

UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA



**SISTEMA WEB PARA LA GESTIÓN DE LA CONGREGACIÓN Y
LOS EVENTOS DE LA IGLESIA ROCA ETERNA**

TESIS
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO INFORMÁTICO

PRESENTADO POR:

Bach. ALAMA VENTOCILLA, RICARDO RYAN

ASESOR: MG. ING. LINAREZ COLOMA, HUMBERTO VICTOR

LIMA – PERÚ

2021

Dedicatoria

Dedico este trabajo a mi papá
Por confiar en mí a pesar de todo

Agradecimientos

Agradezco a mi familia por ser de ánimo para mi vida por apoyarme en todo este tiempo, por ser de ejemplo para mi vida en distintas ocasiones.

RESUMEN

En la iglesia Roca Eterna se pudo identificar algunos problemas, en el ámbito organizacional, que hacían que los procesos que tenían que ver con los miembros y los eventos dentro de la misma, no se pudieran completar o se completaran con deficiencias. Los datos de las personas no eran registrados correctamente, no se tenía un inventario de los activos que tenía la iglesia. Los eventos a veces no se llegaban a completar, o se posponían y los líderes muchas veces no se enteraban la razón de eso; ocurrían problemas y obstáculos en los eventos y los responsables se olvidaban; los mismos eventos se hacían sin tener ningún propósito en mente. Los mismos miembros no sabían los datos de la iglesia, no tenían acceso a información de ámbito público ni de sus propios líderes. Todo eso lograba que el prestigio de la iglesia disminuyera y por consecuencia que el compromiso de los miembros bajara.

La solución que sirvió para subsanar dichos problemas gradualmente fue el desarrollo de un sistema web que ayudará a los líderes de la iglesia a, en términos generales, organizar todos los datos de la iglesia. Se construyeron módulos capaces de registrar desde los datos más generales de una persona (como su información básica) hasta algunos más específicos, como el ministerio al que pertenece y que rol cumple ahí. Se creó una página donde los miembros puedan ver la información pública que debe ser compartida dentro de la iglesia. Asimismo, se pudo construir un módulo que es capaz de añadir distintas características a un evento, como participantes, objetivos, actividades, obstáculos, necesidades, parafernalia, fotos y un pequeño CMS. El sistema web se diseñó responsivamente para que pueda adaptarse a los dispositivos móviles y está alojado en un hosting con su respectivo dominio, para su disponibilidad las 24 horas del día.

Palabras clave: Eventos, Miembros, Iglesia

ABSTRACT

In the Roca Eterna church, some problems could be identified, in the organizational sphere, that made the processes that had to do with the members and the events within it, could not be completed or were completed with deficiencies. The data of the people was not registered correctly, there was no inventory of the assets that the church had. The events were sometimes not completed, or were postponed and the leaders often did not know the reason for that; problems and obstacles occurred in events and those responsible would forget; the same events were held without having any purpose in mind. The same members did not know the data of the church, they did not have access to information from the public sphere or from their own leaders. All this made the prestige of the church decrease and consequently the commitment of the members decreased.

The solution that gradually solved these problems was the development of a web system that will help the church leaders to, in general terms, organize all the data of the church. Modules capable of registering from the most general data of a person (such as their basic information) to some more specific ones, such as the ministry to which they belong and what role they play there were built. A page was created where members can view public information that should be shared within the church. Likewise, it was possible to build a module that is capable of adding different characteristics to an event, such as participants, objectives, activities, obstacles, needs, paraphernalia, photos and a small CMS. The web system was responsively designed so that it can adapt to mobile devices and is hosted on a hosting with its respective domain, for availability 24 hours a day.

Keywords: Events, Members, Church

INDICE GENERAL

INTRODUCCIÓN	1
I. CAPÍTULO I: VISIÓN DEL PROYECTO	3
1.1. Antecedentes del problema	3
1.1.1. El negocio	3
1.1.2. Procesos del negocio	5
1.2. Descripción del problema	5
1.3. Objetivos	8
1.3.1. Marco lógico.....	8
1.3.2. Objetivo general	10
1.3.3. Objetivos específicos.....	11
1.4. Importancia	11
1.4.1. Justificación académica	12
1.5. Beneficios del proyecto.....	12
1.5.1. Beneficios tangibles.....	12
1.5.3. Beneficios intangibles	13
1.6. Alcance.....	13
II. CAPÍTULO II: MARCO TEÓRICO	14
2.1. Antecedentes	14
2.1.2. Los eventos en la iglesia.....	15
2.1.3. Un miembro dentro de la iglesia	16
2.1.4. Responsabilidades de los líderes	17
2.1.5. Crecimiento de la iglesia	18
2.2. Soporte teórico de la investigación	19
2.2.1. Aplicación Web	19
2.2.2. MySql Workbench.....	20

2.2.3. PHP.....	21
2.2.4. Laravel.....	22
2.2.5. Enterprise Architect.....	23
2.2.6. RUP	23
2.2.7. UML	24
III. CAPÍTULO III: ESTADO DEL ARTE.....	25
3.1. Introducción	25
3.2. Casos revisados	26
3.2.1. Sistema de información para la gestión de la membresía de iglesias cristianas (Anillo Farelo & Pájaro Martinez, 2014)	26
3.2.2. Desarrollo De Un Sistema Web Para Administración De Eventos Y Control De Entrada Y Salida De Empleados (Moyano Mejía, 2015).....	28
3.3. Comparación con otros sistemas similares	30
3.3.1. Iglesia Hoy.....	30
3.3.2. Hablemos de Iglesia	31
3.3.3. Softlife	32
3.3.4. Tabla de Comparación.....	33
IV. CAPÍTULO IV: MODELADO DEL NEGOCIO.....	34
4.1. Reglas del Negocio	34
4.1.1. Regla del Negocio 1	34
4.1.2. Regla del Negocio 2	34
4.1.3. Regla del Negocio 3	34
4.1.4. Regla del Negocio 4	35
4.1.5. Regla del Negocio 5	35
4.1.6. Regla del Negocio 6	35
4.1.7. Regla del Negocio 7	35
4.1.8. Regla del Negocio 8	35
4.1.9. Regla del Negocio 9	35

4.1.10.	Regla del Negocio 10.....	35
4.1.11.	Regla del Negocio 11.....	35
4.1.12.	Regla del Negocio 12.....	35
4.1.13.	Regla del Negocio 12.....	36
4.2.	Procesos del Negocio	36
4.2.1.	Relación de Procesos del Negocio	36
4.3.	Casos de Uso del Negocio.....	37
4.3.1.	Relación de casos de uso del negocio.....	37
4.3.2.	Diagrama de casos de uso del negocio	38
4.3.3.	Descripción de los casos de uso del negocio	38
4.4.	Diagramas de Actividades.....	39
4.4.1.	CUN Organizar Evento	39
4.4.2.	CUN Realizar Seguimiento	40
4.5.	Diagrama de Clases del Negocio	41
4.5.1.	CUN Organizar Evento	41
4.5.2.	CUN Realizar Seguimiento	41
V.	CAPÍTULO V: REQUERIMIENTOS DEL PRODUCTO.....	42
5.1.	Requerimientos del Software	42
5.1.1.	Requerimientos funcionales	42
5.1.2.	Requerimientos No Funcionales.....	43
5.2.	Casos de Uso del Sistema	44
5.2.1.	Actores del Sistema	44
5.2.2.	Diagrama de Paquetes	45
5.2.3.	Relación de CUS	45
5.2.4.	Diagrama General de CUS	50
5.2.5.	Diagrama de CUS por paquetes	51
5.2.6.	Matriz CUN vs CUS.....	55

5.3.	Modelo Conceptual del Sistema.....	56
5.3.1.	Modelo conceptual	56
5.4.	Prototipos	57
5.4.1.	Prototipo Solicitar Evento	57
5.4.2.	Prototipo Gestionar Participantes	57
5.4.3.	Ver Calendario.....	58
5.4.4.	Reportes.....	58
5.4.5.	Gestionar Eventos.....	59
5.5.	Matriz CUS vs FEATS.....	59
VI.	CAPITULO VI: ARQUITECTURA	60
6.1.	Casos de uso más significativos para la arquitectura	60
6.1.1.	Propósito.....	60
6.1.2.	Definiciones, Abreviaturas y Acrónimos	60
6.2.	Representación de la arquitectura	60
6.2.1.	Modelo Arquitectónico (4 + 1).....	60
6.3.	Vista de Casos de Uso.....	62
6.3.1.	Diagrama de Casos de Arquitectura	62
6.3.2.	Casos de uso más significativos	62
6.3.3.	Especificaciones de Casos de Uso.....	86
6.3.4.	Diagramas de clases Análisis	98
6.3.5.	Diagramas de Diseño.....	100
6.4.	Vista Lógica	109
6.4.1.	Vista de Componentes.....	109
6.4.2.	Vista de Despliegue.....	110
6.4.3.	Modelo Físico.....	111
6.4.4.	Diccionario de Datos	112
6.5.	Calidad	120

6.6.	Tecnologías a utilizar para la implementación.....	120
VII.	CAPÍTULO VII	121
7.1.	Desarrollo.....	121
7.1.1.	Estándares de codificación	121
7.1.2.	Estándares de campos de tablas en la Base de Datos	122
7.1.3.	Estándares de documentación.....	122
7.2.	Pruebas	123
7.2.1.	Plan de pruebas.....	123
7.2.2.	Informe de pruebas: Solicitar Evento	124
7.2.3.	Informe de pruebas: Evaluar Solicitud	125
7.2.4.	Informe de pruebas: Gestionar Eventos	126
7.2.5.	Informe de pruebas: Gestionar Objetivos.....	127
7.2.6.	Informe de pruebas: Gestionar Obstáculos.....	129
7.2.7.	Informe de pruebas: Participar en un evento	130
VIII.	CAPÍTULO VIII	131
8.1.	Viabilidad del Proyecto.....	131
8.1.1.	Viabilidad Técnica.....	131
8.1.2.	Viabilidad Económica	132
8.1.3.	Viabilidad Legal	137
IX.	CAPÍTULO IX: GESTIÓN DEL PROYECTO	138
9.1.	Organización del Proyecto	138
9.1.1.	Organigrama del proyecto	138
9.1.2.	Estructura de Desglose del proyecto (EDT).....	139
9.2.	Estimación y ejecución del proyecto.....	140
9.2.1.	Cronograma de ejecución del proyecto	140
9.2.2.	Cronograma detallado	140
	CONCLUSIONES	141

RECOMENDACIONES.....	142
BIBLIOGRAFÍA	143
ANEXO 1: Manual de Usuario - Líder de Ministerio	146
ANEXO 2: Manual de Usuario - Consistorio.....	154
ANEXO 3: Manual de Usuario - Miembro	159
ANEXO 4: Manual de Instalación.....	162

ÍNDICE DE FIGURAS

FIGURA 1 ORGANIGRAMA DE LA IGLESIA	4
FIGURA 2 DIAGRAMA DE PROCESOS DE LA IGLESIA.....	5
FIGURA 3 PORCENTAJE DE NÚMERO DE PERSONAS EN LA IGLESIA.....	6
FIGURA 4 CANTIDAD DE EVENTOS REALIZADOS 2016 - 2017.....	7
FIGURA 5 ÁRBOL DE PROBLEMAS.....	8
FIGURA 6 ÁRBOL DE OBJETIVOS	9
FIGURA 7 CICLO DE VIDA DEL RUP.....	23
FIGURA 8 DIAGRAMA DE CUS	27
FIGURA 9 DIAGRAMA DE CUS EVENTOS.	28
FIGURA 10 DIAGRAMA DE CUS CLIENTES.....	28
FIGURA 11 DIAGRAMA DE TABLAS.....	29
FIGURA 12 PROTOTIPO CALENDARIO.....	29
FIGURA 13 PANTALLA PRINCIPAL DE LA PÁGINA WEB IGLESIA HOY	30
FIGURA 14 PANTALLA PRINCIPAL DE HABLEMOS DE IGLESIA.....	31
FIGURA 15 PANTALLA PRINCIPAL DE LA PÁGINA WEB SOFTLIFE.....	32
FIGURA 16 BENCHMARKING.....	33
FIGURA 17 DIAGRAMA DE CUN.....	38
FIGURA 18 DIAGRAMA DE ACTIVIDADES ORGANIZAR EVENTO.....	39
FIGURA 19 DIAGRAMA DE ACTIVIDADES REALIZAR SEGUIMIENTO	40
FIGURA 20 DIAGRAMA DE CLASES DEL NEGOCIO ORGANIZAR EVENTOS	41
FIGURA 21 DIAGRAMA DE CLASES DEL NEGOCIO SEGUIMIENTO	41
FIGURA 22 DIAGRAMA DE ACTORES DEL SISTEMA	44
FIGURA 23 DIAGRAMA DE PAQUETES.....	45
FIGURA 24 DIAGRAMA GENERAL DE CUS.....	50
FIGURA 25 DIAGRAMA DE CUS DEL PAQUETE EVENTOS	51
FIGURA 26 DIAGRAMA DE CUS DEL PAQUETE SEGURIDAD	52
FIGURA 27 DIAGRAMA DE CUS DEL PAQUETE MANTENIMIENTO	52
FIGURA 28 DIAGRAMA DE CUS DEL PAQUETE PÚBLICO	53
FIGURA 29 DIAGRAMA DE CUS DEL PAQUETE CONGREGACIÓN	53
FIGURA 30 DIAGRAMA DE CUS DEL PAQUETE REPORTES	54
FIGURA 31 MODELO CONCEPTUAL.....	56
FIGURA 32 PROTOTIPO DEL CUS SOLICITAR EVENTO	57

