



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLÁN**

CREACIÓN DE APLICACIÓN MÓVIL PARA LA PUBLICACIÓN DE
EVENTOS CULTURALES.

Tesis

QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTA:

HERNÁNDEZ CHÁVEZ JOSÉ DE JESÚS

ASESOR:

L.C. CARLOS PINEDA MUÑOZ

CUAUTITLÁN IZCALLI, ESTADO DE MÉXICO, 2018



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES CUAUTITLÁN
UNIDAD DE ADMINISTRACIÓN ESCOLAR
DEPARTAMENTO DE EXÁMENES PROFESIONALES

U. N. A. M.
FACULTAD DE ESTUDIOS
SUPERIORES CUAUTITLÁN

ASUNTO: VOTO APROBATORIO

M. en C. JORGE ALFREDO CUÉLLAR ORDAZ
DIRECTOR DE LA FES CUAUTITLÁN
PRESENTE

ATN: I.A. LAURA MARGARITA CORTEZ FIGUEROA
Jefa del Departamento de Exámenes Profesionales



DEPARTAMENTO DE EXÁMENES PROFESIONALES
de la FES Cuautitlán.

Con base en el Reglamento General de Exámenes, y la Dirección de la Facultad, nos permitimos comunicar a usted que revisamos el: **Trabajo de Tesis**

Creación de aplicación móvil para la publicación de eventos culturales

Que presenta el pasante: **JOSÉ DE JESÚS HERNÁNDEZ CHÁVEZ**

Con número de cuenta: **30708421-5** para obtener el Título de la carrera: **Licenciatura en Informática**

Considerando que dicho trabajo reúne los requisitos necesarios para ser discutido en el **EXAMEN PROFESIONAL** correspondiente, otorgamos nuestro **VOTO APROBATORIO**.

ATENTAMENTE

"POR MI RAZA HABLARÁ EL ESPÍRITU"

Cuautitlán Izcalli, Méx. a 24 de mayo de 2018.

PROFESORES QUE INTEGRAN EL JURADO

	NOMBRE	FIRMA
PRESIDENTE	L.C. Carlos Pineda Muñoz	
VOCAL	L.I. Oscar Vilchis Guerra	
SECRETARIO	L.I. Mauricio Jaques Soto	
1er. SUPLENTE	L.I. Elizabeth Barrera Romero	
2do. SUPLENTE	L.I. Maricela Lara Martínez	

NOTA: los sinodales suplentes están obligados a presentarse el día y hora del Examen Profesional (art. 127).

LMCF/ntm*

DEDICATORIAS

A MI MADRE TERESA.

POR HABERME APOYADO EN TODO MOMENTO, POR SUS CONSEJOS, SUS VALORES, POR LA MOTIVACIÓN CONSTANTE QUE ME HA PERMITIDO SER UNA PERSONA DE BIEN, PERO MÁS QUE NADA, POR SU AMOR.

A MI PADRE LUIS.

POR LOS EJEMPLOS DE PERSEVERANCIA Y CONSTANCIA QUE LO CARACTERIZAN Y QUE ME HA INFUNDADO SIEMPRE, POR EL VALOR MOSTRADO PARA SALIR ADELANTE Y POR SU AMOR.

A MI ASESOR.

POR SU GRAN APOYO Y MOTIVACIÓN PARA LA CULMINACIÓN DE NUESTROS ESTUDIOS PROFESIONALES Y PARA LA ELABORACIÓN DE ESTA TESIS

A MIS PROFESORES

A LA LIC. MARICELA LARA MARTINEZ Y ELIZABETH BARRERA ROMERO POR SU APOYO OFRECIDO EN ESTE TRABAJO, AL LIC OSCAR VILCHIS GUERRA POR SU TIEMPO COMPARTIDO Y POR IMPULSAR EL DESARROLLO DE NUESTRA FORMACIÓN PROFESIONAL Y AL PROFESOR MAURICIO JAQUES SOTO POR SU APOYO EN SU MOMENTO.

ÍNDICE

ÍNDICE DE FIGURAS	3
ÍNDICE DE TABLAS	4
INTRODUCCIÓN	5
OBJETIVOS	7
GENERAL	7
ESPECÍFICOS	7
HIPÓTESIS	7
<i>Alternativa</i>	7
<i>Nula</i>	7
METODOLOGÍA EMPLEADA	8
CAPITULO 1: MARCO TEÓRICO	9
1.1 EL MODELADO.....	9
1.2 Modelado BPMN.....	11
1.2.1 Explicación de la notación BPMN	11
1.3 METODOLOGÍA DE DESARROLLO ÁGIL	12
1.4 SCRUM	14
1.5 ARQUITECTURA DE SOFTWARE	16
1.5.1 Arquitecturas mas comunes	17
1.6 EL SERVIDOR.....	19
1.7 MODELO DE BASE DE DATOS.	21
1.7.1 Modelo relacional.	21
1.7.2 Modelo jerárquico	22
1.7.3 Modelo de base de datos orientado a objetos.....	23
1.7.5 Modelo entidad-relación	25
1.8 FASES DE DISEÑO DE UNA BASE DE DATOS	26
1.8.1 Diseño conceptual.....	26
1.8.2 Modelo entidad-relación	26
1.8.3 Diseño lógico.....	27
1.8.4 Diseño físico.....	27
1.9 GESTOR DE BASE DE DATOS	28
1.10 PARADIGMAS DE PROGRAMACIÓN	28
1.10.1 Ventajas y desventajas de la programación orientada a objetos.....	29
1.11 LOS LENGUAJES DE PROGRAMACIÓN.....	30
1.11.1 Frontend y Backend	31
1.12 LENGUAJES PARA EL DESARROLLO WEB	31
1.13 EL API	33
1.14 ¿POR QUÉ UTILIZAR UNA APLICACIÓN MÓVIL?	33
1.14.1 Sistemas operativos móviles.....	34
1.14.2 Tipos de sistemas operativos móviles.....	35
1.14.3 Tipos de aplicación móvil.....	35
1.14.4 La versión adecuada	36
1.15 LENGUAJES PARA EL DESARROLLO DE LA APLICACIÓN MÓVIL	37
1.16 FRAMEWORK	40

1.17 LICENCIAS DE SOFTWARE	40
1.18 TIPOS DE LICENCIAS	41
1.19 POLÍTICAS DE PRIVACIDAD Y MANEJO DE DATOS PERSONALES.....	43
CAPÍTULO 2: ANTECEDENTES	44
2.1 ANÁLISIS DEL PROBLEMA	44
2.2 LA PROPUESTA	44
2.3 JUSTIFICACIÓN.....	45
2.4 DEFINICIÓN Y SU ALCANCE DE LA PROPUESTA	45
2.5 PLANIFICACIÓN DEL SISTEMA.....	48
2.6 EL DIAGRAMA BPMN.....	48
2.6.1 <i>Explicación del diagrama BPMN:</i>	50
2.6.2 <i>Las tareas</i>	56
2.7 MODELADO DE BASE DE DATOS CONCEPTUAL	57
2.7.1 <i>Explicación de la creación de los datos en el modelo relacional.</i>	58
2.8 DISEÑO FÍSICO.....	60
2.9 DICCIONARIO DE DATOS	61
2.10 SCRUM	64
2.11 LICENCIA DE SOFTWARE	68
2.12 POLÍTICA DE PRIVACIDAD Y MANEJO DE DATOS PERSONALES	68
2.13 REQUERIMIENTOS DE SOFTWARE PARA EL DESARROLLO DEL SOFTWARE.	70
2.14 LIMITACIONES DEL SISTEMA	71
2.15 REQUERIMIENTOS DE HARDWARE Y SOFTWARE PARA IMPLEMENTAR LA PLATAFORMA WEB Y LA APLICACIÓN MÓVIL.	72
CAPÍTULO 3: DESARROLLO E IMPLEMENTACIÓN.....	73
3.1 INTERFACES DE CONEXIÓN	73
3.2 CREACIÓN DE LA PLATAFORMA WEB	75
3.2.1 <i>Descripción de módulos de la plataforma web</i>	75
3.3 CREACIÓN DEL API.....	75
3.4 CREACIÓN DE LA APLICACIÓN MÓVIL.....	76
3.4.1 <i>Descripción de módulos de la aplicación móvil</i>	76
3.5 NARRATIVA DE IMPLEMENTACIÓN DE SISTEMA	77
3.6 IMPLEMENTACIÓN DE LA PLATAFORMA WEB	77
3.7 PUBLICACIÓN DE APLICACIÓN MÓVIL.	77
3.7.1 <i>Firma de aplicación</i>	78
3.7.2 <i>La consola de desarrollador</i>	79
3.7.3 <i>Versiones del app</i>	80
3.7.4 <i>Ficha de Play Store</i>	81
3.7.5 <i>Clasificación de contenido</i>	81
3.7.6 <i>Precios y distribución.</i>	81
3.7.7 <i>Aplicación en producción</i>	81
CONCLUSIONES	82
RECOMENDACIONES DE MEJORA	83
BIBLIOGRAFÍA	84
ANEXOS.....	86
EXPLICACIÓN DEL FUNCIONAMIENTO DE LA APLICACIÓN MÓVIL	86

SPLASH SCREEN	86
VERIFICAR CREDENCIALES	87
LOGIN	87
MÓDULO DE SISTEMA DE REGISTRO	88
RECUPERACIÓN DE ACCESO:	89
MENÚ PRINCIPAL.....	91
MODULO PERFIL.....	92
MÓDULO EVENTOS.	94
EVENTOS:	95
MODULO AVISOS	96
OPCIÓN SALIR	97
EXPLICACIÓN DEL FUNCIONAMIENTO DE LA PLATAFORMA WEB	98
SISTEMA DE ACCESO	98
ACCESO	98
RECUPERACIÓN DE ACCESO	99
EL MENÚ PRINCIPAL	100
<i>Página inicial del sistema</i>	<i>100</i>
MÓDULO USUARIO	101
ACTUALIZAR ACCESO:	101
MÓDULO EVENTOS.	102
OPCIÓN CREAR CATEGORÍA.....	102
OPCIÓN EVENTOS	104
OPCIÓN EDITAR	105
EDITAR CATEGORÍA:	106
EDITAR EVENTO.....	108
MÓDULO AVISOS	111
INFORMACIÓN LEGAL PROPIEDAD INTELECTUAL Y MARCAS REGISTRADAS	112

Índice de figuras

<i>Tabla 1 Explicación de notación</i>	11
<i>Tabla 2 Diferencias entre metodologías ágiles y no ágiles</i>	13
<i>Tabla 3 Explicación de carril cliente</i>	50
<i>Tabla 4 Explicación de diagrama Administrador</i>	53
<i>Tabla 5 Explicación de datos de usuario administrador</i>	58
<i>Tabla 6 Explicación de los datos de usuario cliente</i>	59
<i>Tabla 7 Datos de usuario</i>	61
<i>Tabla 8 Datos de acceso de usuario administrador</i>	61
<i>Tabla 9 Elementos de la tabla categoría evento</i>	62
<i>Tabla 10 Datos del evento</i>	62
<i>Tabla 11 Datos de los avisos de eventos</i>	63
<i>Tabla 12 Datos de registro de usuario a eventos y categoría</i>	63
<i>Tabla 13 Datos de perfil de usuario</i>	63
<i>Tabla 14 Back log (Lista de tarea) plataforma web</i>	65
<i>Tabla 15 Reordenamiento de tareas para la plataforma web</i>	66

Tabla 16 Back log (Lista de tarea) Aplicación móvil	66
Tabla 17 Reordenamiento de tareas para la aplicación móvil	67

Índice de tablas

Figura 1Ejemplo de diagrama BPMN	10
Figura 2 Datos recopilados durante un período de 7 días del 3/5/2018 hasta 10/5/2018.	37
Figura 3Representación del funcionamiento del API.....	47
Figura 4 Diagrama BPMN	49
Figura 5 Modelo de base de datos relacional.....	57
Figura 6 Símbolo de licencia de creativa comámonos.....	68
Figura 7 Configuración de conexión a la base de datos	73
Figura 8 Interfaz de acceso al API de la aplicación móvil	74
Figura 9 Explicación de como colocar la plataforma web en la carpeta raíz del servidor.....	77
Figura 10 ejemplo de creación de nueva llave	78
Figura 11 Creación de archivo apk y key	79
Figura 12 Configuración de la aplicación móvil en la consola	80
Figura 13 Imagen del departamento de difusión cultural	86
Figura 14 Módulo de verificación de credenciales	87
Figura 15 Módulo de sistema de registro.....	88
Figura 16 Opción para recuperar el acceso desde la aplicación.....	89
Figura 17 Muestra de envío de correo con datos de acceso al usuario desde la aplicación.....	90
Figura 18 Pantalla del menú principal.....	91
Figura 19 Modulo perfil.....	92
Figura 20 Módulo perfil.....	93
Figura 21 Módulo Evento (Categoría)	94
Figura 22 Módulo Eventos (evento)	95
Figura 23 Aviso de categoría sin eventos	95
Figura 24 Módulo avisos	96
Figura 25 muestra del botón salir	97
Figura 26 Pantalla principal de acceso.....	98
Figura 27 Diálogo al presionar el botón para recuperar acceso al sistema	99
Figura 28 Página inicial del sistema	100
Figura 29 opción editar usuario	101
Figura 30 Pantalla del módulo Eventos.....	103
Figura 31 Opción editar.....	105
Figura 32 diálogo al presionar el botón Editar categoría.....	106
Figura 33 Botón eliminar categoría.....	107
Figura 34 Diálogo al presionar el botón editar evento.....	108
Figura 35 Módulo avisos	111

Introducción

El desarrollo del trabajo propone la creación de una propuesta aplicando un sistema de información para el Departamento de Difusión Cultural utilizando tecnologías web y móvil con el objetivo de mejorar el flujo de información entre los miembros de la comunidad y el departamento, mediante el uso de estas herramientas.

El trabajo se encuentra dividido en 3 capítulos, en el primer capítulo se describirá los antecedentes teóricos para desarrollar el proyecto comparando ventajas y desventajas entre diversas metodologías, estructuras de modelos de datos, tecnologías, además se revisarán los requisitos legales como son los tipos de licencia y las políticas de uso de datos, este capítulo se enfoca para determinar los requisitos mas apropiados para la construcción de un sistema informático que pueda ser eficiente, escalable y funcional.

En el segundo capítulo se abordarán los motivos para llevar acabo a la creación de este proyecto, además de explicar las percepciones de alcance y conceptualización que se tienen para definir el funcionamiento operativo del sistema. Con la aplicación del diagramado BPMN permitiendo visualizar de manera mas precisa el funcionamiento de cada uno de los distintos tipos de roles de usuarios y la

interacción entre ellos, además de permitir la definición del modelado de datos y funciones que serán utilizados en el sistema.

.

También se explicará como se lleva a cabo la metodología de desarrollo ágil Scrum para crear el desarrollo del software. Así mismo se definirá el uso de la licencia para el proyecto y la política de uso de datos en personales, que se utilizará.

En el último capítulo se describirá el proceso de desarrollo e implementación de las plataformas web y la descripción del proceso para realizar la publicación de la aplicación móvil en la tienda, también se explicará como se lleva a cabo la

comunicación entre ambas plataformas utilizando un API y la funcionalidad de los módulos del software .

Al final del documento se podrá encontrar en el anexo el funcionamiento de la plataforma web y la aplicación móvil de forma gráfica.

.

Objetivos

General

Desarrollar una aplicación móvil para el departamento de difusión cultural que apoye a la promoción de eventos culturales en la FES Cuautitlán.

Específicos

- Desarrollar una propuesta permita atender las necesidades de difusión.
- Aplicar la metodología ágil Scrum para el desarrollo del software.
- Crear una plataforma web que sirva para llevar acabo la gestión y la publicación de los eventos culturales.
- Describir los procesos de desarrollo e implementación de la plataforma web y la aplicación móvil.
-

Hipótesis

Alternativa

Desarrollar un sistema informático utilizando tecnología móvil mediante el uso de metodologías de desarrollo, puede apoyar a la difusión de eventos culturales.

Nula

Desarrollar e implementar un sistema con tecnología móvil, mediante el uso de metodologías de desarrollo no simplifica ni mejora la difusión de eventos culturales.

Metodología empleada

El proyecto utilizará una metodología de desarrollo ágil Scrum guiando la construcción del proyecto paso a paso ya que debido a la información recabada en la investigación se ajusta a las características para hacer uso de ella en cada una de las fases que lo componen. Utilizando el modelado de procesos BPMN

podremos modelar los procedimientos internos de forma gráfica permitiéndonos observar el funcionamiento del software de manera concisa.

Una vez identificado el flujo de procesos y de información en el modelado del sistema, se realizó una investigación sobre los modelos de datos, arquitectura, tecnologías y paradigmas más adecuados para este proyecto. Para ello se utilizó una arquitectura cliente servidor, utilizando un modelo de datos entidad relación. Para el almacenamiento de datos se optó por el uso de un modelo de base de datos relacional ya que presentan una gran facilidad al momento de crearlas y recuperar datos al momento de realizar consultas.

Teniendo en cuenta las características se investigó sobre las herramientas más adecuadas para el desarrollo de aplicaciones destacando sus ventajas y desventajas se decidió por utilizar los lenguajes web HTML, PHP, CSS, JavaScript, JSON.

En la investigación también se comparó los distintos tipos de sistemas operativos móviles y aplicaciones para determinar cuál era el más adecuado, por ello se determinó en crear una aplicación nativa en el sistema operativo Android en su versión 4.1, utilizando un entorno de desarrollo especializado utilizando los lenguajes Java y XML.

Capítulo 1: Marco teórico

El departamento de difusión cultural de la FES Cuautitlán se encarga de promover el acercamiento de las diferentes expresiones culturales y artísticas a su comunidad,

para ello recurre al uso de medios como pancartas, folletos, un sitio web y otros medios para llegar a la mayor parte de la comunidad.

