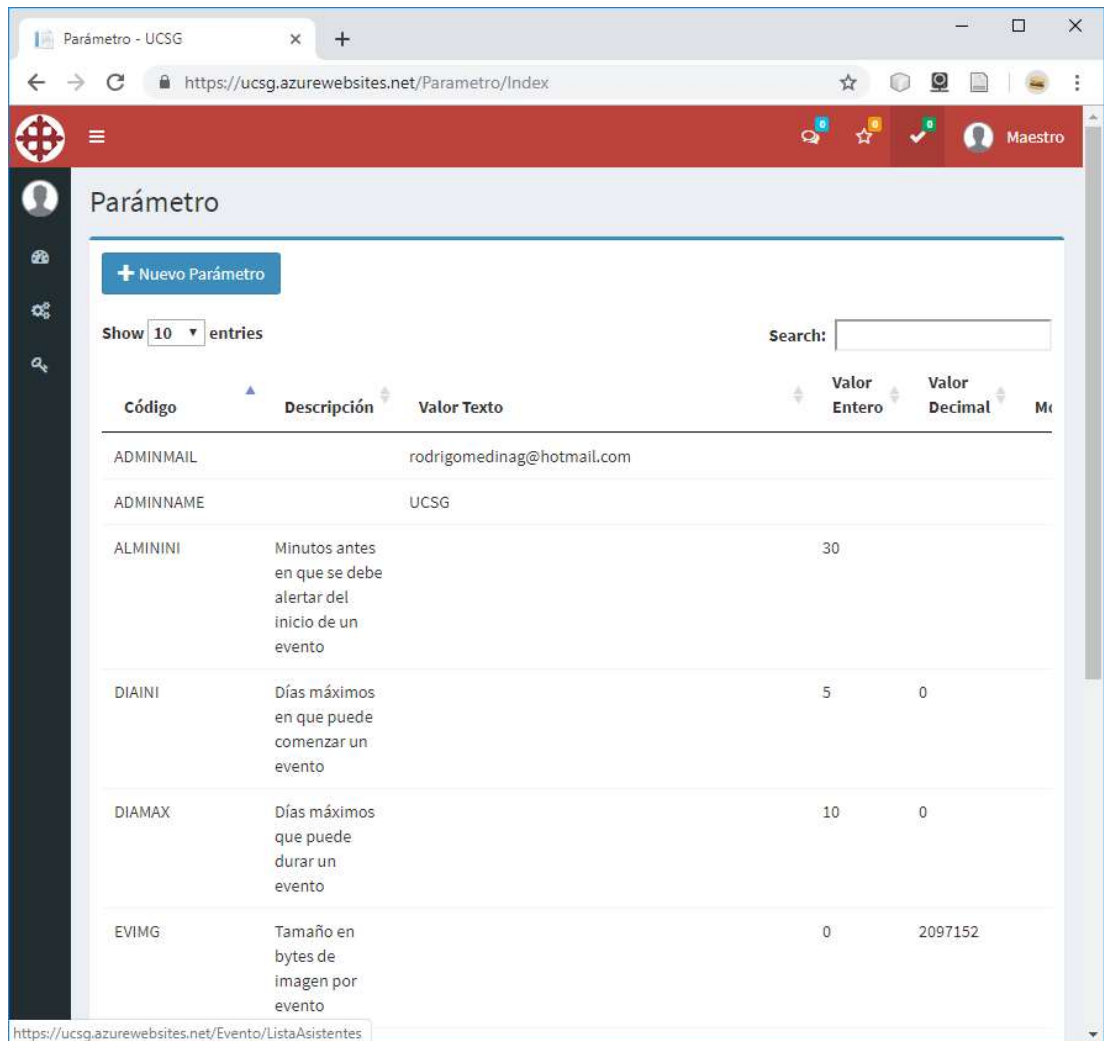


Figura 48. Estadística de Asistentes. Fuente: El Autor

Parámetros

Mediante esta opción se permite configurar los distintos parámetros necesarios para el correcto funcionamiento de la aplicación. Se encuentra disponible accediendo a la dirección <https://ucsg.azurewebsites.net/Parametro/Index> o por medio del menú de opciones en Configuración->Parámetros.