FIGURA 33	PROTOTIPO DE CUS GESTIONAR PARTICIPANTES	57
FIGURA 34	PROTOTIPO DEL CUS VER CALENDARIO	58
FIGURA 35	PROTOTIPO DE VISTA DE REPORTES	58
FIGURA 36	PROTOTIPO DEL CUS GESTIONAR EVENTOS	59
FIGURA 37	MATRIZ CUS VS FEATURES	59
FIGURA 38	DIAGRAMA DEL MODELO ARQUITECTÓNICO 4 + 1	61
FIGURA 39	DIAGRAMA DE CUS DE ARQUITECTURA	62
FIGURA 40	DIAGRAMA DE CLASES ANÁLISIS GESTIONAR OBJETIVOS	69
FIGURA 41	DIAGRAMA DE CLASES DISEÑO GESTIONAR OBJETIVOS.....	70
FIGURA 42	DIAGRAMA DE CLASES DISEÑO GESTIONAR ACTIVIDADES ...	71
FIGURA 43	DIAGRAMA DE SECUENCIA DISEÑO GESTIONAR OBJETIVOS ..	73
FIGURA 44	DIAGRAMA DE SECUENCIA DISEÑO ACTIVIDADES	76
FIGURA 45	DIAGRAMA DE CLASES ANÁLISIS GESTIONAR OBSTÁCULOS .	81
FIGURA 46	DIAGRAMA DE CLASES DISEÑO GESTIONAR OBSTÁCULOS.....	82
FIGURA 47	DIAGRAMA DE SECUENCIA DISEÑO OBSTÁCULOS.....	85
FIGURA 48	DIAGRAMA DE CLASES ANÁLISIS SOLICITAR EVENTO	98
FIGURA 49	DIAGRAMA DE CLASES ANÁLISIS EVALUAR EVENTO.....	98
FIGURA 50	DIAGRAMA DE CLASES ANÁLISIS GESTIONAR EVENTOS	99
FIGURA 51	DIAGRAMA DE CLASES ANÁLISIS PARTICIPAR EN EVENTO ..	100
FIGURA 52	DIAGRAMA DE CLASES DISEÑO SOLICITAR EVENTO.....	100
FIGURA 53	DIAGRAMA DE SECUENCIA DISEÑO SOLICITAR EVENTO	101
FIGURA 54	DIAGRAMA DE CLASE DISEÑO EVALUAR EVENTO.....	102
FIGURA 55	DIAGRAMA DE SECUENCIA DISEÑO EVALUAR EVENTO.....	103
FIGURA 56	DIAGRAMA DE CLASES DISEÑO GESTIONAR EVENTOS.....	104
FIGURA 57	DIAGRAMA DE SECUENCIA DISEÑO GESTIONAR EVENTOS ...	105
FIGURA 58	DIAGRAMA DE CLASES DISEÑO PARTICIPAR EN EVENTO.....	106
FIGURA 59	DIAGRAMA DE SECUENCIA DE DISEÑO PARTICIPAR	107
FIGURA 60	DIAGRAMA DE ESTADOS	108
FIGURA 61	DIAGRAMA DE COMPONENTES.....	109
FIGURA 62	DIAGRAMA DE DESPLIEGUE	110
FIGURA 63	MODELO FÍSICO DE LA BASE DE DATOS	111
FIGURA 64	TABLA DE COSTOS.....	134
FIGURA 65	CUADRO DE BENEFICIOS ECONÓMICOS.....	135
FIGURA 66	TABLA DE VIABILIDAD ECONÓMICA	135

FIGURA 67 TABLA DE VAN Y TIR.....	136
FIGURA 68 ORGANIGRAMA DEL PROYECTO	138
FIGURA 69 EDT DEL PROYECTO	139
FIGURA 70 RESUMEN DEL CRONOGRAMA DEL PROYECTO.....	140
FIGURA 71 CRONOGRAMA DETALLADO DEL PROYECTO.....	140

INDICE DE TABLAS

TABLA 1 MATRIZ DE MARCO LÓGICO.....	9
TABLA 2 ESPECIFICACIÓN DEL ALCANCE DEL PROYECTO	13
TABLA 3 MATRIZ CUN VS CUS	55
TABLA 4 ESPECIFICACIÓN DEL CUS GESTIONAR OBJETIVOS	63
TABLA 5 ESPECIFICACIÓN DEL CUS GESTIONAR OBSTÁCULOS	77
TABLA 6 ESPECIFICACIÓN DEL CUS SOLICITAR EVENTO	86
TABLA 7 ESPECIFICACIÓN DEL CUS EVALUAR SOLICITUD	89
TABLA 8 ESPECIFICACIÓN DEL CUS GESTIONAR EVENTOS	93
TABLA 9 ESPECIFICACIÓN DE CUS PARTICIPAR EN EVENTO	96
TABLA 10 DICCIONARIO DE DATOS	112
TABLA 11 ESTÁNDARES DE CODIFICACIÓN	121
TABLA 12 ESTÁNDARES EN LA BASE DE DATOS.....	122
TABLA 13 ESTÁNDARES EN LA DOCUMENTACIÓN	122
TABLA 14 ESTÁNDARES DE LOS TÍTULOS DE LOS DOCUMENTOS.....	122
TABLA 15 PLAN DE PRUEBAS	123
TABLA 16 INFORME DE PRUEBAS DEL CUS SOLICITAR EVENTO	124
TABLA 17 INFORME DE PRUEBAS DEL CUS EVALUAR SOLICITUD	125
TABLA 18 INFORME DE PRUEBAS DEL CUS GESTIONAR EVENTOS	126
TABLA 19 INFORME DE PRUEBAS DE CUS GESTIONAR OBJETIVOS.....	127
TABLA 20 INFORME DE PRUEBAS DEL CUS GESTIONAR OBSTÁCULOS ...	129
TABLA 21 INFORME DE PRUEBA DEL CUS PARTICIPAR EN EVENTO.....	130
TABLA 22 VIABILIDAD TÉCNICA	131

INTRODUCCIÓN

En los últimos tiempos las iglesias cristianas crecen cada vez más. Encontramos iglesias como “Agua Viva”, “Movimiento Misionero Mundial”, “Camino de Vida” que son las más conocidas, si es que hablamos solo del entorno limeño. Estas iglesias tienen grandes cantidades de miembros, fácilmente superan los 200 por local. Sin embargo, a nivel internacional existen lo que se llaman “Mega Iglesias” cuyos miembros superan los 20000 miembros por iglesia. Debido a este aumento de cristianos, los eventos que realizan dichas iglesias han aumentado de igual manera. Estos son más frecuentes, más complejos, y necesitan de una correcta administración para que todo salga de mejor manera posible. Además de eso es necesario tener un registro de los datos de los miembros que asisten a la iglesia, porque esa información como cumpleaños, teléfono, etc., se puede usar para poder hacer un seguimiento cuando se requiera. Dentro de las iglesias también existen ministerios que son formados distintos miembros que realizan una determinada función dentro de la iglesia. Los ministerios también crecen, tiene sus eventos, sus integrantes, sus responsabilidades.

Por esa razón es que en la presente tesis se plantea resolver todos esos temas anteriormente descritos, con la ayuda de un sistema Web que permita realizar todas esas tareas.

En el Capítulo I, se describe al negocio, en este caso será la iglesia Roca Eterna, en la que nos estamos basando para hacer este proyecto. Se muestran los dos procesos principales que se encontraron dentro de la iglesia y el objetivo general seguido de los específicos.

En el Capítulo II se definen los términos necesarios para comprender un poco más sobre la iglesia y cómo está organizada dentro. También se mencionan algunas definiciones técnicas sobre las tecnologías que usaremos.

El Capítulo III se encuentra el estado del arte. Se compone de dos partes, la primera parte muestran documentos de los que nos hemos apoyado para realizar algunas partes del proyecto y la segunda parte con comparaciones con software en la actualidad parecidos al que se ha desarrollado en la presente tesis.

En el Capítulo IV modelamos los dos procesos del negocio principales (eventos y miembros) y detallamos las reglas del negocio que tiene actualmente la iglesia.

El Capítulo V identificamos los requerimientos que obtuvimos a partir del capítulo anterior. También definimos que el alcance del proyecto será un módulo web cuyas funciones son descritas en el capítulo.

En el Capítulo VI podemos encontrar la arquitectura del sistema, la lista de casos de uso que identificamos, el análisis junto al diseño del sistema y también se incluye una descripción de la realización de los casos de uso más significativos: Gestionar Objetivos y Gestionar Obstáculos.

El Capítulo VII muestra las pruebas que se hicieron a los casos de uso más importantes y como resultado se puede observar el buen funcionamiento de cada una de las funciones de estos.

En el capítulo VIII se demuestra que el proyecto es viable técnica, económica y legalmente, sustentando cada uno como corresponde.

Finalmente, en el Capítulo X se muestra el tiempo que llevará a cabo la realización de este proyecto. Utilizamos la herramienta MS Proyecto y obtuvimos un total de 250 días solo trabajando de lunes a viernes.

Esta tesis es presentada para optar el grado profesional de Ingeniero Informático en la escuela de Informática, de la Facultad de Ingeniería, de la Universidad Ricardo Palma.

CAPÍTULO I: VISIÓN DEL PROYECTO

1.1. Antecedentes del problema

1.1.1. El negocio

“Roca Eterna” es una iglesia cristiana evangélica cuyo local se encuentra en Av. Melgarejo 413 Dpto. 101 Urb. Santa Patricia en el distrito de La Molina. La iglesia fue fundada el 2 de julio de 1995. Está constituida por un grupo de líderes que se denomina consistorio. Ellos son los que toman las decisiones de lo que se realiza en la iglesia. Los diferentes eventos que se realizan en la iglesia son anotados en un calendario grande que se realiza al inicio de cada año en una reunión con todos los líderes de los ministerios.

Los ingresos de la iglesia se obtienen a través de las ofrendas (de cualquier persona que está en la iglesia, tanto las visitas como los miembros oficiales) y los diezmos (el 10% del sueldo de los miembros oficiales que trabajan). El monto no es una cantidad fija, pero se mantiene dentro del siguiente intervalo 1500 – 2000 soles.

La iglesia está organizada en ministerios entre los cuales están: jóvenes, música, finanzas, etc. Cada ministerio puede tener uno o más líderes. Además de eso, tienen actividades que son muy diferentes entre sí, por lo que la organización de cada uno es bien diferente. Cada ministerio tiene dos o más miembros, y cada miembro tiene un rol dentro del ministerio.

La calidad de los eventos realizados es una variable que no se puede cuantificar con un valor entero. Existen diversas formas de medir esa calidad, en Roca Eterna, esa calidad se mide teniendo como punto de referencia el propósito por el cual se ha realizado el evento, por lo tanto, no existe una herramienta de medición hasta el momento fija de cómo medirla; sin embargo, podemos hablar de una calidad complementaria (que no es lo mismo que la calidad principal) la cual se refiere al número de personas que han calificado como bueno el evento.

Misión:

Glorificar a Dios en todo, compartiendo el amor, la misericordia y el perdón de Dios, para salvación y transformación de vidas hasta llegar a la plenitud de Cristo, desde el distrito de la Molina hasta lo último de la tierra.

Visión:

Ser una iglesia para la gloria de Dios, fiel a su Palabra, cristo céntrica, madura y pensante, que comparta un cristianismo y un servicio eclesial de calidad para Dios y la sociedad.

Organigrama:

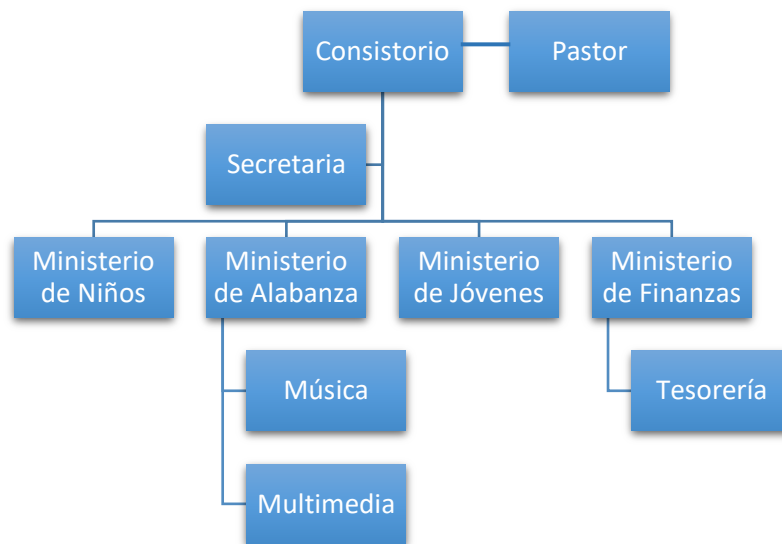


Figura 1 Organigrama de la iglesia

Fuente: Elaboración propia

1.1.2. Procesos del negocio

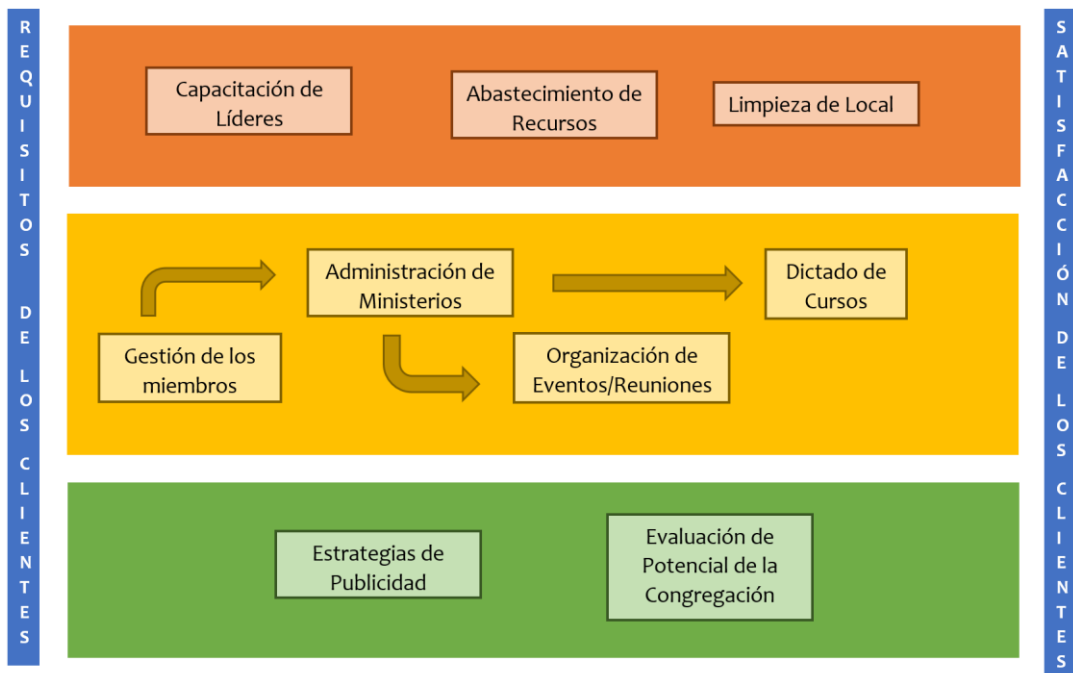


Figura 2 Diagrama de procesos de la iglesia

Fuente: Elaboración propia

1.2. Descripción del problema

El principal problema de la iglesia, en términos generales, es la administración de toda la información que ellos manejan. Llámese datos de la congregación, detalles de los eventos que se realizan, especificación sobre qué ministerios hay, roles, etc.