A pesar del gran trabajo del departamento y contar con diferentes recursos para promover los eventos culturales, la falta de comunicación y la actualización regular de la información genera un gran problema para la gente interesada en participar en los eventos.

El problema representa una oportunidad importante para la implementación y uso de un sistema móvil, que permitirá atender el inconvenientes con la recepción de la información oportuna y rápida

Para ello se establecerá una propuesta del sistema a partir del modelado a través de un diagrama, para ello se recurrirá a diversas metodologías de diseño que permitan establecer el funcionamiento operativo.

1.1 El modelado

Es una técnica para tratar con la complejidad inherente a los sistemas. El uso de modelos ayuda a los programadores de software a "visualizar" el sistema a construir.

Entre los cuales podemos encontrar:

UML: El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc. (Orallo (S.F))

BPMN: "Es una notación a través de la cual se expresan los procesos de negocio en un diagrama de procesos de negocio" (ANALITICA S.F, 4).

BPMN es el enfoque más sólido para modelar los procesos de negocio de una organización. Como lenguaje de modelado, es hoy el estándar de la industria informática, en particular adoptado por las industrias de bancos y seguros . (adopción 2016)

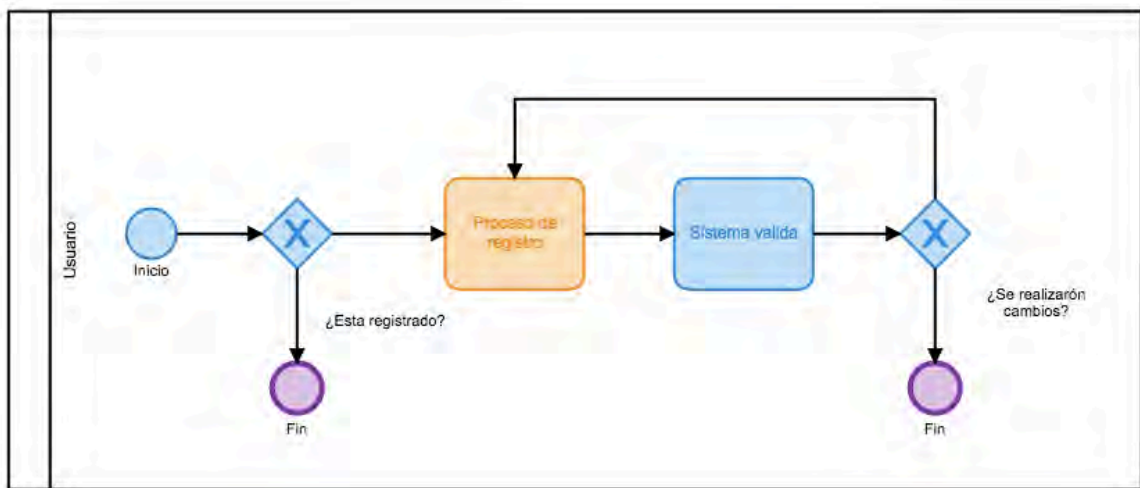


Figura 1Ejemplo de diagrama BPMN

A diferencia del modelado con UML, “el 80% de los errores puede modelarse usando del 20% del UML” (modelado (S.F))

1.2 Modelado BPMN




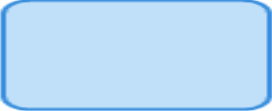


El modelado de datos iniciara con un evento de inicio este puede estar fuera o dentro del LANE esto dependerá de la concepción del proceso de negocios o participantes que se requiera, a partir de este punto se pude construir el flujo del

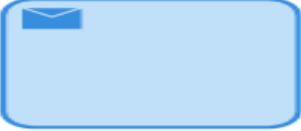

modelo de procesos utilizando las diferentes notaciones que se requieran finalizando la continuidad del o los proceso con el símbolo fin.

1.2.1 Explicación de la notación BPMN

Para comprender mejor el diagrama BPMN a continuación se explica la notación que se utilizará.

Tabla 1 Explicación de notación

Notación	Descripción
	<p>Inicio: Representa el inicio del proceso</p>
	<p>Fin: Indica cuando finaliza un proceso</p>
	<p>Proceso: Es un tarea donde interviene un humano para su ejecución y presenta información para la ejecución de la tarea.</p>
	<p>Servicio: Son todas aquellas tareas que realiza el sistema sin intervención humana, como lo puede ser: enviar un correo o invocar un servicio web.</p>
	<p>Secuencia: Muestra el orden de los eventos, actividades y decisiones que se realizan dentro del proceso.</p>
	<p>Compuerta exclusiva divergente: son decisiones que toma el usuario o el sistema para decir el camino a seguir.</p>

	<p>Envió de mensaje: Notificación de envío de un mensaje al usuario</p>
	<p>LANE: Representa un participante dentro un proceso, el cual contiene un conjunto de actividades asociadas a este rol.</p>

1.3 Metodología de desarrollo ágil

“En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera” (Grupo ISSI 2003).

A continuación se muestra una tabla con las principales diferencias entre metodologías ágiles y las tradicionales.

Tabla 2 Diferencias entre metodologías ágiles y no ágiles

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo

Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños	Grupos grandes y posiblemente distribuidos
Menos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Según Gonzalo Gabino Diestro(S.F) Las metodologías ágiles más usadas son:

SCRUM. Es un marco de trabajo que nos proporciona una serie de herramientas y roles para, de una forma iterativa, poder ver el progreso y los resultados de un proyecto.

KANBAN. Se basa en una idea muy simple. Ésta es que el trabajo en curso (Work In Progress, WIP) debería limitarse y sólo deberíamos empezar con algo nuevo cuando un bloque de trabajo anterior haya sido entregado o ha pasado a otra función posterior de la cadena.

eXtreme Programming (XP): Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software,

promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo.

1.4 Scrum

Creado por Ken Schwaber y Jeff Sutherland “Scrum es una metodología de desarrollo ágil que simplifica el desarrollo y la interacción con el cliente o usuario final al tener ciclos cortos de entrega de pequeñas funciones del software” (Gallego (S.F)).

Esta metodología esta compuesta de la siguiente manera:

Historias de usuario: lo que el cliente o el usuario quiere que se implemente; es decir, son una descripción breve, de una funcionalidad software tal y como la percibe el usuario.

Product backlog: es el conjunto de las historias del usuario, el alcance del producto.

Sprint: ciclo corto en el que se entrega un prototipo a usuario Sprint backlog: las historias de usuario que entran en el sprint.

El “product owner” (o propietario del producto) es aquella persona con una visión muy clara del producto que se quiere desarrollar, que es capaz de transmitir esa visión al equipo de desarrollo y, además, está altamente disponible.

Scrum se basa en tres pilares:

- Transparencia, potenciando la comunicación. Utilizando pizarras para mostrar toda la información
- Inspección, se debe controlar el avance, las historias de usuario, el equipo...
- Revisión o retrospectivas, que nos den feedback después de cada sprint

Todo equipo Scrum tiene tres roles principales:

1. Scrum Master: es el responsable de asegurar que el equipo Scrum siga las prácticas de Scrum
2. Propietario del Producto (ProductOwner)
3. Equipo de desarrollo, con las siguientes características: • Auto-gestionado • Multifuncional • No-distribuido • Tamaño óptimo: mínimo 3 personas

Una vez identificado el product backlog, el product owner ordena las historias de usuario. Se determina la duración del sprint, entre 2 y 4 semanas. Del product Backlog, se escoge una serie de historias que formarán parte del sprint backlog • Para estimar las historias de usuario es el Planning Poker • Aunque teóricamente cada sprint conlleva una entrega en producción, en la actualidad se marcan unos sprints, como sprint de release que serán los que finalmente sí lleguen a producción. El Product Owner puede visualizar, pero no puede modificar el Sprint Backlog. En cambio, puede modificar el ProductBacklog cuantas veces quiera con la única restricción de que los cambios tendrán efecto una vez finalizado el Sprint.

Ventajas teóricas: Alcance totalmente abierto. Por lo tanto, gran capacidad de adaptación al cambio. Planificación de sprints cortos con prototipos en producción. Por lo tanto, el coste se puede asociar a cada sprint. Colaboración estrecha e intensa con el cliente (product owner, retrospectivas) Producto completamente adaptado a requisitos de negocio en la entrega.

A pesar de ser una de las metodologías más usadas en el mundo del desarrollo del software esta también tiene sus limitaciones como son

- **Funciona sobre todo con equipos reducidos.** Las empresas grandes, por ejemplo, deben estar sectorizadas o divididas en grupos con objetivos concretos. De lo contrario, el efecto de la técnica se perderá.
- **Requiere una exhaustiva definición de las tareas y sus plazos.** Cuando estos dos aspectos no se definen adecuadamente, Scrum se desvanece. Recuerda que la división del trabajo en iteraciones (y de éstas en tareas específicas) son la esencia de esta metodología.
- **Exige una alta cualificación o formación.** No es una modalidad de gestión propia de grupos junior o que apenas estén en proceso de formación. Gran parte del éxito de Scrum radica en la experiencia que aportan los profesionales de los equipos, quienes por lo general acumulan años de experiencia.

1.5 Arquitectura de software

“La Arquitectura de Software es a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema.” (Len Bass 2011)

En términos simples una arquitectura de software es un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

1.5.1 Arquitecturas mas comunes

Monolítica: Esta elaborada bajo el supuesto de que será ejecutada en un único equipo de cómputo. Pero este tipo de aplicación también puede ser implementada

en un computador central o mainframe. Desde una terminal de servicio se puede monitorear y recibir los resultados que la aplicación monolítica devuelva.

Ventajas:

Eficiencia, ya que se producen pocos cambios.

Desventajas:

- Difícil de depurar
- Difícil de ampliar
- Difícil de distribuir
- Difícil de implantar.

Cliente-servidor: Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

Ventajas:

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red.

Desventajas:

- La congestión del tráfico ha sido siempre un problema de la arquitectura. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un computador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el costo.

Arquitectura de tres niveles: Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

Ventajas:

- Las llamadas de la interfaz del usuario en la estación de trabajo, al servidor de capa intermedia, son más flexibles que en el diseño de dos capas, ya que la estación solo necesita transferir parámetros a la capa intermedia.
- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario en la PC.
- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular.

- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.

Desventajas

- Los ambientes de tres capas pueden incrementar el tráfico en la red y requiere más balance de carga u tolerancia a las fallas.
- Los exploradores actuales no son todos iguales.
- La estandarización entre diferentes proveedores ha sido lenta en desarrollarse. Muchas organizaciones son forzadas a escoger uno en lugar de otro, mientras que cada uno ofrece sus propias y distintas ventajas.

A pesar de las grandes diferencias entre una y otra arquitectura, esta claro que la arquitectura cliente servidor es la ideal para el proyecto ya que por su versatilidad cumple con los elemento que se requieren para desarrollar el proyecto.

1.6 El Servidor

Manuel Sierra García hace la siguiente definición “Un servidor es un programa y/o computadora que esta al “servicio” de otras computadoras, maquinas o personas llamadas clientes y le suministra a estos, todo tipo de información”. (García 2009)

Existen muchos tipos de servidor según su función o su contenido:

Servidor de Correo: Es el servidor que almacena, envía, recibe y realiza todas las operaciones relacionadas con el e-mail de sus clientes.

Servidor Proxy: Es el servidor que actúa de intermediario de forma que el servidor que recibe una petición no conoce quién es el cliente que verdaderamente está detrás de esa petición.

Servidor Web: Almacena principalmente documentos HTML (son documentos a modo de archivos con un formato especial para la visualización de páginas web en los navegadores de los clientes), imágenes, videos, texto, presentaciones, y en general todo tipo de información. Además se encarga de enviar estas informaciones a los clientes.

Servidor de Base de Datos: Da servicios de almacenamiento y gestión de bases de datos a sus clientes. Una base de datos es un sistema que nos permite almacenar grandes cantidades de información. Por ejemplo, todos los datos de los clientes de un banco y sus movimientos en las cuentas.

Servidores Clúster: Son servidores especializados en el almacenamiento de la información teniendo grandes capacidades de almacenamiento y permitiendo evitar la pérdida de la información por problemas en otros servidores.

Servidores Dedicados: Como ya expresamos anteriormente, hay servidores compartidos si hay varias personas o empresas usando un mismo servidor, o dedicados que son exclusivos para una sola persona o empresa.

Servidores de imágenes: Recientemente también se han popularizado servidores especializados en imágenes, permitiendo alojar gran cantidad de imágenes sin consumir recursos de nuestro servidor web en almacenamiento o para almacenar fotografías personales, profesionales, etc.

A pesar de haber una gran cantidad de tipos de servidores, la decisión de requerir un servidor web es mas que claro ya que el modelo del proyecto se ajusta a los requerimientos para utilizar scriptst y base de datos, para crear el sistema.

1.7 Modelo de base de datos.

Uno de los pasos cruciales en la construcción de una aplicación que maneje una base de datos, es sin duda, el diseño de la base de datos, en donde lo más importante es el diseño que de ya que de no hacerlo correctamente puede generar problemas.

Los modelos de datos aportan la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y uso de sistemas de información.

Con respecto al diseño de bases de datos, el modelado de datos puede ser descrito así (Brodie 1984) "dados los requerimientos de información y proceso de una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos".

1.7.1 Modelo relacional.

Siendo el modelo más común, el modelo relacional ordena los datos en tablas, también conocidas como relaciones, cada una de las cuales se compone de columnas y filas. Cada columna enumera un atributo de la entidad en cuestión, por ejemplo, precio, código postal o fecha de nacimiento. En conjunto, a los atributos en una relación se los llama dominio. Se elige un atributo particular o combinación

de atributos como clave primaria, a la cual se puede hacer referencia en otras tablas, en donde se la denomina clave externa.

Cada fila, también denominada tupla, incluye datos sobre una instancia específica de la entidad en cuestión, por ejemplo, un empleado específico.

El modelo también representa los tipos de relaciones entre esas tablas, incluidas las relaciones uno a uno, uno a muchos y muchos a muchos.

Ventajas:

- Provee herramientas que garantizan evitar la duplicidad de registros.
- Garantiza la integridad referencial, así, al eliminar un registro elimina todos los registros relacionados dependientes.
- Favorece la normalización por ser más comprensible y aplicable.

Desventajas:

- Presentan deficiencias con datos gráficos.
- No se manipulan de forma eficiente los bloques de texto como tipo de dato.

1.7.2 Modelo jerárquico

El modelo jerárquico organiza los datos en una estructura de árbol, en la que cada registro tiene un único elemento o raíz. Los registros del mismo nivel se clasifican en un orden específico. Ese orden se usa a manera de orden físico para almacenar la base de datos. El modelo es bueno para describir muchas relaciones del mundo real.

Este modelo fue usado principalmente por los Sistemas de Gestión de Información de IBM en las décadas de 1960 y 1970, pero raramente se los ve en la actualidad debido a ciertas ineficiencias operativas.

Ventajas:

- Existen una reducción a las dependencias de datos.
- La predicción del funcionamiento se simplifica a través de las relaciones.
- Se puede definir relaciones y simplifica las variaciones a futuro.

Desventajas:

- * AL borrar un nodo padre se eliminan sus nodos subordinados
- * Solo se puede añadir un nodo hijo si existe un nodo padre.

1.7.3 Modelo de base de datos orientado a objetos

Una base de datos orientada a objetos es una base de datos donde los elementos son objetos. Estos pueden ser bases de datos multimedia (videos, imágenes y sonidos), donde la herencia nos permita una mejor representación de la información, estas bases de datos tienen una identidad de ser un Todo, y no solo una parte de una gran base, por ejemplo una base de secuencias de ADN.

El objetivo de una base de datos orientada a objetos son los mismos que los de las bases de datos tradicionales, pero con la ventaja de representar los modelos de datos con un marco mucho más eficiente, manteniendo la integridad y relación entre ellos.

Recordemos que un objeto es una estructura que tiene asociado un estado y un comportamiento (propiedades y métodos). Estas bases tienen las características de todo lo que es orientado a objeto que son Herencia, Polimorfismo, Abstracción y Encapsulamiento.

Ventajas:

- Modelo de objetos intuitivamente más cercano al mundo real.
- Extensibilidad - herencia.
- Valores complejos.
- Eliminación de la impedancia incorrecta.
- Lenguaje de consulta más expresivo
- El estrechamiento/acoplamiento entre datos y aplicaciones permite que el esquema capture más el significado de las aplicaciones.
- Soporte para transacciones largas.

Desventajas:

- La falta de un fundamento teórico, por lo que el significado exacto de un modelo de datos orientado a objetos no está bien definido.
- Con un modelo de datos orientado a objetos es más difícil conseguir personal experimentado.
- Falta de estándares.
- La competencia de los sistemas de gestión de base de datos relacionales y objeto-relacionales.
- La encapsulación está comprometida para optimizar las consultas.
- Un sistema de gestión de base de datos orientado a objetos, generalmente controla la concurrencia bloqueando. Bloquear una jerarquía de herencia es difícil y puede afectar al rendimiento.
- Un modelo de datos orientado a objetos es inherentemente más complejo que el modelo de datos relacional; el sistema de gestión de base de datos orientado a objetos proporciona más complejidad que el sistema de gestión de un modelo de datos relacional. La complejidad lleva a mayores costos de implementación y mantenimiento.
-

1.7.5 Modelo entidad-relación

Este modelo capta las relaciones entre entidades del mundo real de forma muy similar al modelo de red, pero no está directamente ligado a una estructura física de la base de datos. En cambio, con frecuencia se lo usa para diseñar una base de datos conceptualmente.

Aquí, a las personas, lugares y cosas, acerca de las cuales se almacenan puntos de datos, se las denomina entidades, cada una de las cuales tiene ciertos atributos que en conjunto forman su dominio. La cardinalidad, o relaciones entre entidades, también se representa en diagramas.

Ventajas:

- Comprensión de los datos de una organización y del funcionamiento de la organización.
- Obtención de estructuras de datos independientes del entorno físico.
- Control de los posibles errores desde el principio, o al menos, darse cuenta de las deficiencias lo antes posible.
- Mejora del mantenimiento.