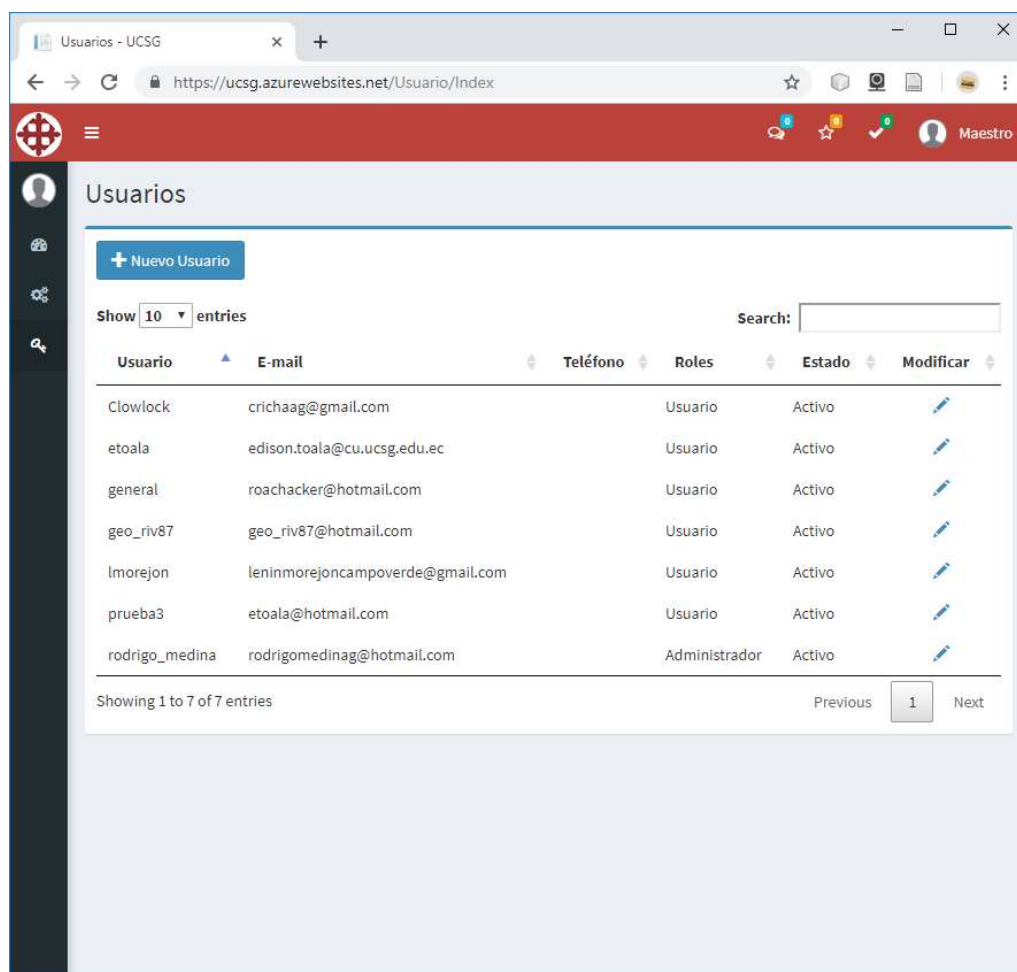
The screenshot shows a web browser window with the URL <https://ucsg.azurewebsites.net/Parametro/Index>. The page title is 'Parámetro'. There is a '+ Nuevo Parámetro' button and a 'Show 10 entries' dropdown. A search bar is present. The table below lists several parameters:

Código	Descripción	Valor Entero	Valor Decimal	Mt
ADMINMAIL				rodrigomedinag@hotmail.com
ADMINNAME				UCSG
ALMININI	Minutos antes en que se debe alertar del inicio de un evento	30		
DIAINI	Días máximos en que puede comenzar un evento	5	0	
DIAMAX	Días máximos que puede durar un evento	10	0	
EVIMG	Tamaño en bytes de imagen por evento	0	2097152	

Figura 49. Parámetros. Fuente: El Autor

Administración de Usuarios

Mediante esta opción se permite administrar los usuarios del sistema, permite crear nuevos usuarios, actualizar su correo electrónico, bloquearlo del sistema o asignarle o quitarle roles. Se encuentra disponible accediendo a la dirección <https://ucsg.azurewebsites.net/Usuario/Index> o por medio del menú de opciones en Seguridad->Usuarios.



The screenshot displays the 'Usuarios' management interface. At the top, there is a '+ Nuevo Usuario' button. Below it, a search bar is labeled 'Search:'. The main content is a table with the following columns: 'Usuario', 'E-mail', 'Teléfono', 'Roles', 'Estado', and 'Modificar'. The table contains 7 entries. The first six entries have the role 'Usuario' and 'Activo' status, while the last entry, 'rodrigo_medina', has the role 'Administrador' and 'Activo' status. Each entry has a 'Modificar' icon (pencil) to its right. At the bottom of the table, it says 'Showing 1 to 7 of 7 entries' and includes 'Previous' and 'Next' navigation buttons.

Usuario	E-mail	Teléfono	Roles	Estado	Modificar
Clowlock	crichaag@gmail.com		Usuario	Activo	
etoala	edison.toala@cu.ucsg.edu.ec		Usuario	Activo	
general	roachacker@hotmail.com		Usuario	Activo	
geo_riv87	geo_riv87@hotmail.com		Usuario	Activo	
lmorejon	leninmorejoncampoverde@gmail.com		Usuario	Activo	
prueba3	etoala@hotmail.com		Usuario	Activo	
rodrigo_medina	rodrigomedinag@hotmail.com		Administrador	Activo	

Figura 50. Administración de Usuarios. Fuente: El Autor

Roles de Usuario

Mediante esta opción se permite administrar los roles del sistema. Se encuentra disponible accediendo a la dirección <https://ucsg.azurewebsites.net/Rol/Index> o por medio del menú de opciones en Seguridad->Roles.

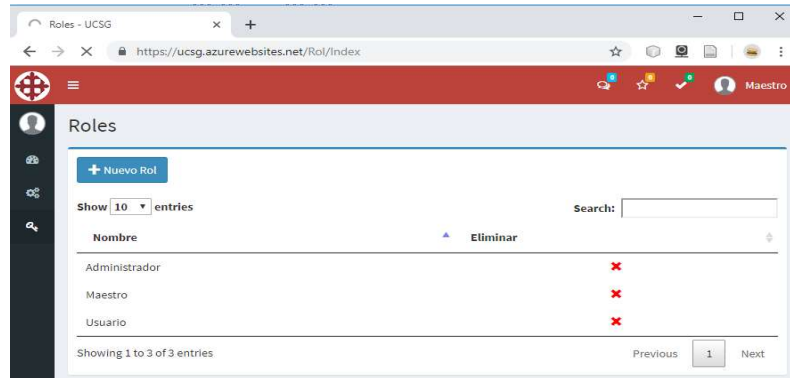


Figura 51. Roles. Fuente: El Autor

Permisos por Rol

Mediante esta opción se permite administrar los permisos por cada Rol de usuario. Se encuentra disponible accediendo a la dirección <https://ucsg.azurewebsites.net/PermisoRol/Index> o por medio del menú de opciones en Seguridad->Permisos por Rol.