La iglesia cuenta con 30 miembros y 10 visitantes aproximadamente. Actualmente utilizan papeles, cuadernos, etc., para guardar los datos de las personas de la congregación. Estos datos muchas veces se pierden. A veces las funciones que delegan, como líderes de ministerio o responsabilidades en ciertas semanas, las cuales están anotadas en papeles (muchas veces ni siquiera lo anotan), se pierden o se olvidan y esto genera un retraso a la hora de buscar información o actualizarla. Hasta la fecha actual se tiene registros completos de 5 personas y 10 registros incompletos de miembros de diversos ministerios.

Para poder llevar a cabo el proceso de seguimiento de los miembros y visitantes, la colaboradora se apoya en el uso de listas en hojas impresas, donde se marca quién vino y quién no. Se usa una lista para cada reunión, y debido a las distintas

reuniones que hay en la semana, a veces se traspapelan ciertas listas y la información se llega a perder. El realizar la búsqueda de un dato, se hace complicado y puede llegar a tomar 5 minutos, y mucho más si se necesita reunir varios datos para poder realizar algún reporte, para lo cual se necesitan unos 30 minutos aproximadamente.

Dentro de la iglesia se han identificado 5 grupos de personas para poder cuantificarlos y poder tener cierto porcentaje. En el siguiente gráfico podrán observar mejor como está la situación de la iglesia actualmente.

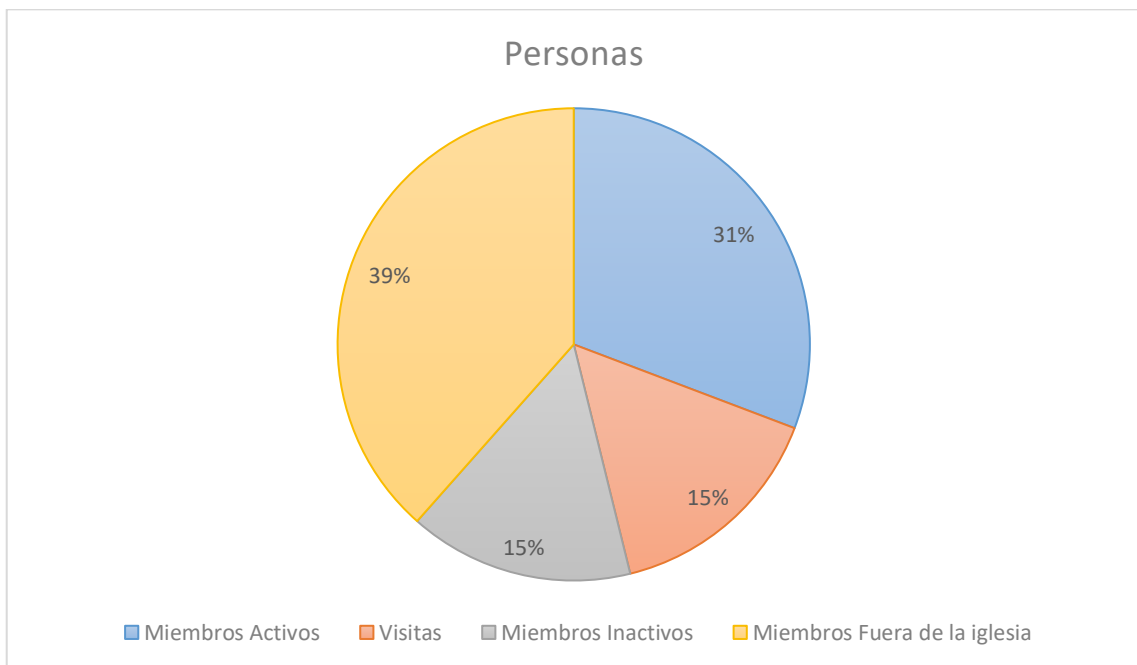


Figura 3 Porcentaje de número de personas en la iglesia según su clasificación

Fuente: Elaboración propia

El grupo más grande lo conforman los “Miembros fuera de la iglesia”, que son los miembros que se han desligado completamente de la iglesia por diversas razones y ahora congregan en otra. El que le sigue es el grupo de “Miembros activos” (miembros que cumplen roles y participan en diversas actividades de la iglesia); la mayoría perteneciente a este grupo son líderes de ministerios, lo cual debería cambiar. Como pueden observar en el gráfico, no existen muchas visitas, y junto con el número de miembros inactivos de la iglesia casi forman la mitad de los miembros que están actualmente en la congregación.

En cuanto a eventos se refiere, el problema sigue siendo el mismo. Mala administración de los datos. Un evento está compuesto de varias cosas: nombre, lugar, fecha, hora, participantes (en cierto tipo de eventos), en que consiste el evento, etc., pero pocas son las veces que se llega a especificar correctamente el detalle de alguno. A lo mucho se registrará su nombre, fecha y lugar, mientras lo otro se olvida o se prescinde de aquello. Lo que se obtiene son eventos desordenados, flácidos. No hay tiempo calculado para la creación y planificación de los eventos; sin embargo, si establecemos nuestro indicador como la cantidad de tiempo que se demora en registrar todas las características necesarias, suponiendo que toda la información relacionada a este se conozca (asistentes, lugar, fecha, etc.), se lograría guardar la información del evento en unos 5 minutos. Lo que sucede en la realidad es que esos 5 minutos pueden transformarse en horas, incluso días, ya que los programas para los eventos, los asistentes, los gastos, los encargados, los materiales a usar, etc., se demoran en buscar, elegir, registrar, etc.

Como se mencionó anteriormente, a inicios de año se realiza el calendario anual de la iglesia, donde se encuentran todos los eventos. Naturalmente, existen eventos que nacen espontáneamente y no son registrados; sin embargo, haciendo una recopilación de todos los eventos aproximadamente del año pasado, se pudo realizar el siguiente gráfico.

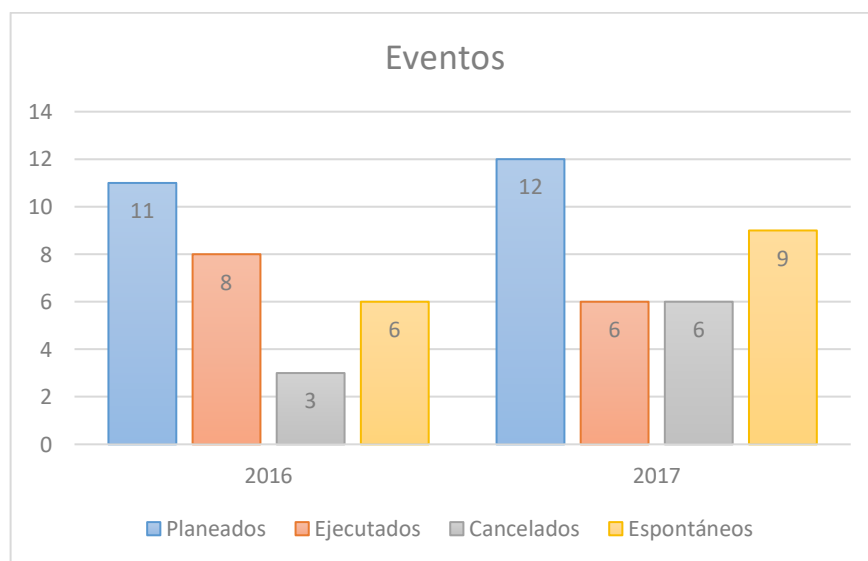


Figura 4 Cantidad de eventos realizados en los años 2016 - 2017

Fuente: Elaboración propia

Como podrán ver en el gráfico, ha habido una caída en lo que a planificación se refiere. Del 2016 al 2017 se incrementó la cantidad de eventos cancelados y se aumentaron los espontáneos. Esto revela la necesidad de la iglesia de algún método que ayude a la planificación, que se adecúe a sus necesidades y resuelva sus problemas específicos.

1.3. Objetivos

1.3.1. Marco lógico

Árbol de problemas

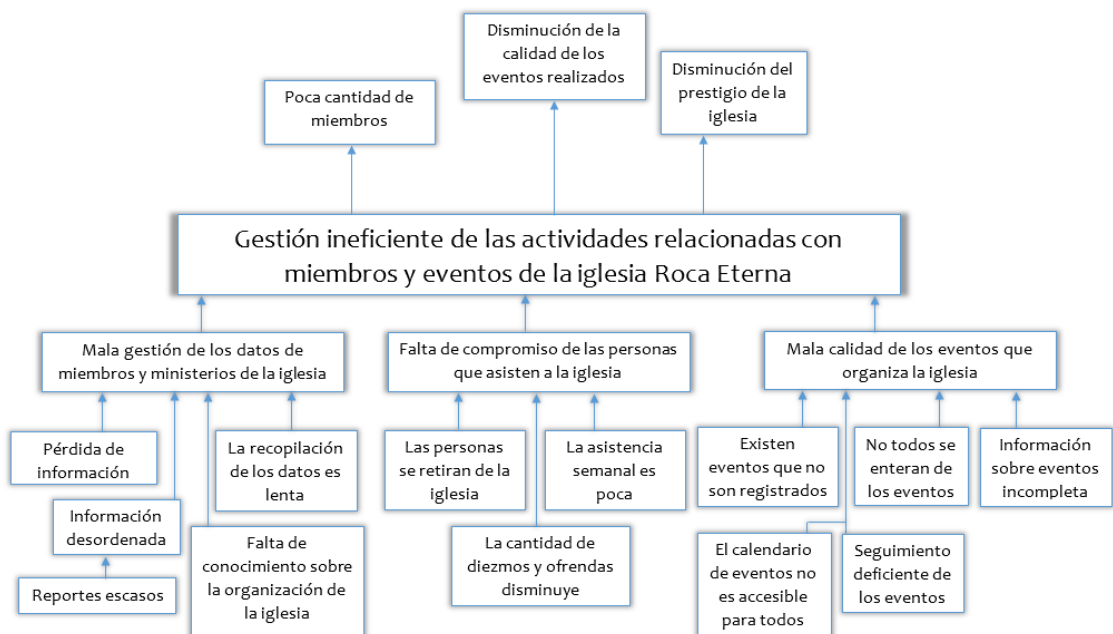


Figura 5 Árbol de problemas

Fuente: Elaboración propia

Árbol de objetivos

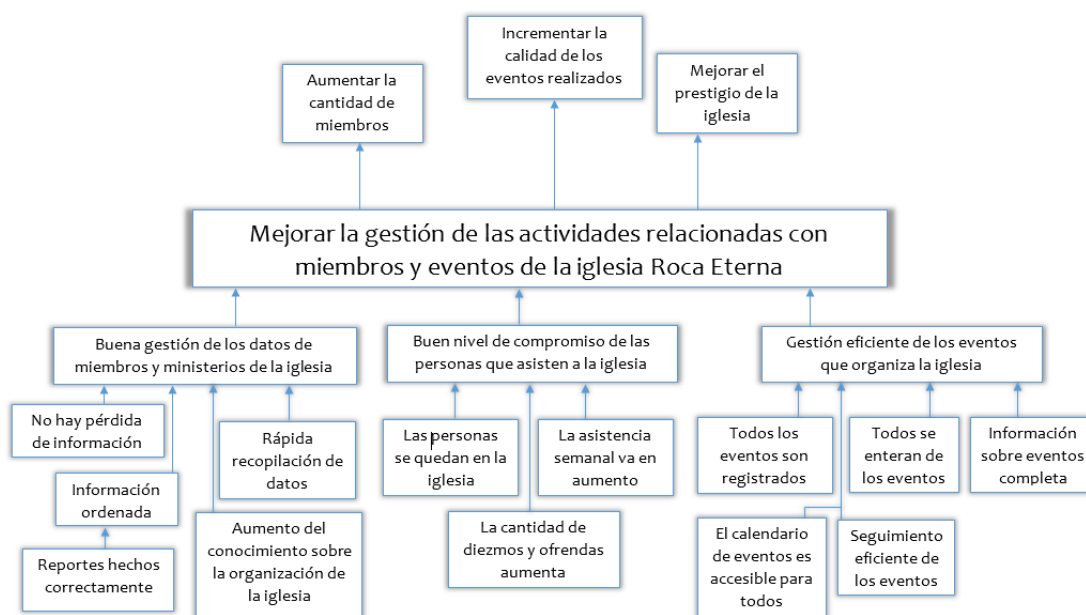


Figura 6 Árbol de objetivos

Fuente: Elaboración propia

Matriz de marco lógico

Tabla 1 Matriz de marco lógico

Resumen Narrativo	Indicadores	Medio de Verificación	Supuestos
Fin: 1. Incrementar la calidad de los eventos que realiza la iglesia 2. Agilizar la generación de reportes 3.	1. La cantidad de eventos calificados como exitosos aumentará en 30% el primer semestre 2. El tiempo de generación de reportes (cuadros estadísticos, etc.), será un 50% más rápido. 3.	1. Encuesta a los participantes de los eventos 2. El sistema mostrará un cuadro comparativo con los tiempos de creación de los reportes. 3.	1. Todos los participantes de los eventos van a la iglesia 2. El usuario del sistema sabe cómo usar el sistema y qué reporte desea realizar. 3.
Propósito: 1. Ayudar a	1. La gestión y seguimiento de los eventos y los	1. Distintos reportes creados por el	1. Los usuarios saben utilizar

mejorar la organización de la iglesia	miembros se realiza un 50% más rápido a partir del primer mes	sistema.	correctamente las funcionalidades del sistema
Componentes: 1. Gestor de miembros 2. Gestor de eventos 3. Generador de Reportes	1. Cantidad de miembros a la que se le ha hecho un seguimiento completo aumenta en 50% el primer mes. 2. El porcentaje de personas que se enteran de los eventos incrementará en 20% el primer bimestre. 3. El tiempo de generación de reportes (cuadros estadísticos, etc.), será un 50% más rápido.	1. Reporte sobre cumplimiento de los pasos que se requieren para hacer un seguimiento completo. 2. Encuesta a los miembros 3. El sistema mostrará un cuadro comparativo con los tiempos de creación de los reportes	1. Los usuarios que realicen el seguimiento han sido capacitados para esto. 2. Los miembros de la iglesia cuentan con conexión a internet. 3. El usuario que realice los reportes todo lo concerniente al reporte que desea realizar
Actividades: 1. Fase de Análisis 2. Fase de Desarrollo 3. Fase de Diseño 4. Fase de Producción	1. El costo del proyecto será 4000 soles. 2. El tiempo del proyecto será de 1 año.	1. Revisar la fecha de entrega pactada.	1. El cliente me pagará el precio que se acordó

1.3.2. Objetivo general

Desarrollar un sistema web que permita mejorar la gestión de las actividades relacionadas con la congregación y los eventos de la iglesia Roca Eterna.