Desventajas:

- Vulnerable a fallos
- Complejidad

1.8 Fases de diseño de una base de datos

Para David Durnning (Dunning 2001) existen 4 grandes fases que componen el diseño tradicional de bases de datos:

1.8.1 Diseño conceptual

El diseño conceptual incluye la creación de un esquema o modelo conceptual de la base de datos. Este modelo es independiente de las consideraciones físicas, incluyendo los sistemas de gestión de base de datos, los lenguajes de programación y las plataformas de hardware. Los usuarios no técnicos deben entender el esquema, así que no debería contener detalles sobre cómo se implementará la base de datos. Pero debe estar detallado en términos de la naturaleza, estructura y significado de los datos.

1.8.2 Modelo entidad-relación

Durante la fase conceptual, los diseñadores de bases de datos suelen crear lo que se conoce como el modelo entidad-relación, o diagrama, para ayudarles a visualizar la base de datos. Este diagrama identifica cada entidad, conocida también como relación o tabla en la base de datos, así como las relaciones entre las entidades. Básicamente, el diagrama es lo bastante sencillo como para permitir que los diseñadores de bases de datos aprendan y entiendan los conceptos básicos, pero lo bastante detallados, que les ayuden a desarrollar aplicaciones complejas.

1.8.3 Diseño lógico

El objetivo del diseño lógico es transformar el esquema genérico y conceptual en un modelo de datos determinado para un sistema de gestión de bases de datos determinado. El diseño lógico puede realizarse manualmente, o automáticamente en algunos casos, a través del uso de herramientas CASE (ingeniería de software asistida por computadora) desde un diseño conceptual. En cualquier caso, el resultado final es un conjunto de comandos de lenguaje de definición de datos, que puede ser usado de forma interactiva, o como parte de un programa de computadora para crear la base de datos.

1.8.4 Diseño físico

El diseño físico es el proceso de implementación física del modelo de datos lógico en un sistema de gestión de bases de datos. Incluye escoger las estructuras de archivos determinadas en las que almacenar las tablas, o relaciones, de la base de datos, y en asegurar que se puede acceder a las relaciones de forma rápida, eficiente y segura. Los diseñadores de bases de datos pueden necesitar pensar sobre los tiempos de acceso a la base de datos para las transacciones que se usan habitualmente, así como el número medio de transacciones procesadas por minuto y la cantidad de espacio que necesitará la base de datos. Un índice en unos campos, o columnas, determinados en una base de datos puede mejorar los tiempos de acceso, pero el diseñador de bases de datos debe decidir qué campos indexar.

1.9 Gestor de base de datos

“Un gestor de base de datos (DataBase Managenent System) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarios para el almacenamiento y búsqueda de la información del modo más eficiente posible.” (Iruela 2014)

Maria DB: Desarrollado Michael (Monty) Widenius (fundador de MySQL), Su velocidad es una de sus características más destacadas. MariaDB es notablemente escalable y puede manejar decenas de miles de tablas y miles de millones de filas de datos. También puede administrar pequeñas cantidades de datos de forma rápida y sin problemas, por lo que es conveniente para pequeñas empresas o proyectos personales. (Maria DB (S.F))

1.10 Paradigmas de programación

“Un paradigma de programación es un modelo conceptual para desarrollar programas. El uso de un paradigma se refuerza por el lenguaje que se escoja para realizar un programa concreto, aunque en general, con mayor o menor dificultad, se puede usar cualquier lenguaje de programación para seguir cualquier paradigma.”

- **Orientado al procedimiento.**- Que se expresa de manera imperativa en forma de algoritmos. Por ejemplo C y Fortran.
- **Orientado a funciones.**- Se basa en el concepto matemático de función y se expresa de manera declarativa. Por ejemplo Lisp, SML, Hope, Hasckel.
- **Orientado a la lógica.**- Que se expresa por metas en forma de cálculo de predicados. Utilizan reglas e inferencia lógica. Por ejemplo Prolog.

- **Orientado a objetos.**- Se expresa en forma de relaciones entre objetos. Por ejemplo php o Java.

Esta clasificación no es fija sino que algunos lenguajes pertenecen a varios paradigmas. Existen paradigmas más generales, como el paradigma imperativo o el estructurado, que engloban a otros. Además, continuamente aparecen nuevos paradigmas, como la orientación a aspectos que complementa a la programación orientada a objetos fomentando la separación de conceptos. Cada uno de estos paradigmas tiene ciertas ventajas. Así, el paradigma procedimental aporta ventajas cuando son tareas sencillas que se pueden describir con unos pocos pasos, o cuando es importante optimizar la velocidad de ejecución. El uso del paradigma orientado a la lógica facilita la implementación de sistemas expertos en los que se deba manejar una base de conocimiento. El paradigma funcional permite construir programas concisos, fáciles de probar y paralelizar, lo cual es muy adecuado para la prueba matemática de algoritmos y para programas en los que se requiera un alto grado de fiabilidad. El paradigma de programación orientada a objetos está demostrando su utilidad en una amplia variedad de problemas (interfaces gráficas, simuladores, aplicaciones ofimáticas, juegos, entre otros). Además, está demostrando que puede ser un marco de alto nivel ideal para integrar sistemas desarrollados siguiendo diferentes paradigmas

1.10.1 Ventajas y desventajas de la programación orientada a objetos

Ventajas:

- **Reusabilidad.** Cuando hemos diseñado adecuadamente las clases, se pueden usar en distintas partes del programa y en numerosos proyectos.
- **Mantenibilidad.** Debido a la sencillez para abstraer el problema, los programas orientados a objetos son más sencillos de leer y comprender, pues nos permiten ocultar detalles de implementación dejando visibles sólo aquellos detalles más relevantes.

- **Modificabilidad.** La facilidad de añadir, suprimir o modificar nuevos objetos nos permite hacer modificaciones de una forma muy sencilla.
- **Fiabilidad.** Al dividir el problema en partes más pequeñas podemos probarlas de manera independiente y aislar mucho más fácilmente los posibles errores que puedan surgir.

Desventajas:

- Cambio en la forma de pensar de la programación tradicional a la orientada a objetos.
- La ejecución de programas orientados a objetos es más lenta.
- La necesidad de utilizar bibliotecas de clases obliga a su aprendizaje y entrenamiento.

1.11 Los lenguajes de programación

Un lenguaje de programación es un lenguaje formal que especifica una serie de instrucciones para que una computadora produzca diversas clases de datos. Los lenguajes de programación pueden usarse para crear programas que pongan en práctica algoritmos específicos que controlen el comportamiento físico y lógico de una computadora.

“Existen dos tipos de lenguajes claramente diferenciados; los lenguajes de bajo nivel y los de alto nivel.

El ordenador sólo entiende un lenguaje conocido como código binario o código máquina, consistente en ceros y unos. Es decir, sólo utiliza 0 y 1 para codificar cualquier acción.

Los lenguajes más próximos a la arquitectura hardware se denominan lenguajes de bajo nivel y los que se encuentran más cercanos a los programadores y usuarios se denominan lenguajes de alto nivel. “ (Alvarez 2006)

1.11.1 Frontend y Backend

Son términos que se refieren a la separación de intereses entre una capa de presentación y una capa de acceso a datos, respectivamente

- Frontend: Son todas las tecnologías con las que interactúan con los usuarios como son: HTML ,JavaScript, CSS, XML, etc.
- Backend: Son todas las tecnologías encargadas de procesar la información, como el manejo y almacenamiento de datos como PHP, JAVA, etc.

1.12 Lenguajes para el desarrollo web

Son las tecnologías principales usadas en el Internet y sirve como columna vertebral de la mayoría de los sitios web.

PHP: Es un lenguaje con propósito de uso general para ser utilizado del lado del servidor, Creado originalmente por Rasmus Lerdorf en el año 1995.

Este lenguaje se caracteriza por:

- 1.- Lenguaje totalmente libre y abierto
- 2.- Curva de aprendizaje muy baja
- 3.- Los entornos de desarrollo son de rápida y fácil configuración
- 4.- Fácil despliegue de paquetes totalmente autoinstalables que integran PHP
- 5.- Fácil acceso a bases de datos

CSS: Es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios.

HTML: Es un lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad.

JavaScript: Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. Leer más sobre JavaScript.

JSON: Es una sintaxis para serializar objetos, arreglos, números, cadenas, booleanos y nulos. Está basado sobre sintaxis JavaScript pero es diferente a ella: algo JavaScript no es JSON y JSON no es JavaScript.

XML: (eXtensible Markup Language en español **Lenguaje de Marcado eXtensible**) es un lenguaje de marcado genérico especificado por la W3C. La industria de tecnologías de la información (*IT Industry*) utiliza muchos lenguajes de descripción de datos (*data-description language*) que están basados en XML.

1.13 El API

“Una API (siglas de ‘Application Programming Interface’) es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software.” (Merino 2014)

Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software (o de la infraestructura ya existente en otras plataformas) para no estar reinventando la rueda constantemente, reutilizando así código que se sabe que está probado y que funciona correctamente. En el caso de herramientas propietarias (es decir, que no sean de código abierto), son un modo de hacer saber a los programadores de otras aplicaciones cómo incorporar una funcionalidad concreta sin por ello tener que proporcionar información acerca de cómo se realiza internamente el proceso.

1.14 ¿Por qué utilizar una aplicación móvil?

Una de las ventajas más evidentes de su uso es la facilidad y rapidez con la que se accede a la información, ya que las aplicaciones están presentes en sus dispositivos en todo momento y no necesitan introducir datos en cada acceso.

La presencia continua de las aplicaciones en los dispositivos móviles permite a las marcas ganar en presencia y notoriedad frente a otras soluciones.

También se consigue esta presencia en las tiendas de aplicaciones de las plataformas móviles, garantizando la aparición de las marcas en las búsquedas orgánicas desde el dispositivo móvil.

Incluso la propia aplicación puede llegar a convertirse en un canal de venta.

1.14.1 Sistemas operativos móviles

“En el caso de los S.O móviles la diferencia radica en su orientación a la 'conectividad inalámbrica', los cuales a través de diversas funcionalidades y aplicaciones permiten la usabilidad de los dispositivos.” **(Corporación Colombia Digital 2012)**

Componentes:

La 'Capa' o 'máquina virtual' del Sistema Operativo Móvil está integrada de los siguientes elementos:

Kernel o Núcleo. Es el encargado de dar paso seguro a los distintos programas de acceso al hardware, así como la gestión de desarrollar procesos en el sistema de archivos y la memoria.

El middleware. Conjunto de módulos que integra las diversas aplicaciones que hacen parte de los dispositivos móviles como el sistema de mensajería y comunicaciones, códecs multimedia, etc.

Entorno de ejecución de aplicaciones. Este componente es un gestor de aplicaciones e interfaces programables que facilitan la creación de software.

Interfaz de usuario. Este elemento no es más que el medio o instrumento por el cual el usuario se comunica y relaciona con el dispositivo móvil, la cual incluye los menús, teclado, aplicaciones, listas, botones, gráficos, etc.

1.14.2 Tipos de sistemas operativos móviles

Android: Es un sistema operativo móvil. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró.

IOS: Es un sistema operativo móvil de la empresa Apple Inc. Inicialmente este sistema tuvo varios nombres pero no fue hasta el lanzamiento del iPhone OS tuvo lugar el 29 de junio de 2010 y no es hasta el lanzamiento del iPad - que también lo utiliza como sistema operativo que pasa a llamarse simplemente iOS.

“Según los datos recopilados por StatCounter, en 2017 Android fue el sistema operativo con la mayor participación de mercado a nivel mundial. Hablamos de 38.97% , contra 37.07% de Windows , que tiene el segundo lugar. Casi dos puntos porcentuales de desapego, en una pelea por el primer lugar que ve al sistema operativo de Google y Microsoft luchando solo, dado que el tercer lugar de iOS está lejos del 13.18% . Siguiendo OS X , sería mejor llamar a macOS , con 5.24%. Linux, si lo preguntaras, es solo 0.76%.” **(Solucioneswindroid 2018)**

1.14.3 Tipos de aplicación móvil

“Existen tres tipos principales de aplicaciones móviles, cada una de las cuales se caracteriza por sus distintas propiedades, limitaciones o proceso de programación.” (Zeledon (S.F))

Encontramos, por ejemplo, las Aplicaciones Nativas, aquellas que han sido desarrolladas con el software que ofrece cada sistema operativo. De esta forma, iOS, Android y Windows Phone tienen software distintos, comúnmente denominados SDK o Software Development Kits.

Las aplicaciones web, comúnmente llamadas “web apps” son construidas principalmente en los lenguajes HTML, JavaScript y CSS. A diferencia de las

nativas, este tipo de aplicaciones no emplean un SDK o Software Development Kit. Esto quiere decir que, como desarrollador de web apps, puedes programar en la plataforma que desees, independientemente del sistema operativo en el cual tu aplicación será utilizada. Esto evita también el tedioso proceso de desarrollar un código distinto para cada una de las plataformas o sistemas operativos.

Las aplicaciones Híbridas, las cuales se caracterizan por ser una combinación o, como su nombre lo indica, un “híbrido” entre las dos aplicaciones que hemos visto anteriormente. En cuestiones de diseño, desarrollo y programación, una aplicación híbrida será realizada a partir de HTML, JavaScript y CSS, -tal como las web apps; la diferencia radica en que una vez que la aplicación está finalizada en cuanto a su diseño y programación, se compilará de tal manera que el resultado final será muy similar a una aplicación nativa.

1.14.4 La versión adecuada

Para determinar la versión mas apropiada del proyecto utilizaremos en Android “las estadísticas de instalación de Google Play”, esta es una herramienta que podemos encontrar en el IDE creado para desarrollar aplicaciones para este sistema operativo.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.2%
4.2 Jelly Bean	17	96.0%
4.3 Jelly Bean	18	91.4%
4.4 KitKat	19	90.1%
5.0 Lollipop	21	71.3%
5.1 Lollipop	22	62.6%
6.0 Marshmallow	23	39.3%
7.0 Nougat	24	8.1%
7.1 Nougat	25	1.5%

Figura 2 Datos recopilados de las estadísticas de instalación de Google Play durante un período de 7 días del 3/5/2018 hasta 10/5/2018.

Por el contrario IOS al ser un entorno mas cerrado el IDE obliga al desarrollador programar sus aplicaciones en las versiones mas actuales por lo que genera mas problemas para los dispositivos que no pueden actualizar su sistema operativos y por ende el software no puede ser ejecutado correctamente.

A pesar de esta gran diferencia entre los diferentes tipos de aplicaciones, sistemas operativos, y versiones esta claro que se debe de tomar una decisión Android o IOS ya que esto tienen una mayor cobertura en la sociedad.

1.15 Lenguajes para el desarrollo de la aplicación móvil

JAVA: Es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas

aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

Ventajas: Es un lenguaje independiente de la plataforma, es decir, cualquier programa creado a través de Java podrá funcionar correctamente en ordenadores de todo tipo y con sistemas operativos distintos. Eso es un beneficio para los programadores, pues les facilita el trabajo ya que ya no se ven obligados a crear un programa diferente que se adapte a Windows, Linux, Mac.

Desventajas: Android acepta todas las funcionalidades de Java 7 y sólo un subconjunto de funciones de la versión 8. Aunque en futuras versiones de Android Studio se incluirán más funcionalidades de Java 8 y sin dependencia a una mínima versión de Android. Por lo que resulta difícil, acceder a las nuevas mejoras de Java en el desarrollo móvil. Por lo tanto, los desarrolladores están más limitados en conseguir una eficiencia a la hora de escribir aplicaciones en Android.

C#: es un lenguaje orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos.

Ventajas: La principal ventaja es su potencia como lenguaje, pero también su flexibilidad. Soporta la mayoría de paradigmas, destacando el paradigma funcional que combinado con el paradigma orientado a objetos hacen del lenguaje uno de los más potentes.

Desventajas: radica en su portabilidad y en la dificultad que trae un desarrollo completo de un producto software empleando este lenguaje. Java es un lenguaje que presenta una curva de aprendizaje menor que la de lenguajes como C# o Perl.

Kotlin: Kotlin es un lenguaje de programación de tipado estático que corre sobre la Máquina Virtual de Java y que también puede ser compilado a código fuente de JavaScript. Es desarrollado principalmente por JetBrains en sus oficinas de San Petersburgo.

Ventajas: Una aplicación de Kotlin es tan rápida como una equivalente de Java, Con el soporte de Kotlin para funciones en línea, el código que usa lambdas y a menudo se ejecuta incluso más rápido que el mismo código escrito en Java además de ser interoperable con Java, permitiendo utilizar todas las bibliotecas existentes de Android en una aplicación de Kotlin.

Desventajas: Kotlin no es un lenguaje muy difundido todavía, por lo que la comunidad de desarrolladores es escasa comparado con otros lenguajes bien consolidados, como Java. Por lo tanto, la resolución de dudas, documentación, tutoriales o información sobre Kotlin es mucho menor.

Swift: Es el lenguaje de programación de Apple. Éste fue presentado en la WWDC14 (“World Wide Developers Conference”) del año 2014.

Ventajas: Al igual que kotlin la curva de aprendizaje es menor además de ser un lenguaje exclusivo de la misma empresa.

Desventajas: A decir verdad por este lenguaje de programación se ve limitado su uso debido a que se requiere de un equipo de la misma empresa para poder ser utilizado.

Después de observar los diferentes lenguajes de programación orientados a objetos considerando sus ventajas y desventajas se determino que el uso de los lenguajes java es el ideal ya que su soporte aporta mas cualidades al proyecto que se desea construir.

1.16 Framework

“En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. “ (Gutiérrez (S.F))

1.17 Licencias de software

“Son básicamente, un contrato entre el autor del programa y el usuario en las que se establecen una serie de términos y cláusulas que el usuario debe cumplir para usar el software.” (BBVA 2004)

No es más que obtener la determinada licencia o autorización que le permita el uso legal de determinado programa, esta licencia es un documento bien sea electrónico, en papel original o número de serie autorizado por el autor.