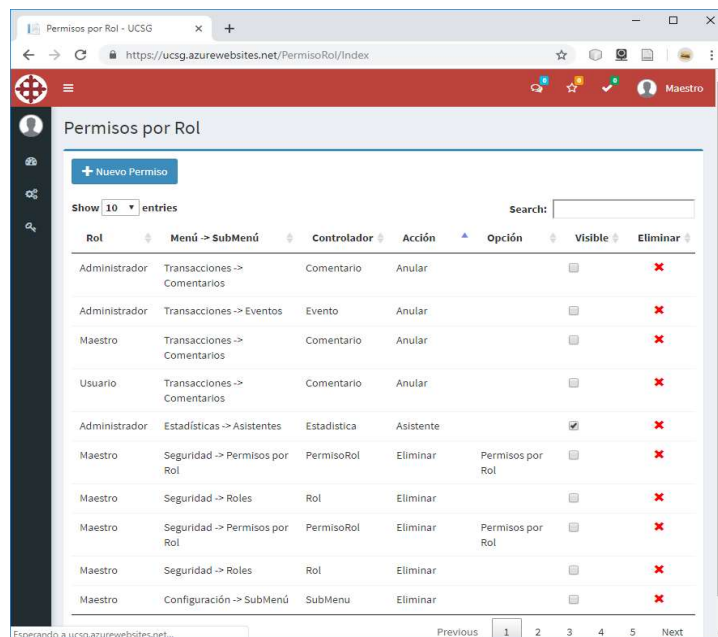


Figura 52. Permisos por Rol. Fuente: El Autor

ANEXO E: MANUAL TÉCNICO

El siguiente manual tiene como objetivo describir la arquitectura de desarrollo utilizada en el proyecto e indicar para qué sirven las componentes del sistema con el fin de que su ejecución sea satisfactoria.

El sistema fue desarrollado utilizando .NET Framework v4.6.2 y EntityFramework v6.2.0 para la gestión de entidades creadas en la base de datos con la aplicación empleando el esquema DataBase First que consiste en tener un modelo ER en la base y el EntityFramework lo toma como referencia para generar todas las entidades y poderlas utilizar realizando instancias a esas clases autogeneradas.

Por otra parte, se utilizó la arquitectura de software aspNet.MVC v5.2.6 que básicamente consiste en separar la aplicación en 3 componentes que son Modelos, es decir entidades que se utilizan en el aplicativo como Eventos, Comentarios, etc.; Vistas, que son la interfaz gráfica para el usuario y Controlador que es donde se van a realizar los procesos del lado del servidor.

Controladores

AlertaController: contiene las acciones para la generación y notificación de las alertas por correo electrónico.

- **Generar:** acción que recibe como parámetros usuario y contraseña, llama al procedimiento de generación de alertas y realiza la notificación por correo electrónico.

ComentarioController: contiene las acciones para la gestión de los comentarios.

- **Nuevo:** acción para insertar en la tabla de comentarios un nuevo registro.
- **Anular:** acción para cambiar el estado de un comentario en Anulado.

ErrorController: contiene las acciones para la gestión de errores que se generen en la aplicación.

- **InternalServerError:** acción que devuelve vista cuando se produce un error de servidor.
- **NotFound:** acción que devuelve vista cuando no se encuentra un resultado.
- **NoAutorizado:** acción que devuelve vista cuando se intenta acceder a otra acción a la cual no se posee permisos.
- **SessionExpired:** acción que devuelve vista cuando la sesión de usuario a expirado.

EstadisticaController: contiene las acciones para la generación de estadísticas en la aplicación.

- **Asistente:** acción ejecuta procedimiento almacenado para obtener un listado de asistentes por evento.

EventoController: contiene las acciones para la gestión de eventos.