1.3.3. Objetivos específicos

- a) Aumentar la calidad de los eventos realizados dentro de la iglesia, lo que involucra mejorar la organización de los mismos, así como también los comentarios positivos de los mismos.
- b) Incrementar la cantidad de miembros que estén comprometidos con la iglesia, que no se retiren y que den sus diezmos y ofrendas cuando deben.
- c) Mejorar la gestión de los datos de la iglesia que involucran a los miembros esporádicos, frecuentes y a sus respectivos ministerios y roles dentro de estos.

1.4. Importancia

Este trabajo permitirá a la iglesia, una sencilla y efectiva manera de poder administrar la gran cantidad de información que maneja. Así como la seguridad de poder obtener cualquier dato en cualquier momento de forma rápida.

Esto ayudará a mejorar la organización dentro de la iglesia y mejorar los procesos dentro de ella. Ayudará a tomar decisiones, a integrar los datos y mantenerlos al alcance de todos los miembros de la congregación y a estar preparados para soportar un futuro crecimiento de la iglesia, tanto cualitativamente como cuantitativamente.

“Porque ¿quién de vosotros, queriendo edificar una torre, no se sienta primero y calcula los gastos, a ver si tiene lo que necesita para acabarla? No sea que después que haya puesto el cimiento, y no pueda acabarla, todos los que lo vean comiencen a hacer burla de él, diciendo: Este hombre comenzó a edificar, y no pudo acabar. ¿O qué rey, al marchar a la guerra contra otro rey, no se sienta primero y considera si puede hacer frente con diez mil al que viene contra él con veinte mil?” extraída de San Lucas 14:28 (Reina Valera, 1960)

1.4.1. Justificación académica

Para poder desarrollar el presente sistema, se ha hecho uso del conocimiento adquirido a lo largo de la carrera de Ingeniería informática, conocimiento que consta de: toma de requerimientos, modelado del negocio, modelado del sistema, pruebas, etc. No obstante, para poder desarrollar el sistema, se ha tenido en cuenta una competencia muy importante aprendida también a lo largo de esta carrera: “nunca dejar de aprender”. La innovación es un punto peculiar dentro de este proyecto. El uso de tecnologías del momento, junto con la aplicación de estas al campo de estudio, demuestran que no existe exclusivismo para la tecnología, ni para el conocimiento.

1.5. Beneficios del proyecto

Teniendo en cuenta el blog de Osmer Clavo (Clavo, 2013) , tenemos dos tipos de beneficios que nos brinda un sistema de información, los beneficios tangibles y los intangibles.

Parafraseando a Osmer, los beneficios tangibles son ventajas que se acreditan a la organización gracias al uso del sistema en cuestión. Estos beneficios se pueden medir en términos de ahorros de dinero, recursos en incluso tiempo utilizado para realizar determinadas tareas.

Siguiendo con dicha definición y aplicándolo a nuestro tema de tesis, los beneficios intangibles son un tipo de resultados que incluyen una mejora en el proceso de organización de los eventos, bienestar de los miembros, mejora de la imagen frente a las personas externas, etc.

1.5.1. Beneficios tangibles

- a) La cantidad de eventos calificados como exitosos aumentará en 30% el primer semestre.
- b) El tiempo de generación de reportes (cuadros estadísticos, etc.), será un 80% más rápido.
- c) El tiempo de registro de los eventos será un 80% más veloz.

1.5.2. Medición de beneficios tangibles

- a) La cantidad de eventos exitosos será medida comparando el porcentaje de éxito de los mismos antes y después de la implementación del presente proyecto.
- b) Una encuesta a los líderes de la iglesia sobre el tiempo aproximado de desarrollo de los reportes vs el tiempo que toma registrar algunos datos en el sistema (puesto que los reportes se generan automáticamente), servirá para medir dicho beneficio.
- c) Para poder comparar el tiempo de registro de los eventos, se replicarán muestras aleatorias, en un determinado rango, del tiempo que toma registrar un evento a mano, versus muestras midiendo el tiempo que toma registrar un evento utilizando el sistema.

1.5.3. Beneficios intangibles

- a) La administración de los eventos mejorará.
- b) La satisfacción de los miembros de la iglesia será mayor.
- c) La imagen pública de la iglesia será mejor.

1.6. Alcance

El sistema que se está desarrollando tiene diversos módulos y distintas funcionalidades, muchas de las cuales no se implementarán para esta presente tesis. En la siguiente tabla podrán ver los puntos que pertenecen al alcance y los que están fuera de él.

Tabla 2 *Especificación del alcance del proyecto*

Dentro del Alcance
Gestionar la información básica de las personas que asisten a la iglesia, así como de los ministerios, sus miembros y roles.
Administrar el ciclo de vida de un evento, desde su concepción, pasando por la solicitud, evaluación, planificación, difusión y posteriormente, culminación.
Desarrollar una página web para la publicación de los eventos y la información básica de la iglesia.
Fuera del Alcance
Crear un módulo para el ministerio de finanzas de la iglesia.
Administrar todos los activos que se manejan dentro de la iglesia de una forma específica.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes

2.1.1. Definiendo el término iglesia

Usualmente cuando se menciona la palabra “Iglesia”, automáticamente sale a luz la imagen de la “Iglesia Católica”. En el contexto de la tesis, el término no se usa en ese sentido. A lo que me refiero como iglesia es a la Iglesia cristiana, para dejar la ambigüedad a un lado, a la iglesia evangélica, como comúnmente se llama, o protestante.

En el siglo XV se dio la famosa “reforma religiosa” encabezada por Martín Lutero. La reforma protestante (otro nombre que se adjudica) dio origen a lo que ahora se le conoce como “iglesia evangélica o protestante”.

“La voz de Lutero al postular reformas no se limitaba a algunos aspectos parciales de la realidad de la vida eclesial de aquel tiempo (incluyendo la doctrina), sino que se refería a *todos* los aspectos en que se plasmaba aquella” (Almarza, 1984).

Aparte de las conocidas “95 Tesis” que publicó Martín Lutero en Wittenberg el 31 de octubre de 1517 (Escobar, 2011), el sacerdote publicó tres manifiestos en 1520 los cuales están considerados dentro de los documentos más importantes de la reforma. En ellos Lutero protesta por las cosas que a su juicio funcionaban mal en la iglesia y exponía las reformas que consideraba necesarias. Los tres manifiestos son:

“A la nobleza cristiana de la nación alemana”. En este manifiesto decía que todos los cristianos eran iguales y que los obispos, sacerdotes y monjes, no tenían (como creía todo el mundo) un estatus espiritual más elevado que el del resto.

“Sobre el cautiverio Babilónico de la iglesia”. Este trataba de los sacramentos, Lutero pensaba que la iglesia estaba esclavizada por los reglamentos y prácticas sacramentales.

“La justificación por la fe”. Lutero volvía a exponer su descubrimiento de la lectura de los Romanos: decía que los cristianos, si tenían fe en Jesucristo, recibían libremente la bondad de Dios. Creía que la iglesia había aprisionado el don de Dios en un sistema de reglas y leyes. Él quería volver a lo que estaba seguro que era el sencillo mensaje de esperanza del Nuevo Testamento: que la gente no tenía que “ganarse” la salvación por medio de las buenas obras (Efesios 2:8-9). (O’Neil, 1991).

De esta forma fue como nació el término protestante para referirse a aquellos que se separaban de las ideologías de la iglesia católica, para seguir lo que decía Lutero. Posteriormente estos protestantes se convertirían en evangélicos, puesto que seguían lo que decía el evangelio, seguían lo que decía la Biblia y este era su libro sagrado: La palabra de Dios.

2.1.2. Los eventos en la iglesia

El término evento proviene del latín *eventus* y, de acuerdo con el diccionario de la Real Academia Española (RAE), tiene tres grandes usos. En varios países latinoamericanos, un evento es un suceso de importancia que se encuentra programado. Dicho acontecimiento puede ser social, artístico o deportivo. Por ejemplo: “El evento de esta noche enfrentará a los dos mejores equipos del torneo”, “El próximo mes se celebrarán tres eventos en la Facultad de Derecho”, “El concierto de los Rolling Stones ha sido el evento más popular del año”, “Ya no quedan entradas para el evento de poesía del viernes”. (definición.de, s.f.)

Roca Eterna tiene registrados distintos tipos de eventos y/o reuniones. Esta misma define los eventos como una actividad que se realiza con un objetivo en concreto, ya sea crecimiento de la iglesia, confraternidad entre los miembros, recaudamiento de fondos, etc. Dentro de todos esos tipos de eventos hay algunos que tienen características similares, otros que se diferencian bastante de otros. Por ejemplo, un campamento puede ser bastante similar a un retiro, puesto que en ambas situaciones es necesario retirarse a un lugar o local distinto de la iglesia y pasar uno o más días en dicho local; mientras que un evento de recaudación es distinto a un concierto, por ejemplo, si bien es cierto que en ambos se puede recibir dinero, ambos tiene

objetivos distintos, mientras que el primero tiene como misión recaudar fondos para la iglesia, para alguna otra actividad, o para un evento que se vaya a realizar a futuro, el segundo tiene como objetivo alcanzar a más personas, integrar a los miembros actuales y evangelizar mediante la música.

Los encargados de organizar los eventos son el consistorio y los líderes de los ministerios. Ellos son los que deciden el día en que va a llevarse a cabo el evento, dónde va a realizarse, la duración, el costo si es que lo tiene, etc. Los líderes de un determinado ministerio deben solicitar la realización de un evento al consistorio y el evento pasa a ser responsabilidad de dicho ministerio. Este tiene que velar por la administración, el cumplimiento de objetivos, las finanzas, etc. También deben encargarse de la publicidad del mismo. Cada semana en el culto dominical se notifica a la congregación sobre la realización de los eventos cercanos, si la importancia del evento es alta, entonces la anticipación con la que se notifica es bastante alta, llegando a comunicar hasta seis meses antes para que los miembros puedan prepararse económicamente y también puedan gestionar y separar sus tiempos con anticipación.

2.1.3. Un miembro dentro de la iglesia

La iglesia universal está conformada por todos los cristianos alrededor del mundo, repartida en iglesias locales en cada parte del mundo. El término cristiano en este contexto significa cristiano evangélico término que también se utiliza para referirse a un “Hijo de Dios”. La biblia nos dice que no todos somos hijos de Dios, pero que todos podemos serlo si tomamos una decisión en nuestras vidas: Juan 1:12 nos dice “más a todos los que le recibieron, a los que creen en su nombre, les dio potestad de ser hecho hijos de Dios”. Por lo tanto, por definición se puede decir que todos los cristianos son hijos de Dios, sin embargo, esta definición solo puede ser ratificada en cada uno. No cualquier que ha de llamarse cristiano lo es. Pero partamos por la definición correcta de cristiano.

Cada iglesia tiene su propio reglamento en donde se estipula los requisitos que debe seguir una persona para ser miembro oficial de la iglesia. Muchas veces no solo basta con ir un par de veces y ya está. En Roca Eterna los

requisitos para ser miembro son 3: el primero es ser cristiano, el segundo es ser bautizado en una iglesia cristiana (evangélica) y el tercero es asistir regularmente a la iglesia. Este último requisito es tratado particularmente con cada persona, pero usualmente el tiempo necesario son 5 meses.

Cuando una persona se hace miembro de la iglesia adquiere nuevos privilegios y también responsabilidades que debe seguir. En cuanto a sus privilegios, está el poder pertenecer a un ministerio de servicio a la iglesia, como el ministerio de música o el ministerio de finanzas, por ejemplo. En cuanto a sus responsabilidades, se encuentra el diezmar mensualmente. Este último punto es algo delicado, puesto que de estos diezmos es de los que vive la iglesia. Con los diezmos (aparte de alguna recaudación, aunque esto no es frecuente) es con lo que se pagan los gastos públicos, los gastos por organización de eventos, el salario del pastor, etc.

Los miembros dentro de la iglesia no necesariamente deben hacer el mismo trabajo. Si bien es cierto que no todos pueden formar parte de los ministerios por cuestiones de tiempo, cada uno cumple una parte importante en la iglesia y en la vida de los demás miembros. La biblia dice en Efesios 4:11-12: “Y él mismo constituyó a unos, apóstoles; a otros, profetas; a otros, evangelistas; a otros, pastores y maestros, a fin de perfeccionar a los santos para la obra del ministerio, para la edificación del cuerpo de Cristo”

2.1.4. Responsabilidades de los líderes

La estructura de gobierno de Roca Eterna, tiene influencia de la presbiteriana, es decir: Está gobernada por Presbíteros, (ancianos), que son elegidos por la congregación. Su forma de gobierno eclesiástico se basa en la Biblia; donde los Ancianos sabios gobernaban la iglesia de Dios de ahí que la palabra presbítero significa anciano. Los ancianos de la iglesia local en conjunto con el Pastor forman una Sesión o Consistorio para proteger el correcto estado espiritual de sus creyentes. (WikiPedia, s.f.).

En Roca Eterna los miembros del consistorio tienen la labor de elegir a miembros capacitados para convertirse en los líderes de la iglesia. Organizan una reunión entre ellos y llegan a un acuerdo para establecer un nuevo líder en el ministerio. El miembro usualmente debe cumplir ciertas características

deseables en cuanto a talentos se refiere, por ejemplo: para el ministerio de música se busca alguien que sepa sobre música al menos lo suficiente como para poder guiar de una buena manera al equipo, o en el ministerio de finanzas alguien que sea organizado económicamente.

Aparte de dichas habilidades, hay otras que son más importantes al momento de decidir por un líder de ministerio. La biblia nos dice qué fue lo que Dios le dijo a Moisés a la hora de elegir líderes para guiar al pueblo de Israel. Éxodo 18:21: “Escoge tú de entre todo el pueblo varones de virtud, temerosos de Dios, varones de verdad, que aborrezcan la avaricia; y ponlos sobre el pueblo por jefes de millares, de centenas, de cincuenta y de diez”.

Estos líderes están obligados a responder por el crecimiento de los miembros de su propio ministerio, para eso organizan eventos, reuniones, estudios, etc. Él o los líderes son quienes designan los roles dentro de su ministerio, así como también son responsables de establecer una comunicación estrecha con el consistorio para suplir cualquier necesidad que se suscite.