1.18 Tipos de licencias

Licencias GPL: es una de la más utilizadas y se suele denominar como GNU GPL. Con esta licencia el desarrollador conserva los derechos de autor, pero permite su libre distribución, modificación y uso siempre y cuando, en el caso de que el software se modifique, el nuevo software que se desarrolle como resultado quede obligatoriamente con la misma licencia. Incluso si el software con licencia GPL solo fuera una parte de otro programa, este programa también tendría que mantener la licencia. Está considerada la primera licencia copyleft y, bajo esta filosofía, cualquier código fuente licenciado bajo GPL, debe estar disponible y accesible, para copias ilimitadas y a cualquier persona que lo solicite. De cara al usuario final, el software licenciado bajo GPL es totalmente gratuito, pudiendo pagar únicamente por gastos de copiado y distribución.

Licencia AGPL: se engloba dentro de las licencias destinadas a modificar el derecho de autor derivadas de GNU. La novedad de AGPL es que, aparte de las cláusulas propias de una GNU GPL, ésta obliga a que se distribuya el software que se destine a dar servicios a través de una red de ordenadores, es decir, si se quiere usar como parte del desarrollo de un nuevo software, éste quedaría obligado a su libre distribución.

Licencia BSD: es un buen ejemplo de una licencia permisiva que casi no impone condiciones sobre lo que un usuario puede hacer con el software. El software bajo esta licencia es la menos restrictiva para los desarrolladores, ya que, por ejemplo, el software puede ser vendido y no hay obligaciones de incluir el código fuente. Además, una aplicación licenciada con BSD permite que otras versiones puedan tener otros tipos de licencias, tanto libres como propietarias; un buen ejemplo de ello es el conocido sistema operativo Mac OS X, desarrollado bajo esta licencia. También, BSD permite el cobro por la distribución de objetos binarios.

Licencia Apache: el software bajo este tipo de licencia permite al usuario distribuirlo, modificarlo, y distribuir versiones modificadas de ese software pero debe conservar el copyright y el disclaimer. La licencia Apache no exige que las obras derivadas (las versiones modificadas) se distribuyan usando la misma licencia, ni siquiera que se tengan que distribuir como software libre, solo exige que se informe a los receptores que en la distribución se ha usado código con la licencia Apache. En este sentido, al crear nuevas piezas de software, los desarrolladores deben incluir dos archivos en el directorio principal de los paquetes de software redistribuidos: una copia de la licencia y un documento de texto que incluya los avisos obligatorios del software presente en la distribución.

Licencias Creative Commons: su definición se basa en cuatro condiciones

- ‘Atribución’, con la cual se puede distribuir, exhibir, representar siempre y cuando se reconozca y se cite a su autor
- ‘No comercial’, que no permite usar el software con fines comerciales
- ‘No derivadas’, con la cual no se puede modificar dicha obra
- ‘Compartir igual’, que incluye la creación de obras derivadas siempre que mantengan la licencia original.

Bajo las licencias Creative Commons, existen otros tipos de licencia las cuales se diferencian entre sí con el “juego” de la combinación de las cuatro condiciones que acabamos de ver.

Decidir la licencia adecuada puede llegar a implicar a la renuncia de algunos derechos reservados inherentes al derecho de autor, tales como el derecho de reproducción y el derecho de transformación que, por otra parte, puede contribuir a la divulgación de su trabajo y permitir el derecho de acceso por parte del público a sus obras.

1.19 Políticas de privacidad y manejo de datos personales

“Es una política que se encarga de cuidar la información confidencial registrada en cualquier base de datos. Esta ley permitirá que un tercero no pueda acceder a los datos.” (ADALID corp 2016)

Para realizar la política debemos acatar las instrucciones y requerimientos de las autoridades, para ello debemos acudir al DECRETO que se expido en el diario oficial de la federación el 26 de enero 2017 por el presidente de los Estados Unidos Mexicanos en la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados en el TÍTULO SEGUNDO PRINCIPIOS Y DEBERES Capítulo I y II, en estos artículos se establecen las condiciones con las cuales se debe de realizar la recolección de datos personales.

Capítulo 2: Antecedentes

El departamento de difusión cultural busca promover las diferentes expresiones culturales a toda su comunidad, para ello cuenta con diversos recursos para promover los diferentes eventos culturales alrededor de la FES Cuautitlán haciendo uso de pancartas, volantes, afiches, trípticos e incluso contar con su propio sitio web.

2.1 Análisis del problema

A pesar del esfuerzo realizado por el departamento para promover los diferentes eventos, la falta de comunicación y la actualización regular de la información de los diferentes medios utilizados para fomentar los eventos puede llegar a generar problemas provocando desorientación, descontento y falta de colaboración para futuros eventos.

2.2 La propuesta

La propuesta utilizada para atender la problemática por la falta de comunicación entre el departamento y la comunidad será orientada a través de un proyecto informático utilizando tecnología móvil ya que por su versatilidad y portabilidad la convierten en un canal de comunicación práctico e ideal para una comunidad tan activa.

2.3 Justificación

El motivo para desarrollar este proyecto versa en explicar el proceso de creación software detallando el uso de diferentes requisitos técnicos y de un análisis empleando metodologías de desarrollo ágil aplicados un sistema informático utilizando tecnologías web y móvil que servirá para que el departamento pueda mejorar la difusión de eventos culturales a la comunidad de la FES Cuautitlán.

El sistema beneficiará tanto la comunidad como los organizadores de los distintos eventos culturales al contar con un medio de difusión práctico, veloz, y a su vez permitiendo la captación de una mayor audiencia a los eventos.

2.4 Definición y su alcance de la propuesta

Para poder implementar el proyecto que permita mejorar la comunicación entre ambas partes, se contempla la creación de 2 diferentes plataformas, un sistema web que será utilizado únicamente por un usuario administrador y una aplicación móvil que será manejada por los usuarios cliente, cada una de las diferentes plataformas cumplirá con diferentes funcionalidades las cuales se explicarán a continuación:

El administrador podrá realizar las siguientes acciones

- Acceder al sistema web con un usuario y contraseña que podrá cambiar en cualquier momento. Este módulo será creado para con el fin de mejorar la seguridad del usuario que intente acceder a la plataforma web
- Mandar avisos acerca de los eventos. Este módulo s

- Gestionar la creación de eventos, a través de la creación de categorías .

Limitaciones del rol administrador:

- No se podrá crear nuevos usuarios administrador.
- Los cambios serán totales e inmediatos al eliminar o cambiar datos.

El cliente

- Podrá acceder a la aplicación móvil mediante el registro o el acceso por medio de su usuario y contraseña.
- Visualizará el contenido de las diferentes categorías y los eventos disponibles.
- Acceder a una sección de avisos
- Podrá actualizar la información de su perfil.

Limitaciones del rol cliente:

- No se podrá recibir o enviar información si no se cuenta con una red.
- Se requiere de un dispositivo móvil con una versión Android 4.1 o superior.

Para la creación de los eventos en la plataforma web estos se encontrarán divididos en categorías con el fin de tener una mejor organización al momento de ser publicados en la aplicación móvil. Los datos de las categorías estarán enfocados a su descripción además de contar con una imagen, mientras que los datos de los eventos al acceder a través de las categorías buscarán ser

especificaos en cuanto a su descripción detallando elementos sobre la ubicación y la fecha del mismo.

Todo lo anteriormente mencionado podrá ser materializado por medio de la transmisión utilizando un API que es una interfaz que permite la interacción de datos entre la aplicación móvil y el sistema web como podemos observar en la siguiente figura.

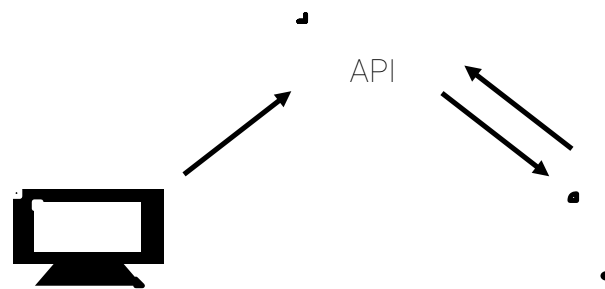


Figura 3 Representación del funcionamiento del API

2.5 Planificación del sistema

Tras analizar la problemática y la propuesta empleada es momento de crear un plan para desarrollar el proyecto con la guía de las metodologías para ello se ha establecido 4 fases que el sistema deberá acatar, las cuales son:

- Primera fase: Se establecerá el funcionamiento del proyecto en un diagrama BPMN mostrando el funcionamiento operativo que el sistema deberá cumplir además de establecer los valores requeridos para el modelado de la base de datos en un diagrama entidad/relación.
- Segunda fase: Utilizando la metodología scrum guiando la elaboración del proyecto se establecerán las diferentes actividades que se realizaran durante el tiempo de creación del sistema.
- Tercera fase: Se establecerán los requisitos técnicos que serán utilizados para la creación del proyecto.
- Cuarta fase: Desarrollo de las plataformas.
- Quinta fase: implementación y publicación de aplicación móvil.

2.6 El diagrama BPMN

El diagrama BPMN nos permitirá ver de una manera sencilla el funcionamiento del sistema relacionando los procesos con las diferentes entidades, de esta manera podremos observar el funcionamiento del sistema de manera clara.

A continuación se muestra el diagrama BPMN que se estableció para el funcionamiento del sistema.

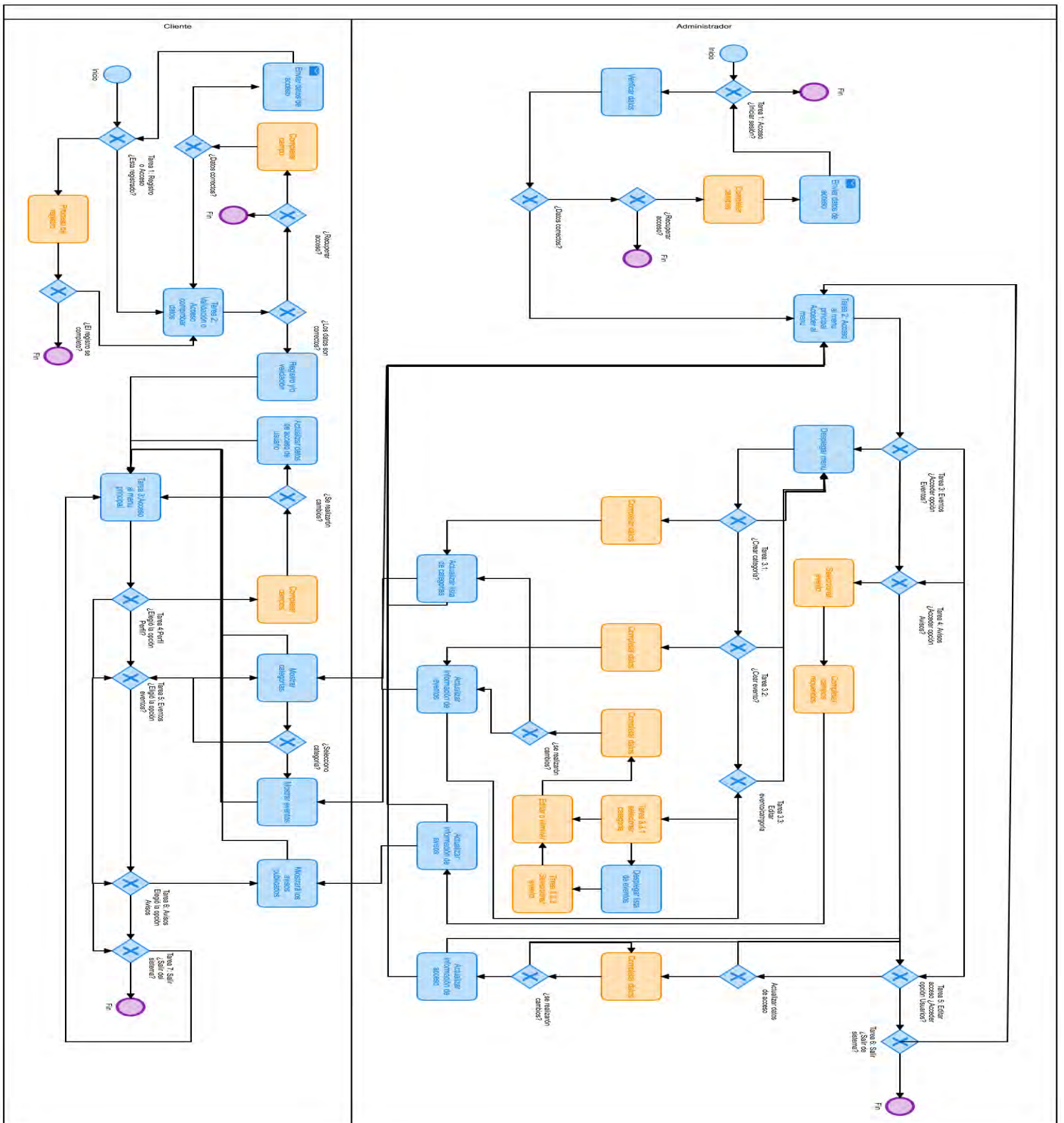


Figura 4 Diagrama BPMN

2.6.1 Explicación del diagrama BPMN:

El diagrama BPMN está dividido en 2 carriles, Cliente y Administrador cada uno de los diferentes roles converge de manera distinta pero a la vez de forma complementaria en algunas tareas el proceso inicia de izquierda a derecha de forma horizontal. En algunos casos el proceso no finalizara dentro de uno de los procesos si no que se tendrá la opción de mantenerse o continuar en una tarea diferente.

Cada una de las tareas en el diagrama esta limitada a su función dentro de su rol, esta acción es fundamental para elaborar la lista de prioridades.

A continuación en la siguiente tabla se explica cómo se ejecutan las tareas del usuario cliente en el diagrama BPMN además de citar los datos requeridos para cumplir con el flujo del proceso.

Tabla 3 Explicación de carril cliente

Usuario cliente		
Tarea 1: Registro o acceso		
Usuario, inicia ¿está registrado?		
Si	No	
Pasa a tarea 2	Inicia proceso de registro Datos requeridos: nombre, apellido, fecha de nacimiento, correo y contraseña.	
	Usuario, ¿Completo el registro correctamente?	
	Si	No
	Pasa a tarea 2	Fin
Tarea 2: Validación y/o Recuperación de acceso		
Sistema, revisa datos		
Sistema, ¿Los datos son correctos?		

Si		No	
Sistema, crea registro y/o validación de usuario		Usuario, ¿Recuperar acceso?	
		Si	No
Pasa a tarea 3		Usuario completa campos requeridos. Datos requeridos: correo	Fin
		Sistema, ¿Datos correctos?	
		Si	No
		Sistema envía datos de acceso regresa a tarea 1	Regresa a tarea 2 o Fin
Tarea 3, Acceso al menú principal			
Sistema, accede al menú principal			
Usuario, puede escoger tarea 4, 5, 6, o 7			
Tarea 4: Perfil			
Usuario, ¿escogió tarea 4?			
Si		No	
Usuario, ¿Actualizar datos?		Pasa a tarea 5	
Si	No		
Usuario, completa datos requeridos. Datos requeridos: nombre, apellido, fecha de nacimiento, correo y contraseña.	Regresa a tarea 3 o continúa en tarea 4.		

Sistema, ¿Se realizaron cambios?			
Si	No		
Sistema, actualiza datos de usuario	Regresa a tarea 3.		
Usuario, continua en tarea 4 o regresa a tarea 3			
Tarea 5: Eventos			
Usuario, ¿escogió tarea 5?			
Si		No	
Sistema, obtiene datos de categorías		Pasa a tarea 6	
Usuario, ¿Selecciono una categoría?			
Si	No		
Sistema, muestra datos de eventos	Usuario, continua en tarea 5 o regresa a tarea 3		
Usuario continúa en tarea 5 o regresa a tarea 3			
Tarea 6: Avisos			
Usuario, ¿escogió tarea 6?			
Si		No	
Sistema, obtiene datos de avisos		Pasa a tarea 7	

Usuario continúa en tarea 6 o regresa a tarea 3	
Tarea 7: Salir	
Usuario, ¿Salir del sistema?	
Si	No
Fin	Regresa a tarea 3

Ahora se explicará el funcionamiento del usuario Administrador y los datos requeridos

Tabla 4 Explicación de diagrama Administrador

Administrador		
Tarea 1: Acceso		
Si	No	
Administrador, completar campos requeridos: correo, contraseña.	Fin	
Sistema, Verifica credenciales		
Sistema, ¿Datos correctos?		
Si		No
Pasa a Tarea 2		Administrador, recuperar contraseña. Campos requeridos: correo
Si		No
Sistema envía datos de acceso Regresa a tarea 1		Fin
Administrador, regresa a tarea 1		

Tarea 2: Acceso al menú principal	
Administrador, puede escoger tarea 3,4,5 o 6	
Administrador, escogió tarea 3: Eventos	
Si	No
Sistema, despliega menú	Pasa a tarea 4
Tarea 3.1: crear categoría	
Administrador, crea categoría Campos requeridos: título, descripción imagen	
Sistema, Actualizar datos de categorías	
Administrador, continua en tarea 3.1 o pasa a tarea 3 o 3.2,	
Tarea 3.2: Crear evento	
Administrador: crea evento Campos requeridos: título, descripción, categoría, imagen, fecha, ubicación final, fecha inicial	
Sistema, Actualiza datos de eventos	
Administrador, continua en tarea 3.2 o pasa tarea 3 o 3.3	
Tarea 3.3: Editar categoría o evento	
Administrador, puede seleccionar tarea 3.3.1 o 3.3.2	
Tarea 3.3.1: Editar o eliminar categoría	
Sistema, despliega datos de categorías y eventos asignados	
Caso: Administrador selecciona categoría	

Si	No	
Administrador, selecciona opciones Editar o eliminar	Continua en 3.3 o regresa a tareas anteriores	
Caso: editar categoría Completa requeridos; título, descripción imagen		
Sistema actualiza datos de categoría		
Caso: eliminar categoría		
Administrador, acepta eliminar categoría		
Sistema actualiza datos de categoría		
Tarea 3.3.2: Editar o eliminar evento		
Administrador, ¿selecciono un evento?		
Si	No	
Administrador puede editar o eliminar evento	Continúa en 3.3 o regresa a tareas anteriores	
Caso Administrador editar evento		
Administrador actualiza datos de evento Campos requeridos: título, descripción, categoría, ubicación, fecha inicial, fecha final		
Sistema, Actualiza datos de evento		
Caso: Administrador elimina evento		
Administrador acepta eliminar evento		

Sistema, Actualiza información de eventos	
Administrador, continua en 3.3 o tareas anteriores	
Tarea 4: Avisos	
Administrador, selecciona evento	
Sistema despliega datos de avisos	
Administrador completa datos. Datos requeridos: mensaje	
Sistema actualiza datos de avisos	
Administrador, continúa en tarea 4 o continúa en tarea 5	
Tarea 5: editar acceso	
Administrador, cambia configuración de acceso Campos requeridos : correo nuevo, contraseña nueva, contraseña vieja	
Sistema ¿Se realizaron cambios?	
Si	No
Sistema, actualiza datos de acceso	Sistema, no realiza cambios
Usuario continua en tarea 5 o continua en tarea 6	
Tarea 6: Salir	
Administrador, escogió salir	
Si	No
Fin	Administrador, regresa a tareas anteriores.