- **Nuevo:** acción para creación de nuevo evento.
- **Modificar:** acción para modificación de evento.
- **EventoEntidad:** acción insertar o eliminar una entidad (me interesa, me gusta) relacionada a un evento.
- **Ruta:** acción que consulta un evento específico y devuelve vista que muestra la ruta desde ubicación actual hasta la ubicación del evento.
- **MostrarAsistentes:** consulta los asistentes de un evento específico.
- **MostrarInteresados:** consulta los interesados de un evento específico.
- **ValidaFechaInicio:** valida que el parámetro recibido y lo compara con la fecha de fin en base a parámetros definidos en la aplicación.
- **ValidaFechaFin:** valida que el parámetro recibido y lo compara con la fecha de inicio en base a parámetros definidos en la aplicación.

HomeController: contiene las acciones para la bandeja de entrada del aplicativo.

- **Dashboard:** acción que obtiene la bandeja de entrada del aplicativo.

MenuController: contiene las acciones para la administración del menú de opciones.

- **Index:** acción que devuelve listado de menú.
- **Nuevo:** acción para creación de nuevo menú.
- **Modificar:** acción para modificación de menú.
- **Eliminar:** acción para eliminación de menú.
- **NombreDisponible:** acción verificar si el nombre del menú a registrar ya existe.

ParametroController: contiene las acciones para la administración de parámetros del sistema.

- **Index:** acción que devuelve listado de parámetros.
- **Nuevo:** acción para creación de nuevo parámetro.
- **Modificar:** acción para modificación de parámetro.
- **Eliminar:** acción para eliminación de parámetro.
- **NombreDisponible:** acción verificar si el nombre del parámetro a registrar ya existe.

PermisoRolController: contiene las acciones para la administración de los permisos por rol del aplicativo.

- **Index:** acción que devuelve listado de permiso por rol.
- **Nuevo:** acción para creación de nuevo permiso por rol.
- **Modificar:** acción para modificación de permiso por rol.
- **Eliminar:** acción para eliminación de permiso por rol.

RolController: contiene las acciones para la administración de roles.

- **Index:** acción que devuelve listado de rol.
- **Nuevo:** acción para creación de nuevo rol.

- **Modificar:** acción para modificación de rol.
- **Eliminar:** acción para eliminación de rol.

SeguridadController: contiene las acciones para la administración de seguridad de la aplicación.

- **LogOff:** acción que finaliza la sesión de usuario.
- **Login:** acción para autenticación de usuario e inicio de sesión en el aplicativo.
- **OlvidoPassword:** acción para restablecer contraseña.
- **Registrarse:** acción para registrarse como nuevo usuario en la aplicación.
- **CambiaPassword:** acción para cambio de contraseña.
- **FotoPerfil:** acción para cambio de foto de perfil.
- **UsuarioDisponible:** acción para verificar si el nombre de usuario a registrar ya existe.
- **CorreoValido:** acción para verificar si el correo electrónico a registrar ya existe.

SubMenuController: contiene las acciones para la administración del sub menú de opciones de la aplicación.

- **Index:** acción que devuelve listado de sub menú.
- **Nuevo:** acción para creación de nuevo sub menú.
- **Modificar:** acción para modificación de sub menú.
- **Eliminar:** acción para eliminación de sub menú.

UsuarioController: contiene las acciones necesarias para la administración de usuarios del aplicativo.

- **Index:** acción que devuelve listado de usuarios.
- **Nuevo:** acción para creación de nuevo usuarios.
- **Modificar:** acción para modificación de usuarios.

Modelos

Listado de modelos que fueron autogenerados por el EntityFramework desde la base de datos, corresponden a las entidades del modelo ER creado y especificado en el capítulo 4 descripción del aplicativo.

Archivo	Menu
Comentario	Opcion
Entidad	Parametro
Evento	PermisoRol
EventoAlertaUsuario	Secuencia
EventoEntidad	SubMenu
EventoInvitado	TipoEvento

Listado de modelos creados en base a los resultados de la ejecución de procedimientos almacenados de la base de datos.

Dashboard: modelo para la bandeja de entrada del aplicativo.