2.1.5. Crecimiento de la iglesia

Según el libro Una Iglesia con Propósito, el crecimiento de la iglesia está definido por dos factores: la calidad de sus miembros y la cantidad de miembros que tiene. El autor señala que la calidad es definida por sí sola: el grado de compromiso y fidelidad hacia Dios que tiene un miembro y la cantidad la define no solo como el número de personas en la congregación, sino como el número de miembros de calidad que tiene una iglesia. (Warren, 1995)

El crecimiento de la iglesia se logra cuando los miembros de la iglesia están comprometidos con la obra. Desde el consistorio hasta los miembros de la congregación. 1 Corintios 12:12 dice: “Porque, así como el cuerpo es uno, y tiene muchos miembros, pero todos los miembros del cuerpo, siendo muchos, son un solo cuerpo, así también Cristo”. El compromiso de cada uno de los miembros es crucial para poder llegar a un crecimiento en la iglesia, no solo material sino también espiritual.

La iglesia no solo crece gracias a una persona, no crece solo si una persona crece espiritualmente. El crecimiento de la iglesia está dado por todos los

miembros de la iglesia, si hablamos de islas en la iglesia, se estaría negando la veracidad del versículo anterior. El crecimiento de la iglesia en cuanto a calidad se refiere, se puede lograr gracias a la educación en cuanto a temas de la biblia se refiere. La iglesia debe encargarse de organizar cursos, dictar conferencias, brindar talleres sobre evangelización o consejería, comunicar sobre distintos cursos en otros lugares, etc. Todo para que los miembros puedan adquirir mayor conocimiento y sobre todo para que puedan poner en práctica lo aprendido.

El crecimiento de la cantidad se logra mediante la evangelización. Sin embargo, no acaba ahí. Así como mencionamos en el principio, cantidad no significa solo cantidad de personas que asisten a la iglesia. “Llenar un salón con gente que canta, aplaude y escucha una predicación no significa que la iglesia ha crecido; solo significa que la asistencia ha aumentado” (Burkholder, 2018)

Ambas partes deben ir de la mano, empiezan a llegar nuevas personas a la iglesia y en ese punto se inicia el proceso para edificar a las nuevas personas, para que crezcan en conocimiento y madurez espiritual. Dicho proceso nunca termina de realizarse, sin embargo, llega un punto donde se puede decir que el miembro ya es maduro, en ese punto recién se puede decir que la iglesia ha crecido en cantidad.

2.2. Soporte teórico de la investigación

Existen diversas tecnologías que se utilizarán para el desarrollo de este proyecto de tesis. Tecnologías que actualmente son bastante conocidas por todas las personas que están trabajando el área de software, de informática, de sistemas. Seguidamente están las definiciones de algunas tecnologías presentes en este sistema.

2.2.1. Aplicación Web

Una aplicación Web es cualquier aplicación que es accedida vía Web por una red como internet o una intranet.

En general, el término también se utiliza para designar aquellos programas informático que son ejecutados en el entorno del navegador o codificado con algún lenguaje soportado por el navegador; confiándose en este mismo para que reproduzca la aplicación.

Una de las ventajas de las aplicaciones Web cargadas desde internet (u otra red) es la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software en, potencialmente, miles de clientes. También la posibilidad de ser ejecutadas en múltiples plataformas por la fácil portabilidad de estas aplicaciones en los navegadores Web. (Alegsa, 2017)

Las aplicaciones web dinámicas son mucho más complejas a nivel técnico. Utilizan bases de datos para cargar a información, y estos contenidos se van actualizando cada vez que el usuario accede a la web app. Generalmente cuentan con un panel de administración (llamado CMS) desde dónde los administrados pueden corregir o modificar los contenidos, ya sean textos o imágenes.

Existen muchos lenguajes de programación para el desarrollo de aplicaciones web dinámicas. Los lenguajes PHP y ASP son los más comunes porque permiten una buena estructuración del contenido.

El proceso de actualización es muy sencillo y ni siquiera necesita entrar en el servidor para modificarlo. Además, que permite implementar muchas funcionalidades como foros o bases de datos. El diseño y no solo el contenido puede modificarse al gusto del administrador. (Yeeply, 2018)

2.2.2. MySql Workbench

MySQL Workbench es un software libre creado por la empresa Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL.

Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, procedimientos almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el modelo en desarrollo con la base de datos real. Se puede realizar una ingeniería directa e ingeniería inversa para

exportar e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de seguridad con MySQL Administrador.

MySQL Workbench puede generar también el guión necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los modelos de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5.x (Aranibar, 2011)

2.2.3. PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. (Álvarez, 2001)

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, etc.

¿Cómo trabaja PHP? El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo <http://www.aprenderaprogramar.com/index.php> en un navegador web como Internet Explorer, Firefox o Chrome, ¿qué ocurre? Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es: Petición de página web al servidor --> El servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores, etc. --> El servidor responde enviando una página web “normal” (estática) pero cuya creación ha sido dinámica (realizando procesos de modo que la página web devuelta no siempre es igual). (Aprende a Programar, s.f.)

2.2.4. Laravel

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el “código espagueti” y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP. (García, 2015)

Laravel es un framework joven con un gran futuro. Cuenta con una comunidad llena de energía, documentación atractiva de contenido claro y completo; y, además, ofrece las funcionalidades necesarias para desarrollar aplicaciones modernas de manera fácil y segura. En el 2011 su creador, Taylor Otwell, lo mostró por primera vez al mundo. Por lo tanto, se trata de un framework con un enfoque fresco y moderno; ya que es bastante joven. Está hecho para arquitectura MVC y resuelve necesidades actuales como manejo de eventos y autenticación de usuarios. Además, cuenta con un código modular y extensible por medio de un administrador de paquetes y un soporte robusto para manejo de bases de datos. No importa si eres un experto en PHP o si son tus primeros pasos; cuando lo conozcas, sabrás que Laravel es el framework que estabas buscando para tus proyectos PHP. (Reicek, 2015)

2.2.5. Enterprise Architect

Enterprise Architect es una herramienta de uso muy sencillo, que aborda el diseño y análisis UML y cubre el desarrollo de software desde la captura de requerimientos a lo largo de las etapas de análisis, diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Además, permite generar documentación e informes flexibles y de alta calidad.

Las bases de Enterprise Architect están sustentadas en la especificación de UML 2. Usa Perfiles UML para extender el dominio de modelado, mientras que la validación del modelo asegura la integridad del proyecto. Combina los procesos de negocio, información y flujos de trabajo en un modelo usando las extensiones gratuitas para BPMN y Eriksson-Penker. (Rambla informática, 2014)

2.2.6. RUP

La metodología RUP, abreviatura de Rational Unified Process (o Proceso Unificado Racional), es un proceso propietario de la ingeniería de software creado por Rational Software, adquirida por IBM, proporcionando técnicas que deben seguir los miembros del equipo de desarrollo de software con el fin de aumentar su productividad en el proceso de desarrollo. La metodología RUP utiliza el enfoque de la orientación a objetos en su diseño y está diseñado y documentado el uso de la notación UML (Unified Modeling Language) para ilustrar los procesos en acción. Utiliza técnicas y prácticas probadas comercialmente. (Metodoss, s.f.)



Figura 7 Ciclo de vida del RUP

Fuente: (Metodoss, s.f.)

2.2.7. UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación. Es comparable a los planos usados en otros campos y consiste en diferentes tipos de diagramas. En general, los diagramas UML describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene

UML no es un lenguaje de programación, pero existen herramientas que se pueden usar para generar código en diversos lenguajes usando los diagramas UML. UML guarda una relación directa con el análisis y el diseño orientados a objetos.

Los lenguajes orientados a objetos dominan el mundo de la programación porque modelan los objetos del mundo real. UML es una combinación de varias notaciones orientadas a objetos: diseño orientado a objetos, técnica de modelado de objetos e ingeniería de software orientada a objetos.

UML usa las fortalezas de estos tres enfoques para presentar una metodología más uniforme que sea más sencilla de usar. UML representa buenas prácticas para la construcción y documentación de diferentes aspectos del modelado de sistemas de software y de negocios. (LucidChart, s.f.)

CAPÍTULO III: ESTADO DEL ARTE

3.1. Introducción

La evolución de los eventos a lo largo de la historia es algo que usualmente pasa desapercibida, y que, sin embargo, es muy importante y ha tenido un impacto enorme en el mundo actual. Desde su significado en el diccionario de la RAE, hasta la cantidad de tipos de eventos que existen. Los eventos son parte fundamental del día de hoy y debido a eso les muestro un poco de su historia. He aquí algunas definiciones:

Los eventos surgen de la necesidad natural que tenemos por reunirnos con nuestros semejantes y pertenecer a algún colectivo, es por esto que la evolución de la industria de reuniones ha dejado huella en la historia de la humanidad. (Carlos, 2016)

Evento se refiere a cualquier acontecimiento, circunstancia, suceso o caso posible. Así, se dice eventualmente o ante todo evento en previsión de algo que, conjetural o previsiblemente, pudiera ocurrir en una circunstancia determinada y es generalmente un hecho imprevisto. (Ecu.red, s.f.)

Hagamos historia... hasta 1973 si buscábamos en un diccionario de lengua española la acepción de la palabra rezaba "*evento: hecho que puede ocurrir o no*". Por la evolución de la industria de los eventos y el crecimiento de empresas privadas dedicadas a la organización de los mismos, la R.A.E. (Real Academia Española) se sintió en la necesidad de reconocer el fenómeno económico que estaba generando en los 70 este sector, por ello autorizó un 2do significado. "Evento es un acontecimiento previamente organizado". (Anzilutti, s.f.)

Cómo podemos ver en esta última definición, fue necesario redefinir el concepto de evento. No es necesario ser historiador para tener una noción de cuanto han cambiado los eventos. Tanta ha sido su complejidad, que fue oportuno clasificarlos según sus características. Para esta presente tesis nuestra definición de eventos serán los eventos sociales.

Son los que nacen primero, ¿por qué? porque desde tiempos bíblicos el hombre buscó la compañía de sus pares, sea para celebrar sus alegrías, sea para compartir sus momentos de tristeza, pues como decía Platón, el hombre tiene “*apetitus societatis*”: hambre de sociedad, nadie, ninguno de nosotros soportaría llevar adelante una vida sin compañía. Es el objetivo de reunirse para compartir la esencia misma de este tipo de eventos, por ende, carecen de un fin económico. Dicho esto, los eventos sociales hoy día son: bodas, cumpleaños, recepciones, velatorios, conmemoraciones de fechas patrias, etc. (Anzilutti, s.f.)

3.2. Casos revisados

3.2.1. Sistema de información para la gestión de la membresía de iglesias cristianas (Anillo Farelo & Pájaro Martínez, 2014)

El autor indica que una iglesia es similar a cualquier organización en cuanto a funcionamiento se requiere, y aunque el éxito o fracaso de la misma depende de varios factores (incluyendo espirituales), el factor humano es uno muy importante.

Según el autor, es un hecho desafortunado que la mayoría de los pastores que dirigen las iglesias actualmente, aunque tengan la capacidad para dirigir espiritualmente una congregación cristiana, no tiene la preparación para dirigirla a un nivel organizacional. Cada vez que la iglesia realiza una actividad, está generando información, y esa información puede ser útil para el buen funcionamiento de las futuras tareas.

El autor menciona que algunas iglesias creen que la iglesia no debe manejarse como lo hacen todas las personas, que no deben preocuparse por el tema organizacional por son temas seculares. Mientras otras iglesias creen que es necesario, no tiene siempre los recursos a su alcance para poder capacitarse en administración o llevar cursos, invertir tiempo, etc., para una buena aplicación de técnica organizacionales.

Por esta razón desarrolla un sistema que permita llevar el registro de una “hoja de vida” para cada miembro con distinta información en ella y un registro de distintos aspectos dentro de la iglesia:

- Evolución de un miembro en la congregación.
- Procesos a los cuales ha sido sometido un miembro.

- Actividad o inactividad de un miembro.
- Cumplimiento de aportes financieros.
- Registro de los eventos realizados.
- Registro de los grupos celulares, ministerios, escuela bíblica.
- Registro de la asistencia a las reuniones.
- Realizar un seguimiento a los visitantes.

En el diagrama de actores realizado, el autor muestra una división de funciones claramente establecida. Realiza una especialización del “Miembro líder” que vendría a ser el principal y luego gracias al sistema le otorga permisos para realizar los distintos casos de uso que ha identificado.

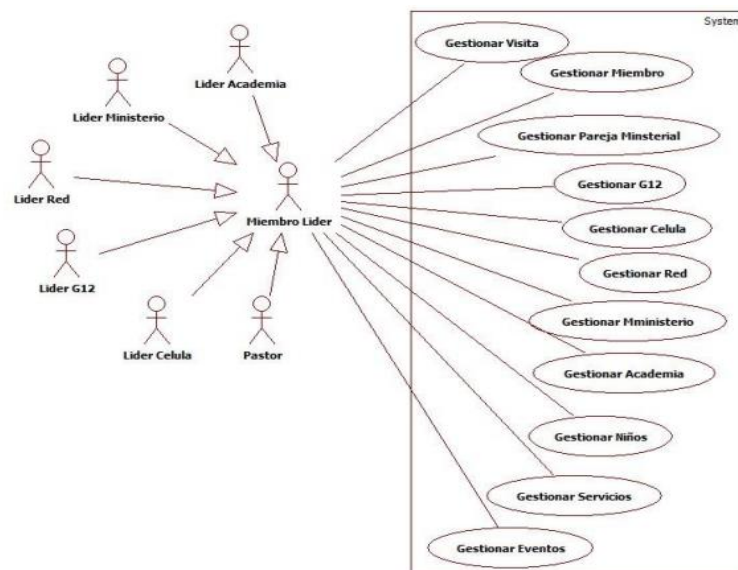


Figura 8 Diagrama de CUS

Fuente: (Anillo Farelo & Pájaro Martínez, 2014)

Utilidad para la tesis: Para la presente tesis se utilizará un poco la estructura del diagrama de actores mostrado y se adecuará al funcionamiento de la iglesia Roca Eterna. También he creído conveniente utilizar lo que el autor llama “estado activo o inactivo de un miembro” y aumentar algunas opciones más, para poder identificar de una manera más precisa a cada persona que asiste a la iglesia.

3.2.2. Desarrollo De Un Sistema Web Para Administración De Eventos Y Control De Entrada Y Salida De Empleados (Moyano Mejía, 2015)

El autor ha identificado dos partes distintas en este escenario: gestionar eventos y gestionar clientes. Ha realizado dos diagramas de casos de uso.