2.6.2 Las tareas

En el diagrama BPMN como en la explicación del mismo podemos observar diferentes subdivisiones llamadas tareas. Las tareas son un conjunto de acciones que se realizarán de forma secuencial entre los usuarios y el sistema, cada una de

ellas esta delimitada de acuerdo a una o a varias acciones con base a la función que se requiera realizar.

Para determinar la segmentación de cada una de las tareas, esta acción se realiza teniendo en cuenta los requerimientos entre el funcionamiento requerido, el uso de las variables y la interacción con otros datos.

2.7 Modelado de base de datos conceptual

Teniendo en cuenta el modelado de procesos del diagrama y ajustando los requisitos por parte de la explicación podemos crear un modelo relacional que nos permita establecer los elementos que deberán ser almacenados, cada uno de los atributos de las diferentes entidades busca ser clara y especifica en su función.

A continuación se muestra el diagrama entidad relación.

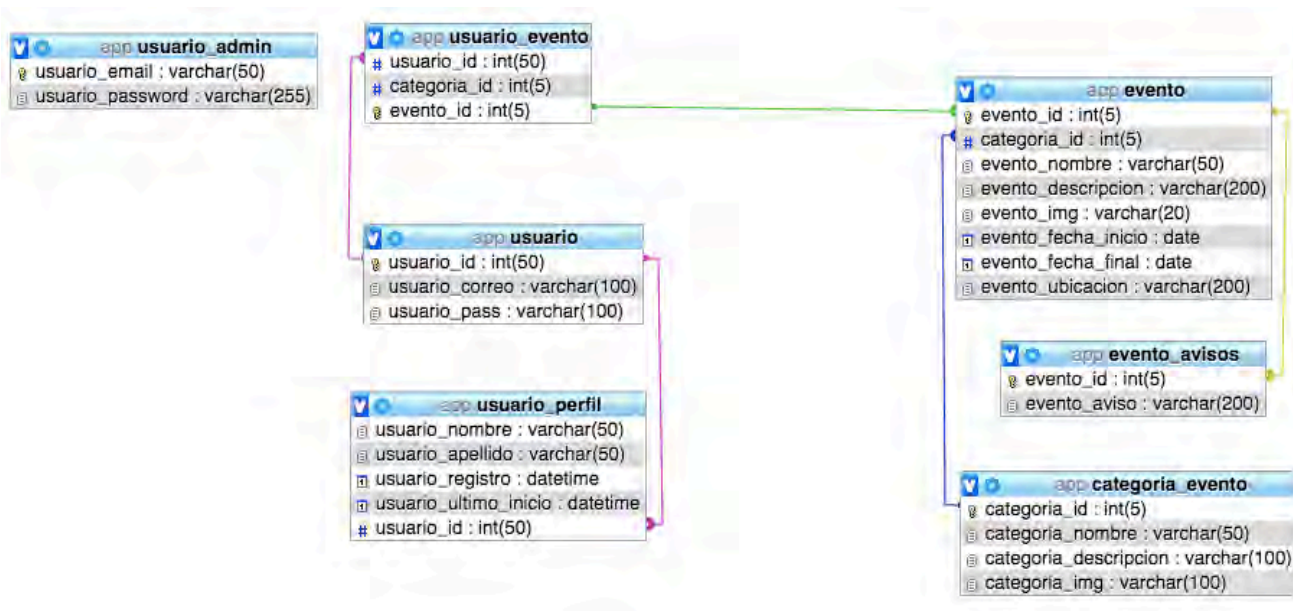


Figura 5 Modelo de base de datos relacional

2.7.1 Explicación de la creación de los datos en el modelo relacional.

Para definir los datos en el modelo entidad relación mostrado en la figura 5 se utilizaron las variables de la propuesta en la definición y alcance ajustando los requerimiento de los procesos tanto en la explicación como en el diagrama BPMN.

A continuación se explica el propósito de cada una de las tablas y el uso de los valores en ellas:

Tabla 5 Explicación de datos de usuario administrador

Datos de usuario administrador
usuario_admin: Esta tabla cumplirá la función de acceso a la plataforma web. Utilizando un correo y una contraseña valido.
categoria_evento: Los eventos estarán divididos en categorías con el fin de contar con un orden al momento de ser seleccionados. Los datos de las categorías serán contituidos por un titulo, una descripcion y una imagen (la imagen es para dar presencia y enfasis a la categoría).
evento: Los datos del evento consisten en un titulo, descripción, fecha de inicio fecha final , ubicación, imagen (imagen o promomocional del evento) y los identificadores. Los datos en esta tabla cumplen la función de describir los aspectos tecnicos del evento publicado. Y los identificadores permitiran crear el enlace con otras tablas

evento_avisos: Los datos de los aviso estan constituidos por un identificador que permitirá realizar el enlace a otras tablas, y un mensaje que será enviado a la aplicación móvil

El concepto de esta tabla radica en la creación de notificaciones de los distintos eventos

Tabla 6 Explicación de los datos de usuario cliente

Datos de usuario cliente
usuario: los datos de usuario serviran para acceder al contenido de la aplicación móvil. Utilizando un correo y contraseña valido
usuario_perfil: Estos son los datos personales del usuario los cuales son el nombre, el apellido y fecha de nacimiento. Los datos requeridos en esta sección serán utilizados conforme al uso de la politica de datos personales.
usuario_evento: esta tabla esta diseñada para registrar las visitas de las categorias o los eventos visitados por el usuario, registrando.

2.8 Diseño físico

```
CREATE TABLE `evento` (  
  `evento_id` int(5) NOT NULL,  
  `categoria_id` int(5) NOT NULL,  
  `evento_nombre` varchar(50) CHARACTER SET utf8 NOT NULL,  
  `evento_descripcion` varchar(200) CHARACTER SET utf8 NOT NULL,  
  `evento_img` varchar(20) CHARACTER SET utf8 NOT NULL,  
  `evento_fecha_inicio` date NOT NULL,  
  `evento_fecha_final` date NOT NULL,  
  `evento_ubicacion` varchar(200) COLLATE utf8_spanish2_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `evento_avisos` (  
  `evento_id` int(5) NOT NULL,  
  `evento_avisos` varchar(200) COLLATE utf8_spanish2_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `usuario` (  
  `usuario_id` int(50) NOT NULL,  
  `usuario_correo` varchar(100) COLLATE utf8_spanish2_ci NOT NULL,  
  `usuario_pass` varchar(100) COLLATE utf8_spanish2_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `usuario_admin` (  
  `usuario_email` varchar(50) COLLATE utf8_spanish2_ci NOT NULL,  
  `usuario_password` varchar(255) COLLATE utf8_spanish2_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish2_ci;
```

```
CREATE TABLE `usuario_evento` (  
  `usuario_id` int(50) NOT NULL,  
  `categoria_id` int(5) NOT NULL,  
  `evento_id` int(5) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `usuario_perfil` (  
  `usuario_nombre` varchar(50) COLLATE utf8_spanish2_ci DEFAULT NULL,  
  `usuario_apellido` varchar(50) COLLATE utf8_spanish2_ci DEFAULT NULL,
```

```

`usuario_registro` datetime NOT NULL,
`usuario_ultimo_inicio` datetime NOT NULL,
`usuario_id` int(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

```

2.9 Diccionario de datos

A continuación se explicará el contenido del diccionario de datos que componen la base de datos.

usuario

Tabla 7 Datos de usuario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
usuario_id (<i>Primaria</i>)	int(50)	No			Identificador del usuario
usuario_correo	varchar(100)	No			Correo del usuario
usuario_pass	varchar(100)	No			Contraseña del usuario

usuario_admin

Tabla 8 Datos de acceso de usuario administrador

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
usuario_email (<i>Primaria</i>)	varchar(50)	No			Correo de acceso del usuario administrador
usuario_password	varchar(255)	No			Contraseña de acceso del usuario administrador

categoria_evento

Tabla 9 Elementos de la tabla categoría evento

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
categoria_id (Primaria)	int(5)	No			Identificador de la categoría
categoria_nombre	varchar(50)	No			Nombre de la categoría
categoria_descripcion	varchar(100)	No			Descripción de la categoría
categoria_img	varchar(100)	No			Ruta de la imagen de la categoría

Evento

Tabla 10 Datos del evento

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
evento_id	int(5)	No		usuario_evento -> evento_id	Identificador del evento
categoria_id	int(5)	No		categoria_evento -> categoria_id	Identificador de la categoría
evento_nombre	varchar(50)	No			Nombre del evento
evento_descripcion	varchar(200)	No			Descripción del evento
evento_img	varchar(200)	No			Ruta de la imagen
evento_fecha_inicio	date	No			Fecha de inicio del evento
evento_fecha_final	date	No			Fecha del final del evento
evento_ubicacion	varchar(200)	No			Ubicación del evento

evento_avisos

Tabla 11 Datos de los avisos de eventos

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
evento_id (Primaria)	int(5)	No		evento -> evento_id	Identificador del evento
evento_avisos	varchar(200)	No			Mensaje de aviso

usuario_evento

Tabla 12 Datos de registro de usuario a eventos y categoría

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
usuario_id	int(50)	No		usuario -> usuario_id	Identificador del usuario
categoria_id	int(5)	No			Identificador de la categoría
evento_id (Primaria)	int(5)	No			Identificador del evento

usuario_perfil

Tabla 13 Datos de perfil de usuario

Columna	Tipo	Nulo	Predeterminado	Enlaces a	Comentarios
usuario_nombre	varchar(50)	No	NOT NULL		Nombre del usuario
usuario_apellido	varchar(50)	No	NOT NULL		Apellido del usuario
usuario_registro	date	No			Fecha de nacimiento del usuario
usuario_ultimo_inicio	datetime	No			Fecha de registro a la aplicación
usuario_id	int(50)	No		usuario -> usuario_id	Identificador de usuario

2.10 Scrum

Una vez analizado esta metodología de desarrollo la podemos a partir de las tareas establecidas en el diagrama BPMN creando bacilos de acuerdo con la investigación realizada estas son las acciones que se realizaran de forma previa a la liberación del sistema en un tiempo estimado priorizando algunas tareas dependiendo de qué tan grande sea o la dificultad que conlleve realizarlas y contando con un tiempo estimado para terminarlas.

Los bacilos están organizados de manera secuencial conforme a la descripción del diagrama BPMN con una señalización de colores haciendo énfasis en el tipo de tarea y la prioridad.

La señalización de colores será la siguiente:

- Verde: prioridad baja
- Amarillo: prioridad media
- Rojo: prioridad alta

A continuación se muestra el backlog del sistema web

Tabla 14 Back log (Lista de tarea) plataforma web

Plataforma web (Administrador)	
Tareas a realizar	Tiempo estimado
Tarea 1	6 hrs
Crea sistema de acceso para sistema web y agregar un usuario y contraseña.	
Tarea 2	6 hrs
Creación de la maquetación del menú principal.	
Tarea 3	16 hrs
Creación de módulos de evento	
Tarea 4	4 hrs
Creación de módulo de avisos	
Tarea 5	4 hrs
Creación de módulo de usuarios	
Tarea 6	8 hrs
Creación de API	
Tiempo estimado:	44 hrs

Como podemos observar en el backlog del sistema web se encuentran 2 tareas de alta prioridad, el trabajo de desarrollo debe enfocarse en ellas primero ya que su implementación depende el funcionamiento de la aplicación móvil además de contar con el mayor tiempo de desarrollo. Quedando la asignación de tareas de la siguiente manera:

Tabla 15 Reordenamiento de tareas para la plataforma web

Tarea 3
Tarea 6
Tarea 2
Tarea 4
Tarea 1
Tarea 5

Ahora se explicara el backlog para la aplicación móvil.

Tabla 16 Back log (Lista de tarea) Aplicación móvil

Aplicación móvil (cliente)	
Tareas a realizar:	Tiempo estimado
Tarea 1	6 hrs
Crear sistema de registro de usuario.	
Tarea 2	6 hrs
Verificación de credenciales y envío de contraseña	
Tarea 3	2 hrs
Diseño del menú: Maquetar el diseño estético del menú principal	
Tarea 4	4 hrs
Opción perfil: Crear la activista con los campos requeridos y verificar los cambios realizados.	
Tarea 5	8 hrs
Opción Eventos: Crear las activistas correspondientes y consumir los datos del API y maquetarlos en listas.	

Tarea 6	6 hrs
Opción Avisos: Crear la activista correspondiente y consumir los datos del Api maquetando los resultados en listas.	
Tarea 7	2 hrs
Opción Salir: Crear la función correspondiente, liberando memoria.	
Tiempo estimado:	34 hrs

De igual forma que en el backlog anterior la asignación de tareas debe priorizar los elementos más complicados y con mayor tiempo de desarrollo en este caso es establecer la comunicación con el API y maquetar los resultados obtenidos, ya que al realizar esta tarea podremos crear otras actividades similares de manera más rápido.

Tabla 17 Reordenamiento de tareas para la aplicación móvil

Tarea 5
Tarea 6
Tarea 2
Tarea 4
Tarea1
Tarea 3
Tarea 7

2.11 Licencia de software

El uso de la licencia de software estará sujeta a licencia “Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional”. Esta licencia permitirá a otros distribuir, re mezclar, retocar, y crear a partir de esta obra siempre y cuando se acate las condiciones de la misma.



(Creative Commons (S.F))

Figura 6 Símbolo de licencia de creativa comámonos

2.12 Política de privacidad y manejo de datos personales

Debido al uso de datos personales se requiere contar con una política de manejo de datos esto con el fin de generar confianza y credibilidad a los usuarios de acuerdo con la legislación vigente, los usuarios de la aplicación móvil podrán consultar la política a través de un link proporcionado en la aplicación móvil.

POLITICA DE PRIVACIDAD

La presente política de privacidad establece que la aplicación móvil Cultural usa y protege la información que es proporcionada por los usuarios al momento de usar el software. El departamento de Difusión cultural de la FES Cuautitlán se compromete con la seguridad de los datos del usuario.

Información que es recogida

Nuestra aplicación móvil podrá recoger información personal por ejemplo:

Nombre, información de contacto como su dirección de correo, nombre, apellido y fecha de nacimiento.

Uso de la información recogida

Nuestra aplicación móvil emplea la información con el fin de proporcionar el mejor servicio posible, particularmente para mantener un registro de usuarios que ayude a mejorar nuestros servicios.

Cultural está altamente comprometido para cumplir con el compromiso de mantener su información segura. Usamos los sistemas avanzados y actualizaciones regulares para asegurarnos que no exista ningún acceso no autorizado.

2.13 Requerimientos de software para el desarrollo del software.

De acuerdo con la investigación realizada anteriormente y considerando las necesidades para desarrollar ambas plataformas y valorando sus ventajas se opto por utilizar los siguientes lenguajes:

Lenguajes para crear la plataforma web

Frontend

- HTML 5: Este lenguaje permitirá crear el maquetado del sistema.
- JavaScript: Este lenguaje se utilizará para apoyar a las diferentes funciones de validación y soporte para algunos componentes visuales.
- Bootstrap (©,2018): Este framework será utilizado para crear un diseño responsivo, esto quiere decir que el diseño podrá adaptarse a diferentes pantallas.
- CSS: las hojas de estilo permitirán darle un estilo visual al sistema web.

Backend

- PHP: Este lenguaje controlará la lógica del sistema web, además de crear la interacción con la base de datos.
- JSON: Este formato de texto ligero permitirá el intercambio de datos entre el sistema web y la aplicación móvil.
- MariaDB: Este sistema de base de datos servirá para almacenar la información de los usuarios y eventos.

Para realizar la aplicación móvil se utilizará el entorno de desarrollo Android Studio además de utilizar los siguientes lenguajes :

Frontend

- XML: Este lenguaje será utilizado para crear la maquetación de la aplicación.

Backend

- Java: Este lenguaje será utilizado para crear la lógica dentro de la aplicación móvil.

Además se utilizaran las siguientes herramientas:

XAMPP® (Apache 2018) que es un servidor web capaz de interpretar un gran número de lenguajes de programación web.

Android Studio (CC) Es un entorno de desarrollo que servirá para crear la aplicación móvil.

Editor de textos: Este es un entorno que nos permitirá desarrollar los scripts para la plataforma web.

2.14 Limitaciones del sistema

De acuerdo a la propuesta empleada se fueron obteniendo diversas limitaciones que el proyecto tuvo al momento de concluir los cuales son:

- La recuperación de datos de eventos y categorías eliminados en la plataforma web.
- Recuperar información fuera de línea en la aplicación móvil.