GetChanges: modelo con el contador de los cambios en el aplicativo.

GetComentarios: modelo para los comentarios por evento.

GetEstadisticaAsistencia: modelo con los campos para visualizar las estadísticas de asistentes por evento.

GetEventoEntidad: modelo para obtener las entidades relacionadas a los eventos.

Vistas

Las vistas corresponden a acciones específicas dentro de cada controlador, las cuales se encuentran dentro de la ruta UCSG Solution\Views. El funcionamiento de cada vista es equivalente a lo descrito en las acciones de cada controlador.

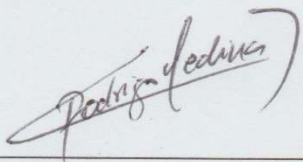
DECLARACIÓN Y AUTORIZACIÓN

Yo, **Medina Guerrero Rodrigo Fernando** con C.C.: # **0925041485**, autor del trabajo de titulación: **Aplicación para Creación e Información de Alertas, Eventos y Actividades que se Generen en el Campus Universitario con Geolocalización aplicado a la Facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil** previo a la obtención del título de **INGENIERO EN SISTEMAS COMPUTACIONALES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de graduación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 18 de marzo de 2019

f. 

Medina Guerrero Rodrigo Fernando

C.C: 0925041485

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA		
FICHA DE REGISTRO DE TESIS/TRABAJO DE GRADUACIÓN		
TÍTULO Y SUBTÍTULO:	Aplicación para Creación e Información de Alertas, Eventos y Actividades que se Generen en el Campus Universitario con Geolocalización aplicado a la Facultad de Ingeniería de la Universidad Católica de Santiago de Guayaquil	
AUTOR:	Medina Guerrero, Rodrigo Fernando	
TUTOR:	Ing. Toala Quimí Edison José, Mgs.	
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil	
FACULTAD:	Ingeniería	
CARRERA	Ingeniería en Sistemas Computacionales	
TÍTULO OBTENIDO:	Ingeniero en Sistemas Computacionales	
FECHA DE PUBLICACIÓN:	18 de marzo de 2019	No. DE PÁGINAS: 125
ÁREAS TEMÁTICAS:	Software, Servicios en la nube, Geolocalización.	
PALABRAS CLAVES/ KEYWORDS:	APLICACIÓN WEB; GEOLOCALIZACIÓN; GOOGLE API; MVC; AZURE; SENDGRID; BLOB STORAGE; C#	
RESUMEN:	<p>Este trabajo de titulación está basado en el desarrollo e implementación de una aplicación web para la creación de eventos que se desarrollen en el campus universitario de la UCSG con geolocalización aplicado a la facultad de ingeniería y la generación de alertas para que los usuarios estén pendientes sobre el desarrollo de las actividades que se registren en el sistema. Luego de analizar los diversos proveedores de geolocalización se decidió utilizar el api de Google y se optó por una aplicación web por la compatibilidad para ejecutarse sobre cualquier dispositivo que tenga un navegador web y una conexión a internet. Este proyecto tuvo un enfoque de investigación cualitativa, se realizó levantamiento de información observando el procedimiento para la difusión de eventos en la facultad y por medio de entrevistas al personal administrativo se identificaron las necesidades y los canales de comunicación actual. Se realizó el diseño, desarrollo e implementación del sistema web de difusión de eventos el cual se encuentra publicado totalmente en la nube, para la constante comunicación entre la comunidad universitaria.</p>	
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO
CONTACTO CON AUTOR:	Teléfono: +593-4-2821483 / 0962763883	E-mail: rodrigomedinag@hotmail.com
CONTACTO CON LA INSTITUCIÓN:	Nombre: Ing. Edison José Toala Quimí	
	Teléfono: +593-042 20 27 63 / 593-9-90976776	
	E-mail: edison.toala@cu.ucsg.edu.ec	

SECCIÓN PARA USO DE BIBLIOTECA	
Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	