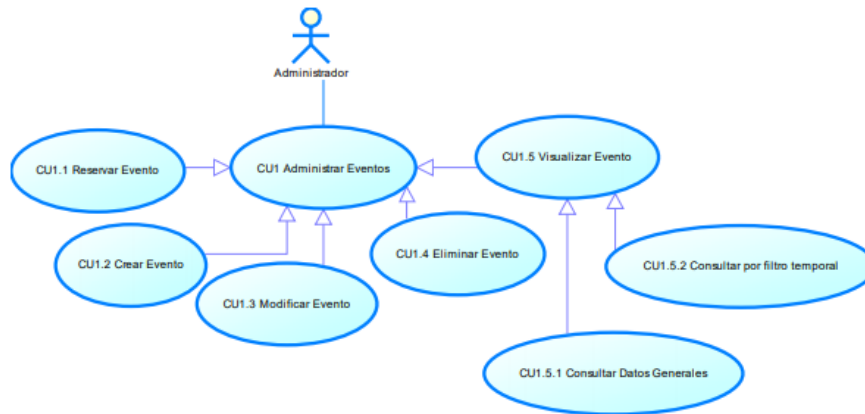


Figura 9 Diagrama de CUS Eventos.

Fuente: (Moyano Mejía, 2015)

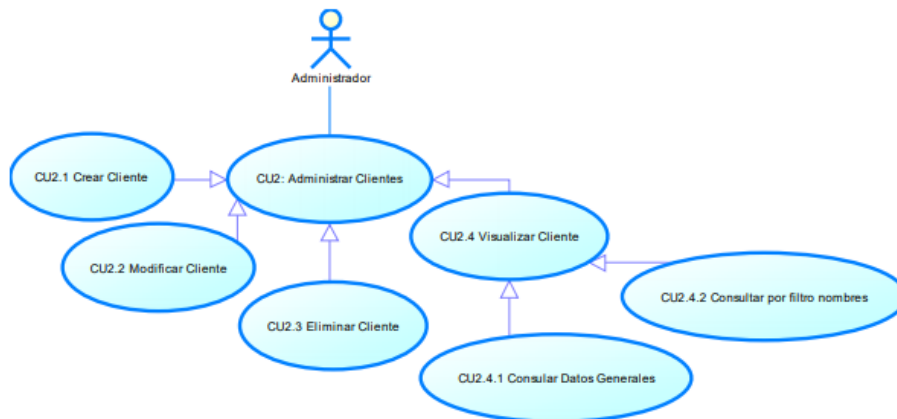


Figura 10 Diagrama de CUS Clientes

Fuente: (Moyano Mejía, 2015)

También podemos observar la pequeña estructura de tablas que ha hecho el autor para almacenar los datos de los eventos y relacionarlos con los clientes. Notemos que un cliente puede tener muchos eventos, los eventos pueden realizar en el mismo espacio y en un solo espacio hay cabida a muchos eventos y cada uno de ellos tiene un grupo que no es otra que una forma de clasificación.

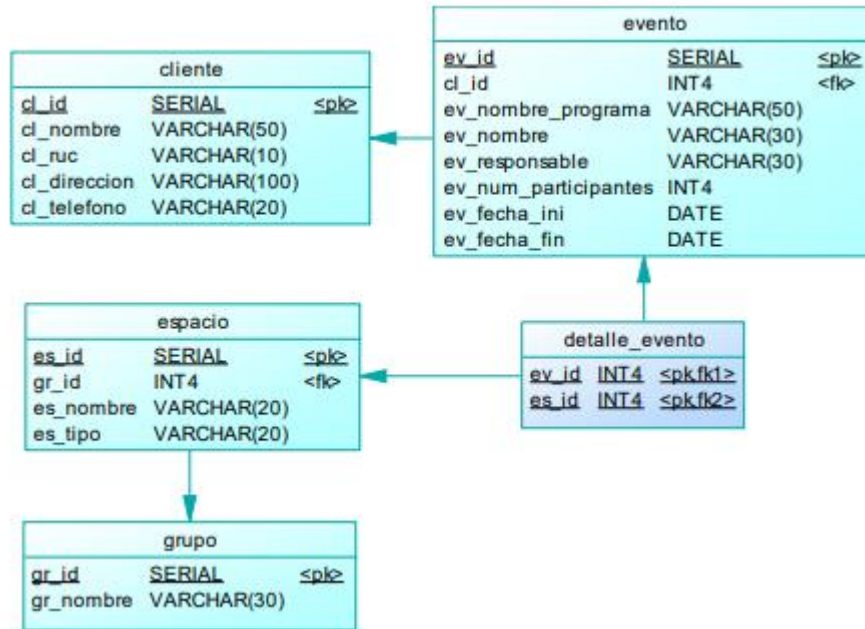


Figura 11 Diagrama de tablas

Fuente: (Moyano Mejía, 2015)

Como podremos ver a continuación ha utilizado un calendario para una mejor representación de los eventos que se reservan.



Figura 12 Prototipo calendario

Fuente: (Moyano Mejía, 2015)

Utilidad para la tesis: De este caso puedo identificar diversas funcionalidades y adaptarlas a mi trabajo. La idea de tener una reservación para los eventos la voy a transformar en una solicitud de evento, donde un líder solicita la creación de un evento a un miembro con rango superior. No se podrá eliminar un evento ya solicitado o aceptado, pero se puede cancelar

o rechazar (respectivamente) el mismo, gracias a la división de funciones que planteo.

Los clientes pasarán a ser participantes, y se los podrá asignar a los eventos, así como dejar que ellos “participen” por si mismos, también se les asignará un rol a algunos de ellos.

La utilidad del diagrama de tablas que realizo se puede extraer convirtiendo los atributos de la tabla eventos en tablas. Así se creará una tabla de participantes, una tabla donde vayan los responsables de los eventos y también respecto a los lugares donde se realizarán los eventos, los cuales se pueden clasificar en *indoor* o *outdoor*.

3.3. Comparación con otros sistemas similares

3.3.1. Iglesia Hoy

Es un software de escritorio para las iglesias cristianas de habla hispano, fue desarrollado en 1998. Tiene como función ayudar a administrar toda la información que gira en torno a los miembros de la iglesia de una manera eficiente y ordenada. (Hoy, 2017)



Figura 13 Pantalla principal de la página web Iglesia Hoy

Fuente: (Hoy, 2017)

Ventajas

- Manejo de su propia base de datos local
- Soporta múltiples usuarios

Desventajas

- Solo disponible para Windows
- Se necesita realizar un pago para una versión de más de 50 miembros

3.3.2. Hablemos de Iglesia

Es una aplicación que cuenta con parte Web y parte móvil, que se centra en la administración de los miembros de la iglesia y también un seguimiento de los cursos que han llevado. Permite la administración de ministerios, así como de las finanzas que existen dentro de la iglesia. Cuenta con un amplio sistema de reportes y permite administrar distintas iglesias.

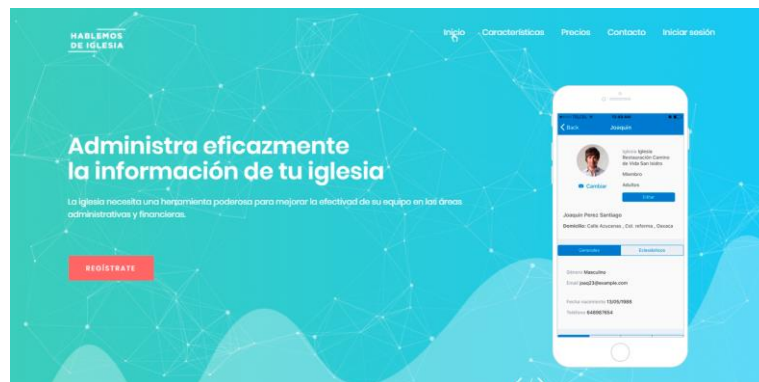


Figura 14 Pantalla principal de la página web Hablemos de Iglesia

Fuente: (Iglesia, 2017)

Ventajas

- Es multiplataforma.
- Base de Datos en la Nube

Desventajas

- Es de pago

3.3.3. Softlife

Es un software en línea que facilita la gestión de los procesos en la iglesia, adaptándose al modelo de trabajo de la iglesia, y orientado a un trabajo pastoral efectivo.

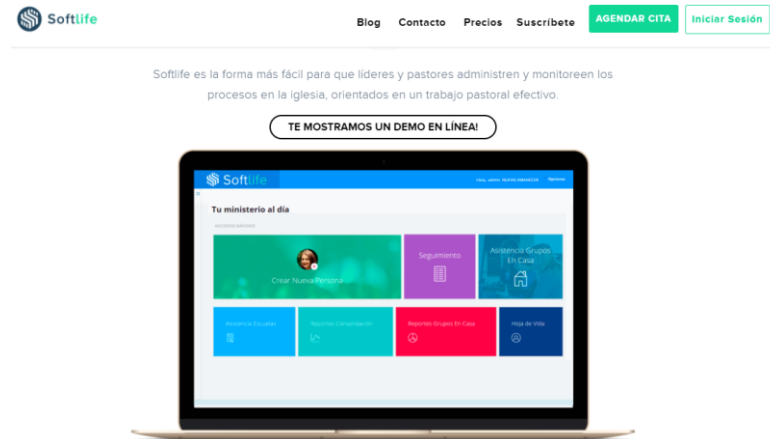


Figura 15 Pantalla principal de la página web Softlife

Fuente: (Softlife, 2017)

Ventajas

- Es multiplataforma

Desventajas

- Es de Pago

3.3.4. Tabla de Comparación

Análisis Comparativo		Rumbo Ágil		 SOFTWARE ESPECIALIZADO PARA IGLESIAS		 HABLEMOS DE IGLESIA		 Softlife	
Aspectos Funcionales	Peso	Sistema de Gestión de Eventos y miembros para la Iglesia Roca Eterna		Iglesia Hoy		Hablemos de Iglesia		Softlife	
		Funcionalidad	Total	Funcionalidad	Total	Funcionalidad	Total	Funcionalidad	Total
Administración de Miembros	3	3	9	3	9	3	9	2	6
Gestión de Eventos	3	3	9	0	0	0	0	2	6
Publicación Eventos	3	3	9	0	0	0	0	2	6
Seguimiento de Miembros	3	2	6	3	9	3	9	3	9
Generación de Reportes	3	3	9	3	9	3	9	2	6
Total Puntaje			42		27		27		33

Software Base				
Sistema Operativo	Multiplataforma	Windows Xp, Vista, 7, 8, 8.1, 10	Multiplataforma	Multiplataforma
Servidor de Base de Datos	MySQL	No Especifica	No Especifica	No Especifica
Lenguaje de Programación	PHP Laravel	PHP Laravel	No Especifica	ASP.NET

Legenda de Funcionalidades
0 = No tiene esta funcionalidad
1 = Baja (poco amigable y pocos datos)
2 = Media (amigable o con datos insuficientes)
3 = Alta (amigable y con datos suficientes)

Peso
1 = Poco importante
2 = Importante
3 = Muy Importante

Figura 16 Benchmarking

Fuente: Elaboración propia

CAPÍTULO IV: MODELADO DEL NEGOCIO

En el presente capítulo presentaremos un poco del estudio de negocio que hemos hecho a la iglesia Roca Eterna, presentaremos las reglas del negocio que existen actualmente en la iglesia, los procesos del negocio identificados. Este capítulo tiene una importancia peculiar y es que en esta sección se identifican los procesos que se van a automatizar en un futuro cercano.

4.1. Reglas del Negocio

Las reglas del negocio de una empresa son esenciales para conocer el funcionamiento de los procesos internos de esta. En el presente problema se habla de una congregación eclesial, pero no por eso deja de tener estas características.

IBM define una regla de negocio como una condición que se debe satisfacer cuando se realiza una actividad de negocio. Una regla puede imponer una política de negocio, tomar una decisión o inferir nuevos datos de datos existentes. (IBM, s.f.)

A continuación, indico cuáles son las reglas del negocio actuales que maneja la Iglesia Bíblica Roca Eterna:

4.1.1. Regla del Negocio 1

Para poder ser un miembro oficial de la congregación, una persona debe haber sido bautizada en una iglesia cristiana y ser un asistente frecuente de esta iglesia.

4.1.2. Regla del Negocio 2

Las decisiones importantes son tomadas por el consistorio junto con el pastor.

4.1.3. Regla del Negocio 3

Los miembros oficiales de la congregación deberán dar un diezmo semanal de acuerdo con los ingresos personales semanales.

4.1.4. Regla del Negocio 4

Para poder llevar a cabo un evento, se deberá notificar al consistorio, este se encargará de evaluar y aceptar o rechazar el pedido.

4.1.5. Regla del Negocio 5

Solo el líder o líderes de los ministerios puedes solicitar que se realice algún evento dentro de la iglesia.

4.1.6. Regla del Negocio 6

No se puede retirar las pertenencias de la iglesia sin previo aviso al ministerio encargado de dichas cosas.

4.1.7. Regla del Negocio 7

Cada ministerio puede tener su propia caja chica para gestionar las finanzas de dicho ámbito.

4.1.8. Regla del Negocio 8

Para gastos mayores, el ministerio correspondiente deberá solicitar al consistorio el monto adecuado y deberá justificar las razones por las cuáles necesita dicho dinero.

4.1.9. Regla del Negocio 9

Los líderes de los ministerios deberán asistir a todas las clases para líderes que se lleven a cabo dentro de la iglesia.

4.1.10. Regla del Negocio 10

Para las reuniones congregacionales dedicadas a la toma de decisiones, se tomará asistencia a dicho reunión y se procederá solo si hay un aforo del 50% de miembros más uno.

4.1.11. Regla del Negocio 11

El plan anual de la iglesia se define a inicios de año y debe contener todos los eventos y reuniones que se celebraran a lo largo del año.

4.1.12. Regla del Negocio 12

La secretaria deberá llevar la asistencia sobre los miembros que asisten a determinadas reuniones de la iglesia.

4.1.13. Regla del Negocio 12

Cada vez que llegue alguna visita a la iglesia, se deberá brindarle una ficha de datos para que este los llene.

4.2. Procesos del Negocio

Un proceso de negocio es cualquier medida o procedimiento que una organización sigue para conseguir un objetivo comercial mayor. Cuando se desglosa, un proceso de negocio es una serie de tareas o actividades individuales que se realizan en un orden específico. (IBM, s.f.)

En el siguiente apartado, se muestra una lista con los procesos de negocio que fueron identificados en el estudio de Iglesia Roca Eterna que se realizó para el presente proyecto.