2.15 Requerimientos de hardware y software para implementar la plataforma web y la aplicación móvil.

Para poder implementar las plataforma web se requerirá de un equipo de computo con estas especificaciones mínimas:

- Procesador Core i5
- 4GB de memoria RAM
- Acceso a Internet
- Navegador web (Mozilla Firefox, Opera y Chrome).

Para poder implementar la aplicación móvil se requerirá de los siguientes requisitos:

- Un dispositivo electrónico con una versión de Android 4.1 como mínimo.
- Una cuenta como desarrollador en la tienda de aplicaciones.

Capítulo 3: Desarrollo e implementación

Una vez establecido el flujo de funciones y tareas que tendrá el proyecto para los distintos casos de usos en los diferentes roles, a partir de la lista de tareas en el baklog con base a la propuesta se puede proceder a llevar acabo la programación de las plataformas.

3.1 Interfaces de conexión

El proyecto utilizará 3 interfaces de comunicación distintas, la primera será la configuración utilizada para enlazar la base de datos con la plataforma web. A continuación en la siguiente imagen se muestra la configuración de la interfaz que utilizará la plataforma web.

```
<?php
//Configuracion de host

define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_DATABASE', 'app');

$db = mysqli_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DATABASE);

?>
```

Figura 7 Configuración de conexión a la base de datos

La segunda interfaz que se utilizará será un API esta interfaz enlazará funciones de consulta de forma bidireccionalmente con la aplicación móvil y la base de datos suministrando de información a ambas partes.

Los acceso que se utilizarán para interactuar serán los siguientes:

- Login
- Registro
- Recuperar acceso
- Categoría
- Eventos
- Perfil
- Avisos

Por último la interfaz que utilizará la aplicación móvil para interactuar con el API se mostrará en la siguiente imagen:

```
public class URL_API {  
  
    public static String SERVER ="http://variablecrud.com/cultural/api/";  
    public static String SOURCE= "http://variablecrud.com/cultural/upload/";  
  
    /* URL MENU DATA*/  
    public static String AVISOS =SERVER+"avisos.php";  
    public static String CATEGORIA =SERVER+"categorias.php";  
    public static String EVENTO = SERVER +"eventos.php?id_categoria=";  
  
    /* URL USER DATA*/  
    public static String REGISTRO ="http://variablecrud.com/cultural/api/registro_usuario.php";  
    public static String LOGIN ="http://variablecrud.com/cultural/api/login.php";  
    public static String GET_PERFIL =SERVER+"getPerfil.php?correo=";  
    public static String RECUPERARACCESO="http://variablecrud.com/cultural/api/recuperar.php";  
  
    public static String ACTUALIZAR_ACCESO="http://variablecrud.com/cultural/api/actualizar_acceso.php";  
    public static String ACTUALIZAR_PERFIL="http://variablecrud.com/cultural/api/updateusuario.php";  
  
}
```

Figura 8 Interfaz de acceso al API de la aplicación móvil

3.2 Creación de la plataforma web

Una vez establecido la lista de tareas se procederá a crear la plataforma web, escalando cada una de las tareas de acuerdo al tipo de prioridad de mayor a menor esto permitirá agilizar la creación de software centrándose en resolver un problema en específico en cada una de ellas.

3.2.1 Descripción de módulos de la plataforma web

- **Eventos:** Este modulo servirá para gestionar toda la información de los eventos que se requiera publicar en la aplicación móvil.
- **Usuario:** El modulo servirá para cambiar la configuración de acceso a la plataforma web, como medida de seguridad se requerirán actualizar regularmente esta información.
- **Mensajes:** Este modulo servirá para enviar notificaciones de un evento en específico, con el fin de retroalimentar la información de un evento en específico.

3.3 Creación del API

Como anteriormente se especifico en el apartado de interfaces de conexión aquí se crearán los enlaces con las funciones con las que proveerán de información y realizarán la interacción entre la aplicación móvil y la base de datos a través de repuestas en formato JSON. Al igual que la plataforma web se utilizará el servidor creando un espacio en el, para su elaboración utilizando el editor de textos con el lenguaje de scripting php.


```

[
  {
    "categoria": [
      {
        "categoria_nombre": "Arte", "categoria_descripcion": "Disfruta de nuestras nuevas exposiciones", "categoria_img": "54477.png", "categoria_id": "41" }, {
        "categoria_nombre": "Ciencia", "categoria_descripcion": "Descubre las nuevas ponencias y exposiciones.", "categoria_img": "32682.png", "categoria_id": "42" }, {
        "categoria_nombre": "Cultura", "categoria_descripcion": "Descubre los nuevos talleres y eventos disponibles que tenemos para la nueva temporada", "categoria_img": "68558.png", "categoria_id": "44" }, {
        "categoria_nombre": "prueba", "categoria_descripcion": "descripcion de categoria", "categoria_img": "28878.png", "categoria_id": "55" }, {
        "categoria_nombre": "Categoria \u00e9dada", "categoria_descripcion": "Nuevos eventos disponibles", "categoria_img": "54278.png", "categoria_id": "57" }, {
        "categoria_nombre": "General", "categoria_descripcion": "Nuevas actividades de interes", "categoria_img": "12569.jpeg", "categoria_id": "52" } ] } ]

```

Figura 9 Respuesta del API a travez de una consulta en formato JSON para obtener las categorías

3.4 Creación de la aplicación móvil

Gracias a la investigación realizada se descubrió que el entorno de desarrollo especializado para el desarrollo de la aplicaciones móviles nativas cuenta con las herramientas para poder facilitar su creación entre ellas podemos encontrar un sistema de depuración de código y un creador de emuladores de dispositivos para realizar las pruebas.

3.4.1 Descripción de módulos de la aplicación móvil.

- **Perfil:** En la primera opción el usuario podrá ver y/o actualizar sus datos personales o de acceso
- **Eventos:** En opción activista el usuario podrá ver los datos de las diferentes eventos, fechas y descripciones.
- **Avisos:** En opción el usuario podrá recibir información de los eventos.

3.5 Narrativa de implementación de sistema

Una vez completado el desarrollo de las plataformas se procederá a la fase de implementación en esta sección se describirán los procesos de implementación de ambas plataformas.

3.6 Implementación de la plataforma web

Para la instalación del sistema web, de no contar con un servidor web propio se puede hacer uso de algún sistema de host de un proveedor que pueda ser compatible con las tecnologías empleadas, a partir de ello se puede proceder a colocar con cuidado en la carpeta raíz todo el sistema de archivos que componen la aplicación. Adicionalmente se debe de cambiar la configuración del archivo que permia hacer la conexión con la base de datos.

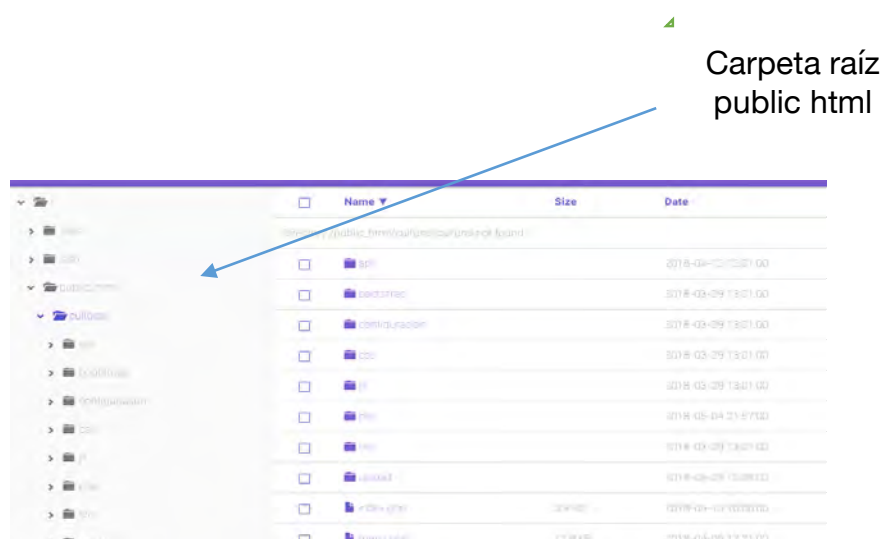


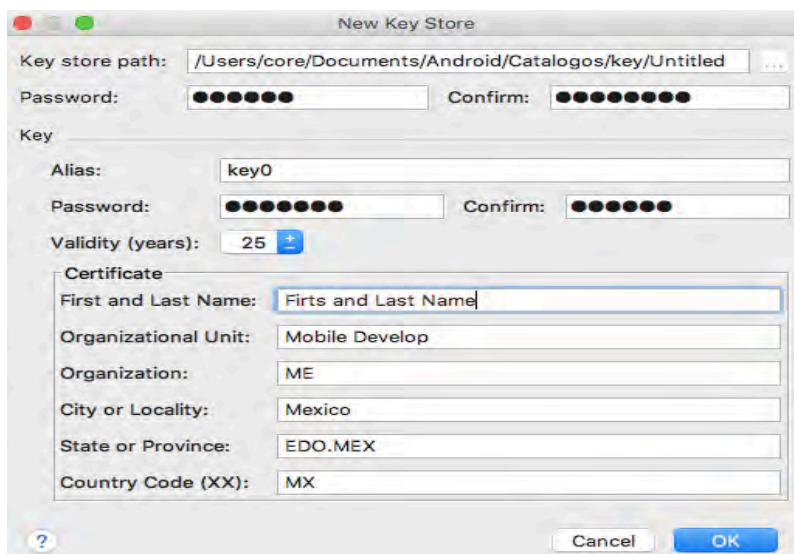
Figura 10 Explicación de como colocar la plataforma web en la carpeta raíz del servidor

Como anteriormente se estableció como requisito para publicar la aplicación móvil se debe de contar con una cuenta de desarrollador de Google Play,

3.7.1 Firma de aplicación

“Firmar la aplicación sirve como una “huella digital” que asocia de manera exclusiva el APK contigo y con tu clave privada correspondiente. Esto permite a Android verificar que cualquier actualización futura a tu APK sea auténtica y provenga del autor original. La clave que se usa para la creación de este certificado se llama *clave de firma de aplicaciones*” (Google, Firmar tu aplicación S.F).

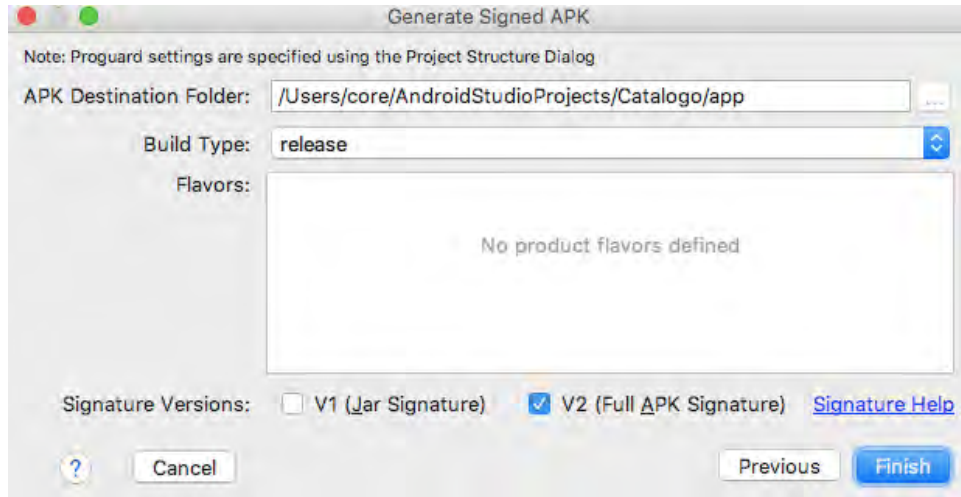
Para poder firmar la aplicación, el IDE provee una herramienta interna que permite firmar digitalmente con un certificado, para ello primero se debe crear una clave de firma, como se muestra en la siguiente imagen.



(Google, Firmar tu aplicación (S.F))

Figura 11 ejemplo de creación de nueva llave

Una vez creado la llave se puede proceder a generar el archivo APK para ello dentro de la misma sección donde generamos la llave podemos crearlo.



(Google, Firmar tu aplicación (S.F))

Figura 12 Creación de archivo apk y key

Una vez presionado el botón Finish se creara un archivo APK con la firma y un archivo KEY, cabe señalar que el archivo key servirá no solamente para firmar otras aplicaciones con la misma llave sino que también será necesaria si se desea realizar una actualización.

3.7.2 La consola de desarrollador

Ya con ambos archivos creados se puede proceder a subir la aplicación a la tienda para ello se deberá acceder a la consola de desarrollador de Android utilizando tu cuenta de desarrollador google a través del navegador.

Una vez en la consola de podremos subir la aplicación empezando por presionar el botón CREAR UNA APLICACIÓN.

Inmediatamente aparecerá un cuadro de diálogo que permitirá ponerle un nombre al proyecto. Una vez colocado nombre al proyecto una nueva pantalla aparecerá, para configurar el contenido de nuestra aplicación en la tienda.

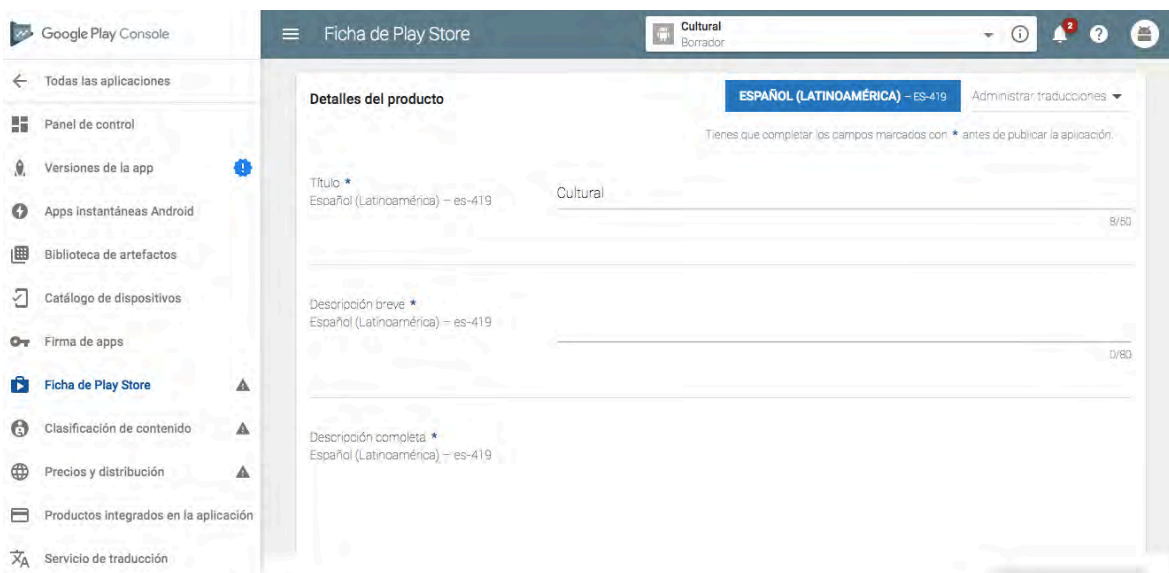


Figura 13 Configuración de la aplicación móvil en la consola

A pesar de haber una gran cantidad de opciones en esta sección solo nos enfocaremos a las mas importantes las cuales son:

- Versiones del app
- Ficha de Play Store
- Clasificación de contenido
- Precios y distribución.

3.7.3 Versiones del app

En esta sección podremos subir nuestra aplicación móvil para ser puesta en funcionamiento en diferentes fases ya sea como de prueba interna, Alfa, Beta o llevarla directamente a producción.

3.7.4 Ficha de Play Store

En esta sección podremos añadir descripciones y recursos gráficos para publicitar nuestra aplicación. Además de agregar elementos de categorización, el contacto con el equipo de desarrollo y la política de privacidad.

3.7.5 Clasificación de contenido

Este apartado es muy importante ya que los calificadores determinarán a que sector de publico a la que ira dirigida la aplicación, a través de un cuestionario tendremos que responder con total honestidad con base al contenido del software.

3.7.6 Precios y distribución.

En esta opción podremos colocar el precio si es que se requiere colocarlo para descargar la aplicación o colocarla de forma gratuita, además de administrar los países a los cuales podrá estar disponible.

3.7.7 Aplicación en producción

Una vez hecho los paso anteriores aparecerá un botón en la parte superior llamado Publicar aplicación al hacer clic aparecerá un mensaje señalando que tu aplicación será publicada en un par de horas, Además de proporcionarte un link con la dirección en donde estará publicada.

Conclusiones

Como pudimos observar durante la elaboración del proyecto, guiados por información recopilada de distintos autores y poniéndola en práctica, pudimos establecer las guías metodológicas y tecnológicas más apropiadas para establecer las bases para desarrollar la propuesta..

Con la ayuda del diagramado BPMN ajustando los requerimientos de la propuesta se consiguió crear un modelado de procesos que permitió identificar las funciones y variables que el sistema requiere para crear la interacción entre el sistema con los usuarios de la aplicación móvil y la plataforma web.

Una vez establecido el modelado de datos y funciones se designaron actividades en listas de tareas utilizando la metodología Scrum haciendo un reordenamiento de estas para priorizar actividades de acuerdo a su complejidad y/o dificultad para desarrollar cada uno de los módulos que componen el software.

Esto permitió guiar el desarrollo del software enfocándose a atender el problema principal, desarrollando el módulo principal para crear eventos en la plataforma web, seguido de la creación del API para posteriormente crear cada uno de los módulos de la aplicación móvil.

Una vez finalizado la creación del software se procedió a la implementación del sistema especificando los recursos que serían necesarios y explicando cada uno de los pasos que se realizaron para instalar la plataforma web en el servidor y publicación de la aplicación móvil en tienda de aplicaciones.

Con lo anteriormente expuesto en el inicio del proyecto se puede confirmar la hipótesis planteada al inicio del proyecto y que se lograron los objetivos, este sistema puede ayudar a mejorar la difusión de eventos culturales.

Recomendaciones de mejora

En esta sección se mencionarán algunas recomendaciones para mejorar el sistema cultural.

- Podría incluir un diseño para crear horarios dentro de los eventos.
- Crear y manejar de lista de asistencia de los eventos mediante el uso de códigos Qr.
- Inscripción a eventos con cupo limitado.
- Gestión de usuarios de la aplicación móvil.
- Envío de mensajes personalizados para un evento en particular
- Crear módulos para la inscripción a eventos que requieran inscripción.
- Crear un sistema alarmas, de recordatorio para el usuario.
- Crear un sistema de pagos si un evento lo requiere.

Bibliografía

- Len Bass, Paul Clements, Rick Kazman. *Software Architecture in practice*. 2 edición. (S.L): Addison Wesley, 2011.
- Corporación Colombia Digital. «Sistemas operativos móviles.» 26 de Enero de 2012. <https://colombiadigital.net/actualidad/articulos-informativos/item/1341-sistemas-operativos-moviles.html> (último acceso: 9 de Mayo de 2018).
- Creative Commons. *creativecommons.org*. (S.F) de (S.,F) de (S.F). <https://creativecommons.org/licenses/by-nc-sa/4.0/> (último acceso: 6 de Abril de 2918).
- ADALID corp. «Ley de protección de datos.» *Security*. 30 de Marzo de 2016. <https://www.youtube.com/watch?v=UpYpiVUdc6c> (último acceso: 18 de Mayo de 2018).
- Alvarez, Sara. *Tipos de lenguajes de programación*. 2006 de Febrero de 2006. <https://desarrolloweb.com/articulos/2358.php> (último acceso: 24 de Abril de 2018).
- adopción, BPMN:Origen y razones para su. «GlobalLogic.» *GlobalLogic*. 25 de Octubre de 2016. https://www.globallogic.com/latam/gl_news/por-que-bpmn-origen-y-razones-para-su-adopcion/ (último acceso: 2018 de Abril de 24).
- ANALITICA. *Manual de diagramación de procesos bajo estándar BPMN*. S.L: S.E, S.F. Apache. *www.apachefriends.org*. XAMPP. (S.F) de (S.F) de 2018. <https://www.apachefriends.org/es/about.html> (último acceso: 5 de Abril de 2018).
- BBVA. «Las 5 licencias de software libre más importantes que todo desarrollador debe conocer.» 29 de 2004 de 2004. <https://bbvaopen4u.com/es/actualidad/las-5-licencias-de-software-libre-mas-importantes-que-todo-desarrollador-debe-conocer> (último acceso: 9 de 5 de 2018).
- Brodie, Michael L. *On the Development of Data Models*. (S.L): (S.E), 1984.
- Diestro, Gonzalo Gabino. *Qué son las metodologías ágiles y su aplicación en el mercado actual*. (S.L), (S.F) de (S.F) de (S.F).
- Dunning, David. *Fases del diseño tradicional de bases de datos*. (S:F) de (S.F) de 2001. https://techlandia.com/tres-fases-del-diseno-tradicional-bases-datos-info_197446/ (último acceso: 24 de Abril de 2018).
- Gallego, Manuel Trigas. *Gestión de proyectos informáticos*. (S.L): (S.E), (S.F).
- García, Manuel Sierra. *¿QUÉ ES UN SERVIDOR Y CUÁLES SON LOS PRINCIPALES TIPOS DE SERVIDORES? (PROXY, DNS, WEB, FTP, SMTP, ETC.) (DV00408A)*. (S.L), (S.F) de (S.F) de 2009.
- Google. *Firmar tu aplicación*. S.F de S.F de S.F. <https://developer.android.com/studio/publish/app-signing.html?hl=es-419> (último acceso: 7 de Marzo de 2018).
- . *Firmar tu aplicación*. (S.F) de (S.F) de (S.F). <https://developer.android.com/studio/publish/app-signing.html?hl=es-419> (último acceso: 6 de Abril de 2018).
- . *play.google.com*. S.F de S.F de S.F. <https://play.google.com/apps/publish> (último acceso: 6 de Marzo de 2018).
- Grupo ISSI. *Metodologías Ágiles en el Desarrollo de Software*. Alicante, 12 de Noviembre de 2003.
- Gutiérrez, Javier J. *Lenguajes y sistemas informáticos*. (S.F) de (S.F) de (S.F). http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf (último acceso: 24 de Abril de 2018).

IntelliJ. <https://developer.android.com/studio/intro/>. Android Studio. 8 de Diciembre de 2014. developer.android.com (último acceso: 6 de Abril de 2018).

Iruela, Juan. *Los gestores de bases de datos más usados*. (S.F) de (S.F) de 2014. <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/> (último acceso: 24 de Abril de 2018).

Maria DB. *About MariaDB Software*. (S.F) de (S.F) de (S.F). <https://mariadb.com/kb/en/library/about-mariadb-software/>.

Merino, Marcos. *¿Qué es una API y para qué sirve?* 12 de Julio de 2014. <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/> (último acceso: 24 de Abril de 2018).

modelado, UML El lenguaje unificado de. «Programación en Java.» *El Lenguaje Unificado de Modelado (UML)*. (S.F) de (S.F) de (S.F). <http://elvex.ugr.es/decsai/java/pdf/3E-UML.pdf> (último acceso: 24 de Abril de 2018).

MS, Enrique M. Suárez. *¿Que es una base de datos relacional?* (S.L), (S.F) de Agosto de 2008.

Orallo, Enrique Hernández. *El Lenguaje Unificado de Modelado (UML)* . (S.L): (S.E), (S.F).

Quiroz, Angel Céspedes. *Tipos de servidores*. (SF) de (S.F) de 2011. <https://www.nubeando.com/2014/02/leccion-12-tipos-de-servidores.html> (último acceso: 24 de Abril de 2018).

Solucioneswindroid. 2 de Enero de 2018. <https://solucioneswindroid.es/2018/01/02/android-fue-el-sistema-operativo-mas-usado-en-2017-incluso-mas-que-windows/> (último acceso: 10 de Mayo de 2018).

Wikipedia®. *Programación orientada a objetos*. 14 de febrero de 2015. https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos (último acceso: 11 de Abril de 2018).

Wikipedia. *Lenguaje de programación*. (S.F) de (S.F) de (S.F). https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n (último acceso: 24 de Abril de 2018).

—. *Front-end y back-end*. (S.F) de (S.F) de (S.F). https://es.wikipedia.org/wiki/Front-end_y_back-end (último acceso: 24 de Abril de 2018).

Zeledon, Andres. *¿QUÉ ES Y CUÁLES SON LOS TIPOS DE APLICACIONES MÓVILES*. (S.F) de (S.F) de (S.F). <https://www.nextu.com/blog/tres-principales-de-aplicacion-movil/> (último acceso: 24 de Abril de 2018).

Anexos

Explicación del funcionamiento de la aplicación móvil

Splash screen

La aplicación arrancará con una presentación de un par de segundo con el símbolo del departamento de Difusión cultural e inmediatamente iniciara sesión y accederá al menú principal si es que ya se ha iniciado anteriormente, de lo contrario accederá al sistema de verificación de credenciales.



Figura 14 Imagen del departamento de difusión cultural

Verificar credenciales.

En esta sección podremos encontrar 3 opciones diferentes el login, crear una cuenta nueva y el área para recuperar la contraseña.

Login

Es la pantalla inicial después del splash screen que aparecerá si no se ha hecho un acceso previamente, el usuario deberá completar los campos requeridos para poder acceder correo y contraseña.



Figura 15 Módulo de verificación de credenciales

En caso de que los datos no sean incorrectos el siguiente mensaje se mostrará:



Por el contrario si lo datos son correctos el usuario podrá acceder al menú principal.

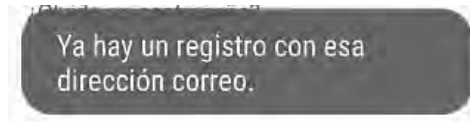
Módulo de sistema de registro

El sistema de registro servirá para dar acceso al usuario si no cuenta con un registro previo para ello debe de completar los campos requeridos: correo, contraseña, nombre, apellido y fecha de nacimiento.



Figura 16 Módulo de sistema de registro

Después de presionar el botón enviar el sistema verificara la existencia previa del campo correo si este ya ha sido utilizado por otro usuario se mostrara el siguiente mensaje:



De no ser el caso el sistema accederá al menú principal.

Recuperación de acceso:

Esta opción permitirá recupera el acceso a la aplicación del usuario introduciendo el campo requerido correo.



Figura 17 Opción para recuperar el acceso desde la aplicación

Si el campo es completado correctamente y la validación del correo es comprobada satisfactoriamente la aplicación mostrara el siguiente mensaje:



Adicionalmente se enviará un mensaje al correo con los datos de acceso del usuario.

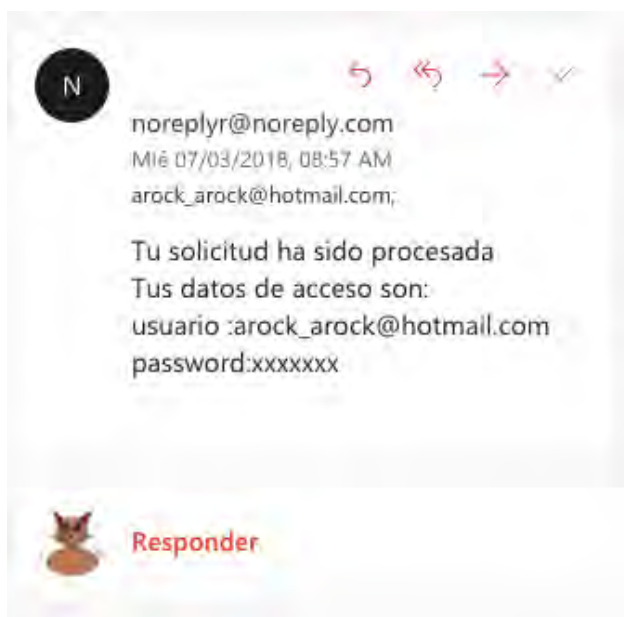


Figura 18 Muestra de envió de correo con datos de acceso al usuario desde la aplicación

En caso de no encontrar datos de accesos se mostrará el siguiente mensaje:



Menú principal.

Una vez realizado el login o el registro de manera satisfactoria se podrá acceder al menú principal, en esta sección se encuentra dividido en 4 opciones Perfil, Eventos, Avisos y una opción que permita salir del menú.



Figura 19 Pantalla del menú principal

Modulo perfil

La primera opción del menú principal Perfil permitirá realizar cambios al sistema de acceso del usuario o modificar los datos personales completando los campos requeridos: contraseña, nombre, apellido y fecha de nacimiento.

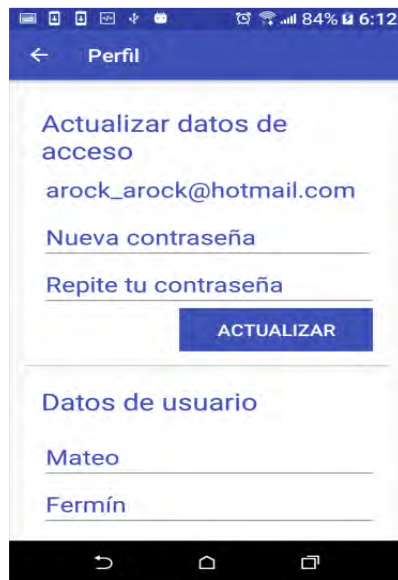


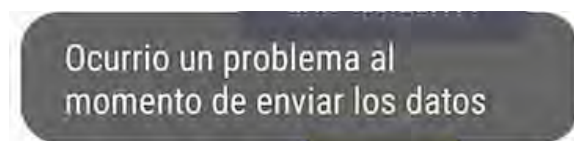
Figura 20 Modulo perfil

Después de haber realizado los cambios al presionar el botón ENVIAR, el servidor realizara la tarea de hacer los cambios en caso de haber realizado cambios correctamente el siguiente mensaje aparecerá:



Por el contrario si hubiera un inconveniente al enviar los datos el usuario recibirá el siguiente mensaje:

Figura 21 Módulo perfil



Módulo eventos.

Esta sección se encontrará dividido en 2 diferentes apartados categorías y eventos al acceder desde el menú principal a este módulo el usuario podrá elegir entre una lista de categorías la de su preferencia y poder acceder a los eventos que contiene la categoría.

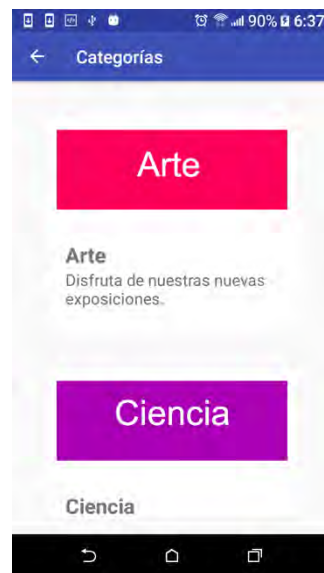


Figura 22 Módulo Evento (Categoría)

Eventos:

Una vez accedido a través de la categoría de elección se podrá ver el contenido de los diferentes eventos que se encuentran asignados a la categoría.



Figura 23 Módulo Eventos (evento)

En caso de no haber eventos disponibles dentro de una categoría aparecerá la siguiente pantalla:



Figura 24 Aviso de categoría sin eventos

Modulo Avisos

Este módulo servirá para recibir mensajes directos de la plataforma para actualizar información de los eventos estos serán visibles a través de una lista vertical con los datos evento y mensaje.

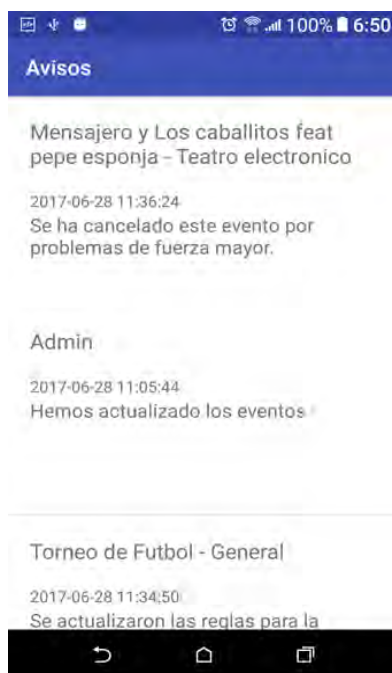


Figura 25 Módulo avisos

Opción salir

Para finalizar con esta opción permitirá salir de la aplicación eliminando los datos de acceso de la aplicación y regresando al sistema de validación de credenciales.

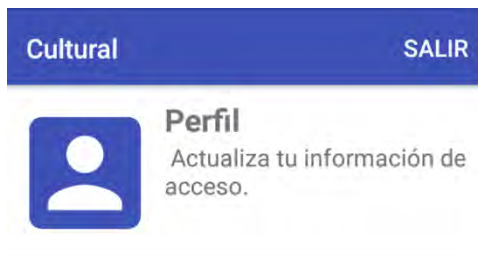


Figura 26 muestra del botón salir

Explicación del funcionamiento de la plataforma web

Sistema de acceso

Será la página principal de acceso al sistema de gestión de eventos e iniciará dando la bienvenida mostrando un formulario para los usuarios administrador, además de contar un botón que permita recuperar el acceso en caso de haber olvidado la contraseña.

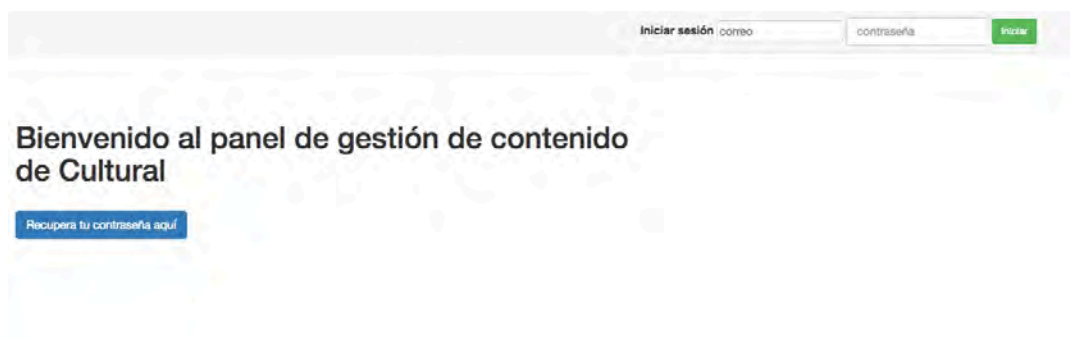
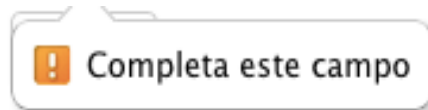


Figura 27 Pantalla principal de acceso

Acceso

Para acceder al menú principal se requiere de un usuario y contraseña y presionar el botón iniciar previamente establecidos

En caso de no completar los campos requeridos, el siguiente mensaje aparecerá:



Por el contrario si alguno de los datos de acceso es erróneo el siguiente mensaje aparecerá:

Atención! correo o contraseña incorrectos.

Recuperación de acceso

Al presionar el botón para recuperar la contraseña, aparecerá el siguiente dialogo.



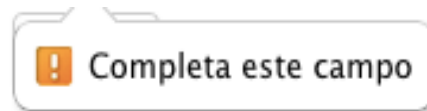
Recupera tu acceso

Enviar

Cerrar

Figura 28 Diálogo al presionar el botón para recuperar acceso al sistema

En caso de no completar los campos requeridos, el siguiente mensaje aparecerá:



Si el dato para recuperar es correcto el siguiente mensaje aparecería:

Bien! Tus datos de acceso han sido enviados.

En caso de no encontrar datos de acceso el siguiente mensaje aparecerá:

Atención! No se encontraron datos para: correo@correo.com.

El menú principal

Página inicial del sistema

La maquetación del menú principal emplea un sistema en pestañas esto permitirá realizar un cambio de acciones rápido y sencillo. En la primera sección podremos encontrar un mensaje de bienvenida unas estadísticas de y un video de guía rápida.