4.2.1. Relación de Procesos del Negocio

- **Seguimiento de Miembros:** Los miembros del consistorio junto con el apoyo de la secretaria evalúan el desenvolvimiento de los miembros y visitas en la iglesia. Toman asistencia a las reuniones, verifican si alguno está llevando discipulado, identifican nuevos creyentes, nuevos miembros, etc.
- **Administración de Ministerios:** Los líderes de los ministerios evalúan la situación de cada grupo, realizan análisis FODA, trazan objetivos, metas, etc.
- **Organización de Eventos y/o reuniones:** Los líderes de los ministerios junto con el consistorio se reúnen a inicio de año para poder planear todos los potenciales eventos que se realizar a lo largo del presente. Indican la fecha y algunos propósitos si es que se tiene.
- **Dictado de Cursos:** El consistorio recluta miembros con conocimientos sólidos en la Biblia para que puedan dictar cursos para los nuevos creyentes y/o para enseñar principios de la vida cristiana.
- **Capacitación de Líderes:** El consistorio provee de un maestro que pueda capacitar a los líderes en temas de enseñanza, liderazgo, organización y cosas que resultan útiles para la instrucción de nuevos creyentes.

- **Abastecimiento de Recursos:** Cada ministerio, incluyendo al consistorio, evalúa la calidad de los recursos materiales con los que cuenta la iglesia. En caso de ser necesario adquirir algún bien material, se consulta con el ministerio de finanzas para evaluar la solicitud y aprobarla o rechazarla.
- **Limpieza de Local:** Una persona contratada por la iglesia realiza la limpieza de esta, para las reuniones dominicales o cuando se requiera sus servicios.
- **Estrategias de publicidad:** Cada ministerio es responsable de dar publicidad al mismo. Ya sea por medio de las redes sociales, o en físico. También cada domingo se anuncia la agenda de la semana y/o eventos futuros de la iglesia, frente a toda la congregación
- **Evaluación de Potencial de la Congregación:** Los líderes de los distintos ministerios evalúan (cuando lo crean necesario) a la congregación e identifican a ciertas personas que podrían formar parte del respectivo ministerio, que puedan discipular a otras personas, evangelizar, etc.

4.3. Casos de Uso del Negocio

A continuación, les muestro los casos de uso del negocio que se han identificado.

4.3.1. Relación de casos de uso del negocio

- CUN: Organizar Evento
- CUN: Realizar Seguimiento

4.3.2. Diagrama de casos de uso del negocio

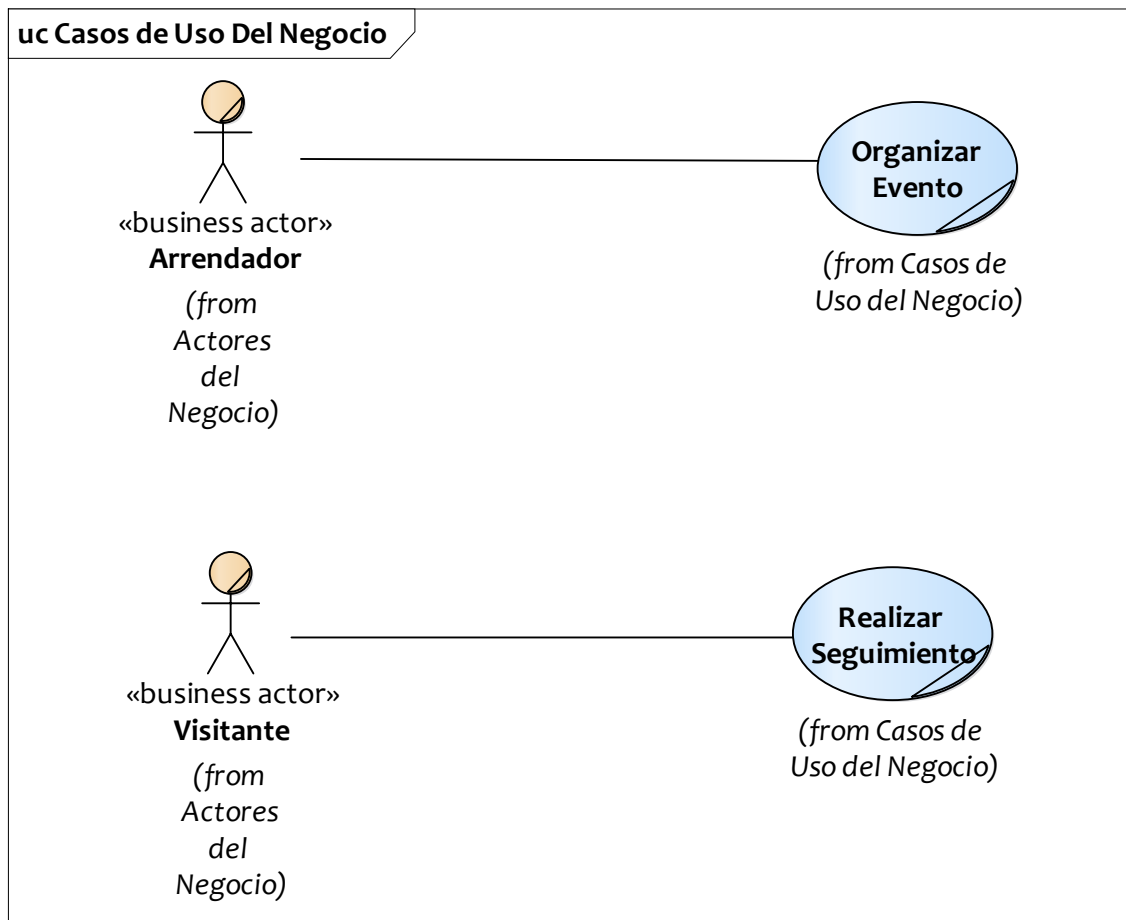


Figura 17 Diagrama de CUN

Fuente: Elaboración propia

4.3.3. Descripción de los casos de uso del negocio

- **Gestionar Evento:** Un determinado líder de un ministerio coordina con su equipo y luego habla con el pastor para proponerle el evento. El pastor es quien acepta la propuesta o la rechaza y de ser necesario llama al arrendador de un local.
- **Seguimiento de miembros:** La secretaria y los líderes se encargan de conseguir los datos de los miembros o de las visitas que asisten a las reuniones de la iglesia. En caso de los miembros la secretaria guarda sus asistencias.