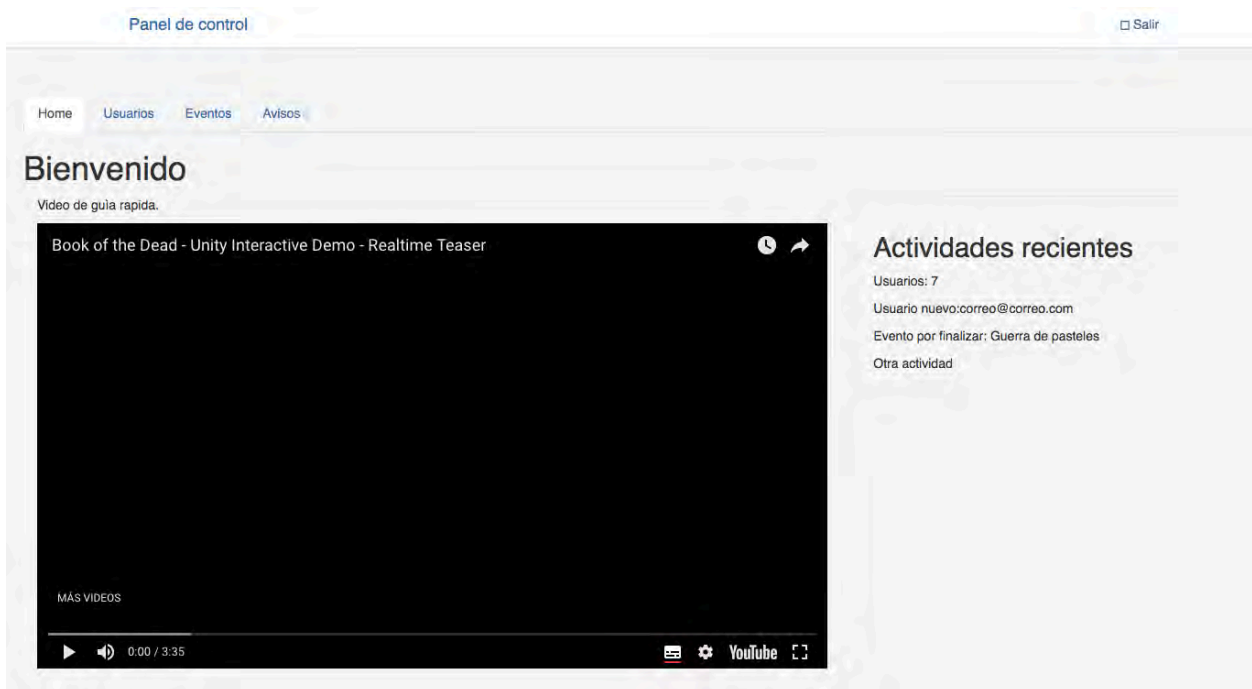


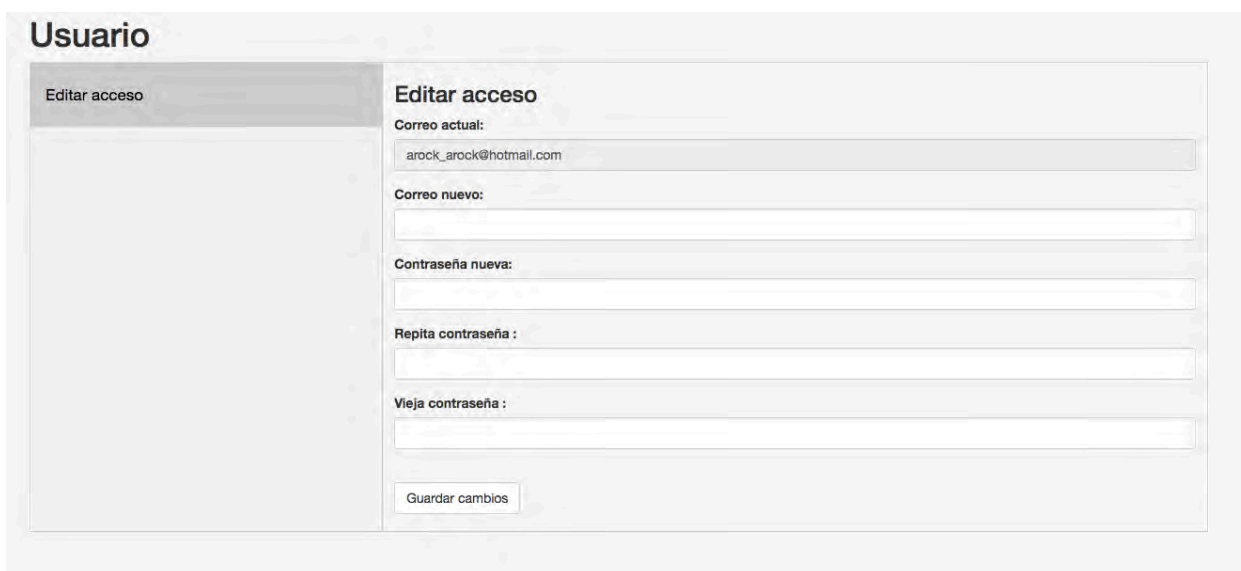
Figura 29 Página inicial del sistema

Módulo usuario

En el módulo usuario esta creado para poder actualizar los datos de acceso del usuario administrador de la plataforma web.

Actualizar acceso:

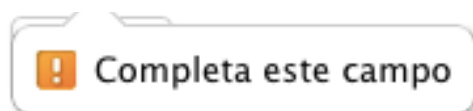
En este módulo el usuario administrador podrá cambiar las credenciales de acceso utilizando su correo y su contraseña vieja por los nuevos datos



The screenshot shows a web interface for editing user access. The main heading is 'Usuario'. Below it, there is a sidebar with 'Editar acceso' and a main content area with the same heading. The form contains the following fields: 'Correo actual:' with the value 'arock_arock@hotmail.com', 'Correo nuevo:', 'Contraseña nueva:', 'Repita contraseña:', and 'Vieja contraseña:'. A 'Guardar cambios' button is located at the bottom of the form.

Figura 30 opción editar usuario

En caso de no completar el campo requerido, el siguiente mensaje aparecerá:



Si los campos son completados requeridos son

Ok! Tus datos de acceso se han actualizado.

En caso de haber un error con la captura de los datos en el sistema un mensaje de error aparecerá como el siguiente:

No se puerón actualizar los datos de acceso! Intentalo mas tarde .

Módulo eventos.

Este módulo está dividido en 3 opciones crear categoría, crear evento y editar categoría o evento, cada una las opciones cumple con tareas específicas.

Opción crear categoría

La primera opción nos permite crear una nueva categoría completando los datos requeridos establecidos Titulo, Descripción y agregar una imagen.

Panel de control Salir

Home Usuarios **Eventos** Avisos

Eventos

- Crear categoría
- Crear evento
- Editar categoría o evento

Crear nueva categoría

Título:

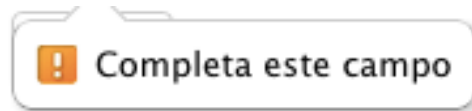
Descripción:

Imagen (jpeg,jpg,png) max 2MB

Seleccionar archivo No se eligió archivo

Figura 31 Pantalla del módulo Eventos

En caso de no completar algún campo requerido, el siguiente mensaje aparecerá:



Si todos los campos fueron completados de manera correctamente el siguiente mensaje aparecerá:

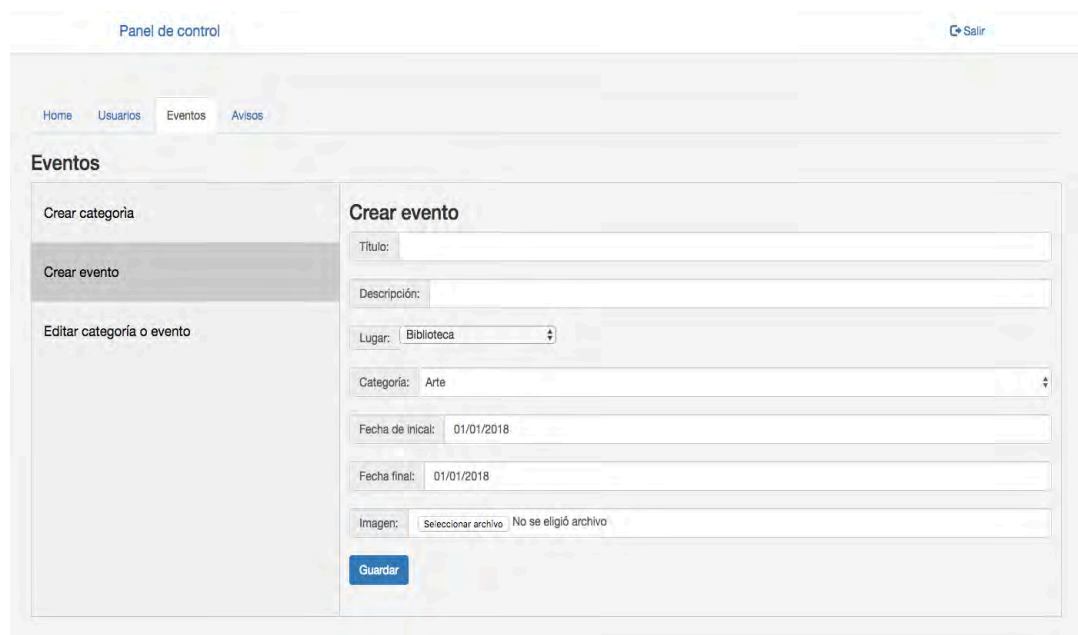
Éxito! Categoría creada.

En caso de haber un error con la captura de los datos en el sistema un mensaje de error aparecerá como el siguiente:

Atención! Hubo un problema al momento de realizar el registro porfavor ponganse con el administrador o intentelo mas tarde.

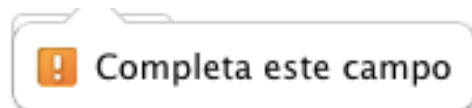
Opción eventos

En esta nueva opción se puede crear un nuevo evento en esta sección el usuario además de completar los campos requeridos: título, descripción, categoría, fecha de inicio, fecha final e imagen (en su caso).



The screenshot shows a web interface for creating an event. At the top, there is a navigation bar with 'Panel de control' on the left and 'Salir' on the right. Below this is a breadcrumb trail: 'Home > Usuarios > Eventos > Avisos'. The main content area is titled 'Eventos' and contains three sub-sections: 'Crear categoría', 'Crear evento' (which is currently selected and highlighted), and 'Editar categoría o evento'. The 'Crear evento' form includes the following fields: 'Título:' (text input), 'Descripción:' (text input), 'Lugar:' (dropdown menu with 'Biblioteca' selected), 'Categoría:' (dropdown menu with 'Arte' selected), 'Fecha de Inicial:' (date input with '01/01/2018'), 'Fecha final:' (date input with '01/01/2018'), and 'Imagen:' (file upload area with 'Seleccionar archivo' and 'No se eligió archivo' text). A blue 'Guardar' button is located at the bottom of the form.

En caso de no completar algún campo requerido, el siguiente mensaje aparecerá:



Si todos los campos fueron completados de manera correcta aparecerá la siguiente mensaje:

Éxito! Evento creado exitosamente.

De haber un error al momento de crear el nuevo evento el siguiente mensaje aparecerá.

Atención! Hubo un problema al momento de realizar el registro porfavor ponganse con el administrador o intentelo mas tarde.

Opción Editar

La opción editar permitirá al usuario cambiar la información de los eventos o las categorías. Al seleccionar una categoría la selección de eventos se desplegara con los eventos asignados a la categoría.

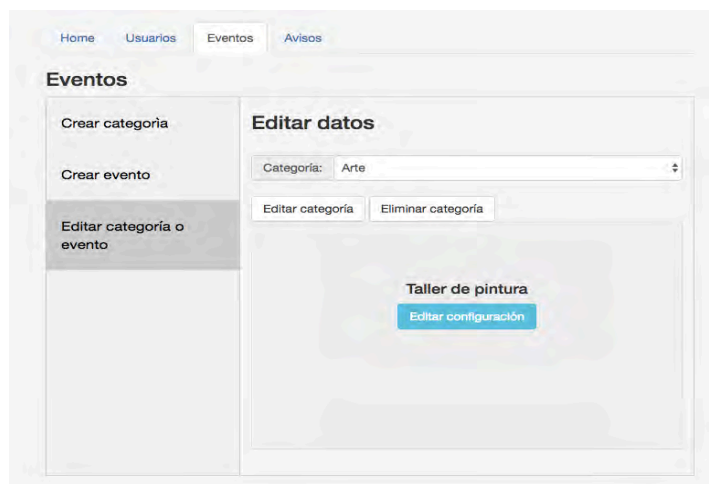


Figura 32 Opción editar

Al presionar el botón Editar categoría aparecerá un cuadro de diálogo en el cual se podrá cambiar la información de la categoría seleccionada.

Editar categoría:

El cuadro de dialogo editar categoría permitirá cambiar los datos de una categoría seleccionada introduciendo los datos requeridos título, descripción e imagen.



Editar categoría

Título: Arte

Descripción: Disfruta de nuestras nuevas exposiciones.

Imagen actual: No se eligió archivo

Arte

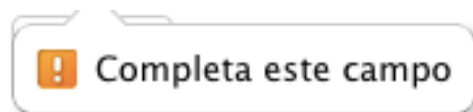
54477.png

Guardar cambios

Cerrar

Figura 33 diálogo al presionar el botón Editar categoría

En caso de no completar algún campo requerido, el siguiente mensaje aparecerá:



Si todos los campos requeridos fueron completados de manera correctamente un mensaje aparecerá:

Éxito! Los cambio se han realizado de manera satisfactoria.

En caso de haber un error con la captura de los datos en el sistema un mensaje de error aparecerá como el siguiente:

Atención! Hubo un problema al momento de realizar el registro porfavor ponganse con el administrador o intentelo mas tarde.

Al presionar el botón eliminar categoría aparecerá un cuadro de dialogo se eliminarán los datos de la categoría incluyendo los eventos a la misma.



Figura 34 Botón eliminar categoría

Al presionar el botón si aparecerá el siguiente mensaje

Atención! Esta categoría no volvera aparecer.

Si se presiona el botón No el cuadro de dialogo desaparecerá.

Editar evento

Al seleccionar un evento de la lista desplegable al seleccionar una categoría se accederá a una nueva página llamada Editar evento estará dividido en 3 partes, Evento, Editar, Eliminar. En la primera sección se podrá ver los datos del evento como se muestra en la siguiente imagen.



The screenshot shows a web interface titled "Editar evento". At the top, there are three tabs: "Evento", "Editar", and "Eliminar". Below the tabs is the section "Datos de evento". Under this section, there is a label "Imagen" followed by a large yellow rectangular area containing the word "Lógica". Below the image area, there are several form fields with labels and values:

Label	Value
Título	Conferencia de lógica
Descripción	El profesor C. Gutiérrez dará una conferencia de sobre la importancia de las métricas en el diseño de sistemas lógicos
Categoría	Ciencia
Fecha de inicio	2018-11-21
Fecha de inicio	2018-11-21

Figura 35 Diálogo al presionar el botón editar evento

En la segunda sección Editar se podrá actualizar el contenido del evento completando los campos requerido o cambiar el evento a una nueva categoría.

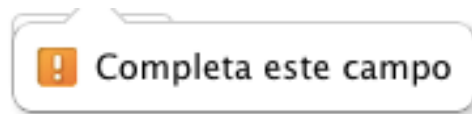
Editar evento

Evento Editar Eliminar

Actualiza el contenido publicado

Título:	<input type="text"/>
Descripción:	<input type="text"/>
Actualizar categoría :	<input type="text"/>
Fecha de inicio:	01/01/2018
Fecha de terminación:	01/01/2018
Imagen:	<input type="text" value="Seleccionar archivo"/> No se eligió archivo
<input type="button" value="Guardar cambios"/>	

En caso de no completar algún campo requerido, el siguiente mensaje aparecerá:



Si todos los campos requeridos fueron completados de manera correctamente un mensaje aparecerá:

Éxito! Los cambio se han realizado de manera satisfactoria.

En caso de haber un error con la captura de los datos en el sistema un mensaje de error aparecerá como el siguiente:

Atención! Hubo un problema al momento de realizar el registro porfavor ponganse con el administrador o intentelo mas tarde.

En la tercera sección Eliminar.

Editar evento

Evento Editar Eliminar

Eliminar

¿Deseas eliminar Conferencia de lógica?

Esta acción eliminará todos los datos asociados al evento

Si

No

Al presionar el botón Si aparecerá el siguiente mensaje:

Ok! Este evento no volverá aparecer.

Después de un par de segundo la página re direccionará al usuario al menú principal.

Al presionar el botón No aparecerá el siguiente mensaje:

Ok! Mantendremos este evento

Módulo Avisos

El módulo avisos permitirá al usuario administrador enviar mensajes a la aplicación completando los campos requeridos: evento y mensaje.



Panel de control

Home Usuarios Eventos Avisos

Avisos

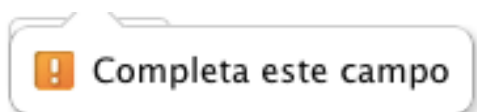
Arte
✓ Cine
Teatro

Mensaje:

Enviar

Figura 36 Módulo avisos

En caso de no completar el campo requerido, el siguiente mensaje aparecerá:



Si el campo requerido es completado correctamente aparecerá el siguiente mensaje:

Éxito! Aviso creado exitosamente.

En caso de haber un error con la captura de los datos en el sistema un mensaje de error aparecerá como el siguiente:

Atención! Hubo un problema al momento de realizar el registro porfavor ponganse con el administrador o intentelo mas tarde.

Información legal propiedad intelectual y marcas registradas

Las marcas registradas y logotipos mostrados en este proyecto son propiedad de sus respectivos dueños y son utilizados únicamente con fines enunciativos.