4.4. Diagramas de Actividades

4.4.1. CUN Organizar Evento

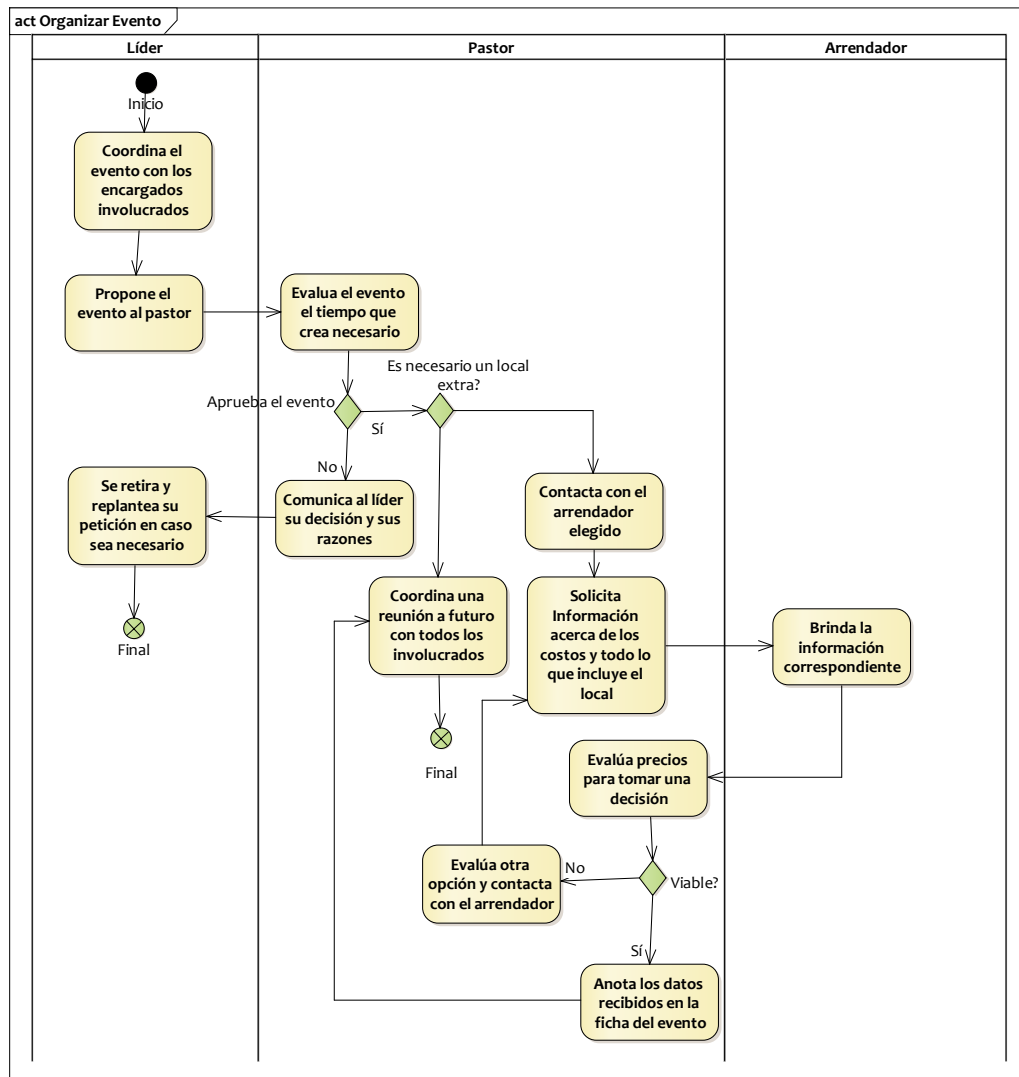


Figura 18 Diagrama de actividades del CUN Organizar Evento

Fuente: Elaboración propia

4.4.2. CUN Realizar Seguimiento

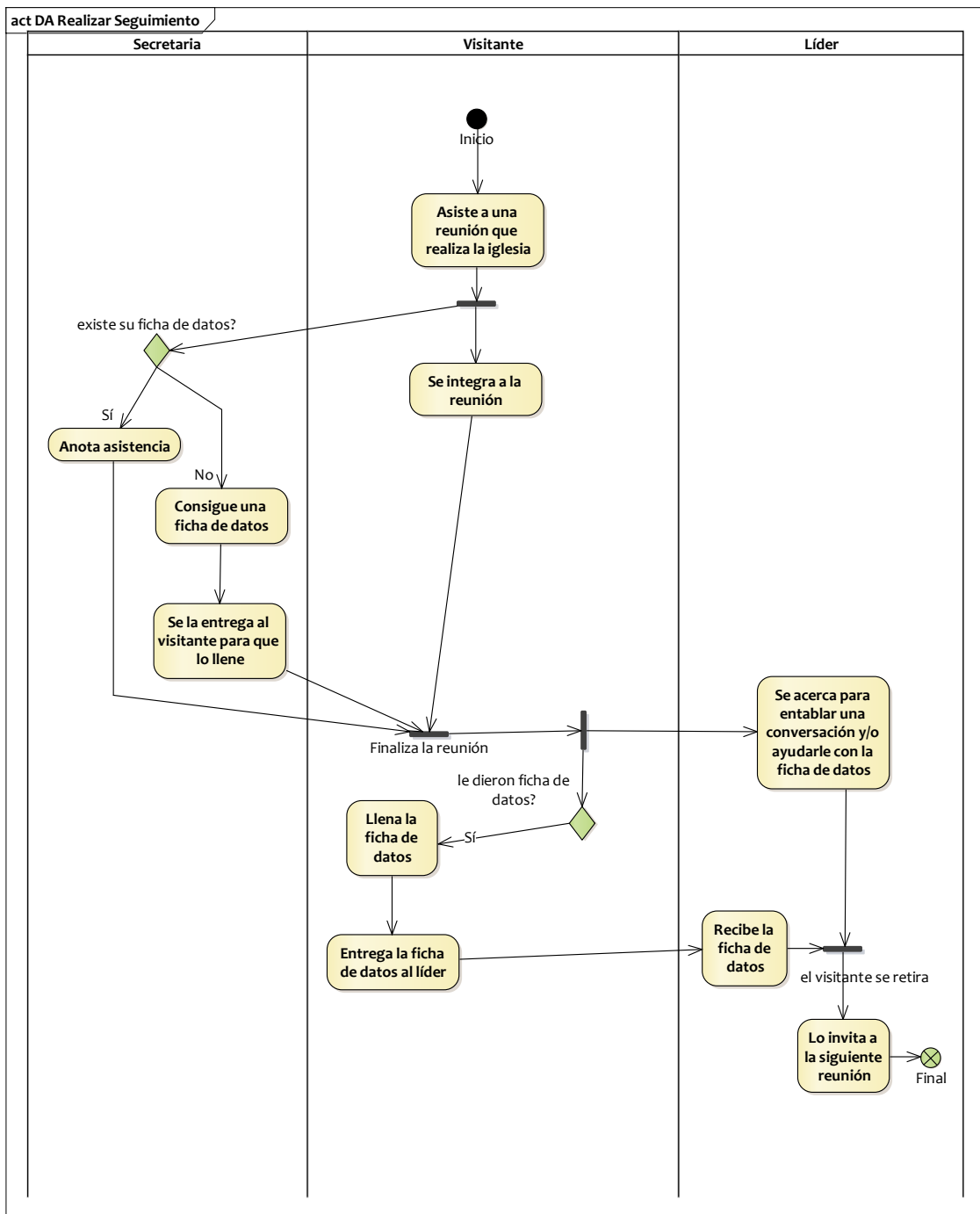


Figura 19 Diagrama de actividades del CUN Realizar Seguimiento

Fuente: Elaboración propia

4.5. Diagrama de Clases del Negocio

4.5.1. CUN Organizar Evento

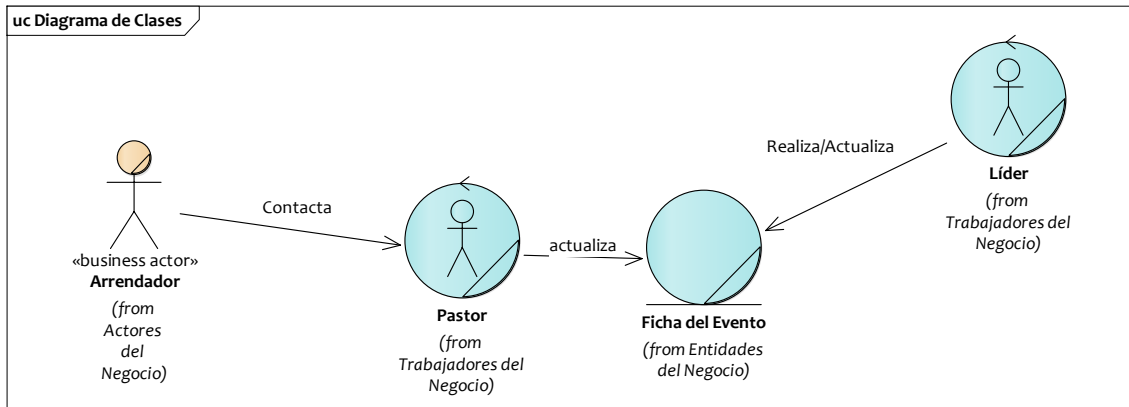


Figura 20 Diagrama de clases del negocio del CUN Organizar Eventos

Fuente: Elaboración propia

4.5.2. CUN Realizar Seguimiento

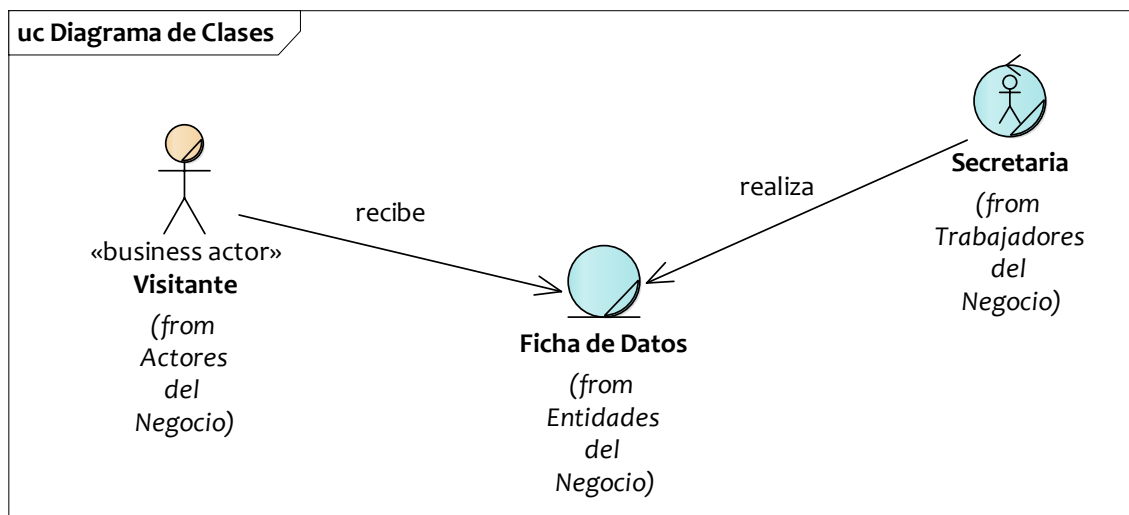


Figura 21 Diagrama de clases del negocio del CUN Realizar Seguimiento

Fuente: Elaboración propia

CAPÍTULO V: REQUERIMIENTOS DEL PRODUCTO

5.1. Requerimientos del Software

Para poder iniciar el proceso de construcción del software, una vez que se ha conocido el negocio e identificado sus procesos, el siguiente paso es la identificación de los requerimientos funcionales y los no funcionales. Los requerimientos funcionales muestran lo que deberá tener el software, a partir de estos podremos definir las funcionalidades del sistema, mientras que los requerimientos no funcionales, muestran cómo debería funcionar el sistema internamente, en términos de rendimiento e interacción con el o los usuarios.

5.1.1. Requerimientos funcionales

A continuación, se muestra la relación de los requerimientos funcionales identificados para el presente trabajo:

- **REQ1 Consultar eventos en página web:** Cualquier usuario que esté navegando por la Web, podrá ver una página con todos los eventos públicos de la iglesia, así como podrá ver el detalle de cada uno.
- **REQ2 Solicitar eventos:** Un líder puede registrar una solicitud para la planeación y posterior realización de un evento.
- **REQ3 Evaluar solicitud de evento:** Un miembro del consistorio puede evaluar el evento que está en cola (solicitados) y decidir si aceptar o rechazarla solicitud.
- **REQ4 Gestionar Eventos:** Un líder puede modificar la información del evento mientras aún no haya sido iniciado, los cambios serán notificados.
- **REQ5 Ver Calendario:** Un miembro del consistorio podrá ver en un calendario todos los eventos registrados en el sistema, modificar sus datos, cancelarlos e ir a la página de planificación de los mismos.
- **REQ6 Planificar eventos:** Los distintos líderes pueden empezar con la gestión de los eventos. Podrán agregar objetivos al evento, actividades a dichos objetivos, asignarles obstáculos a las actividades, agregar participantes y asignarles roles, añadir los objetos que serán necesarios en el evento, añadir fotos del evento, añadir las necesidades extra que se necesitan y generar un reporte final.

- **REQ7 Consultar Información Pública de la Iglesia:** Un usuario será capaz de entrar a la página de la iglesia y ver quiénes con los líderes, la misión, visión, eventos, reuniones, así como otros datos públicos de la iglesia.
- **REQ8 Gestionar Información de ministerios:** El consistorio podrá registrar los ministerios que existen actualmente en la iglesia e información relacionada a ellos, como el objetivo, el líder del ministerio, etc. El líder del respectivo ministerio podrá administrar la información de dicho ministerio. Podrá agregar nuevos miembros, asignar responsabilidades, y realizar funciones propias a dicho ministerio. También podrá ver los reportes para su respectivo ministerio.
- **REQ9 Gestionar Información de Miembros:** La secretaria podrá registrar la información de los miembros. La dirección, nombre, si es visitante, líder, miembro, etc.
- **REQ10 Participar en un evento:** Un miembro que tenga un usuario y contraseña puede logearse en la página web de la iglesia, ir al detalle de un evento y anotarse como participante del mismo.

5.1.2. Requerimientos No Funcionales

Seguidamente, les muestro los requerimientos no funcionales del sistema:

- **Usabilidad:** Estará a disposición de los usuarios un manual para poder manejar adecuadamente las funcionalidades del sistema. También tendrá las validaciones necesarias y se dispararán en el momento que el usuario ingrese data incorrecta y, asimismo, estará desarrollado de tal manera que su uso sea sencillo e intuitivo.
- **Confiability:** La disponibilidad del sistema será un 96% como mínimo. Se pretende mantener el sistema disponible las 24 horas del día.
- **Seguridad:** Las contraseñas de todos los usuarios serán encriptadas, los usuarios estarán clasificados por roles y solo podrán acceder a las funciones asignadas.

- **Rendimiento:** El sistema debe emitir las respuestas solicitadas por cada usuario en un rango de 0 a 3 segundos como máximo. También deberá tener la capacidad de acceso para varios usuarios simultáneos, teniendo como mínimo la cifra de 30 de estos.
- **Soporte:** El sistema estará diseñado de tal forma que el mantenimiento respecto a errores y a inconvenientes que se presenten, se realicen en tiempo de ejecución de este.
- **Escalabilidad:** El sistema deberá ser diseñado de tal forma que puedan agregarse funcionalidades, nuevas tecnologías y/o actualizaciones.

5.2. Casos de Uso del Sistema

5.2.1. Actores del Sistema

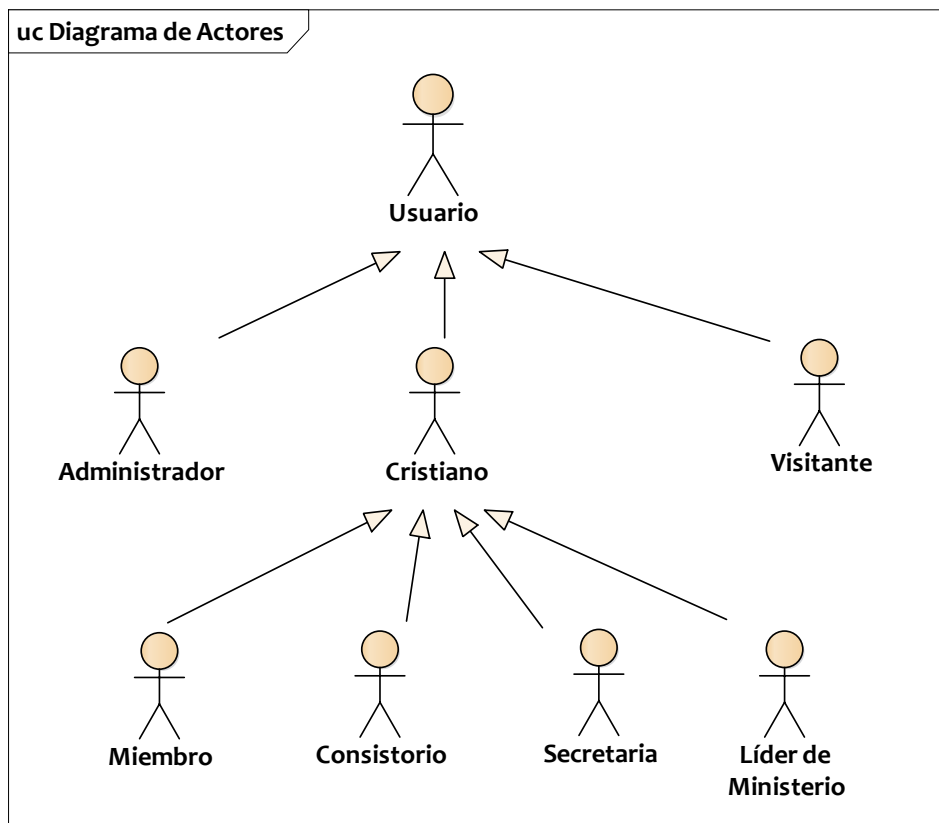


Figura 22 Diagrama de actores del sistema

Fuente: Elaboración propia

5.2.2. Diagrama de Paquetes

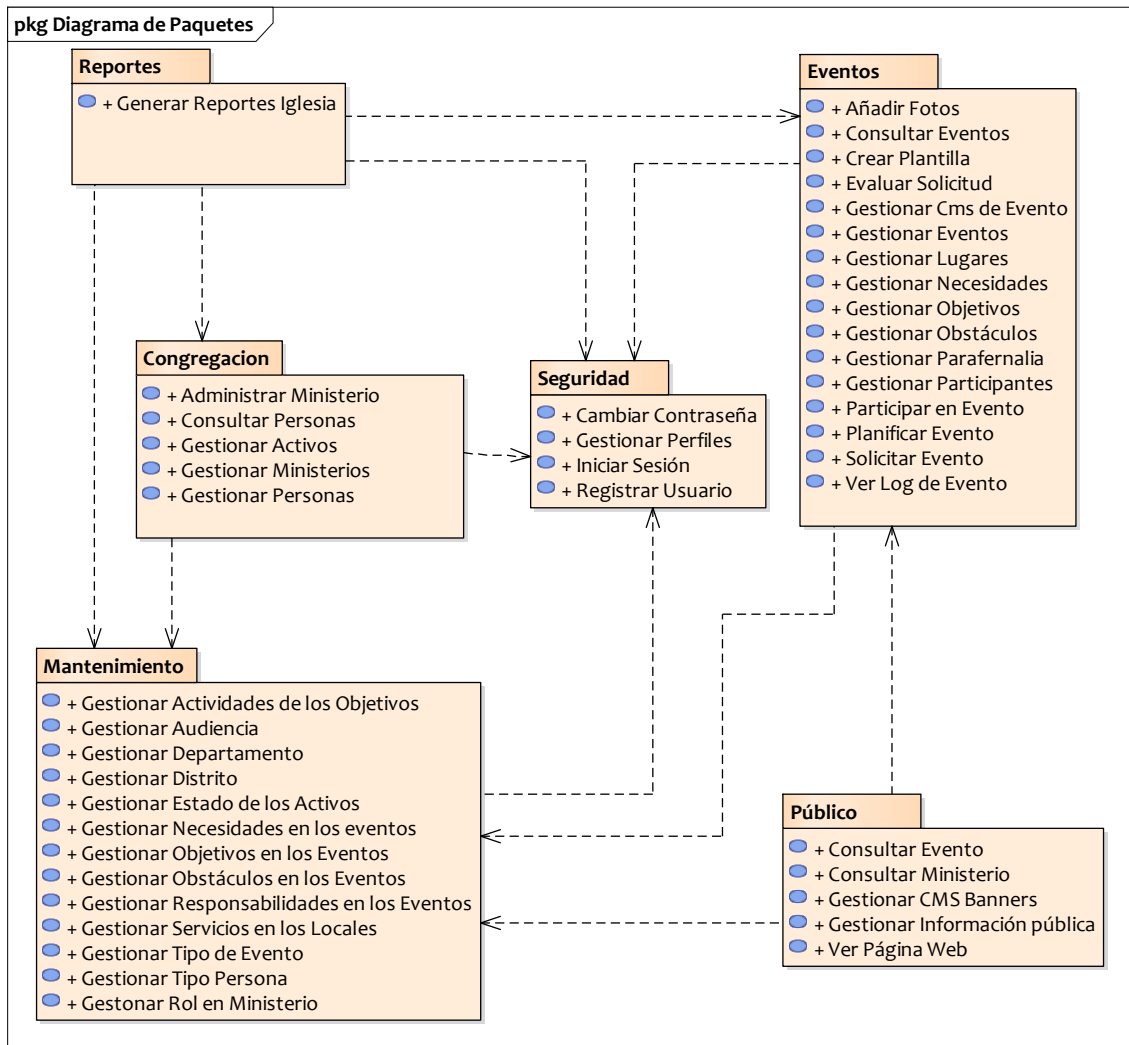


Figura 23 Diagrama de paquetes

Fuente: Elaboración propia

5.2.3. Relación de CUS

Luego del análisis realizado he podido identificar distintos casos de uso del sistema. Estos casos de uso nacieron de los casos de uso del negocio, resuelven los problemas anteriormente mencionados, desglosando los mismos en funcionalidades. En la siguiente lista se encuentran cada uno de ellos con una breve definición del mismo.

- **CUS Añadir Fotos:** El líder de un ministerio podrá añadir las fotos que desee a un determinado evento.

- **CUS Consultar Eventos:** El consistorio, la secretaria y los líderes de los ministerios podrán acceder a un buscador de los diversos eventos que hayan sido registrados en el sistema.
- **CUS Crear Plantilla:** La secretaria puede crear plantillas de los distintos tipos de eventos que existan registrados en el sistema. Podrá especificar datos como lugar, hora, público, etc. Al momento de solicitar un evento, estos datos aparecerán completos.
- **CUS Evaluar Solicitud:** El consistorio será capaz de aceptar o rechazar la solicitud de un evento solicitado por el líder de algún ministerio, así como de realizar comentarios sobre dicha solicitud.
- **CUS Gestionar Cms de Evento:** El líder de un ministerio podrá realizar cambios en la información pública del evento.
- **CUS Gestionar Lugares:** El líder podrá registrar un lugar donde se lleva a cabo un evento.
- **CUS Gestionar Necesidades:** El líder podrá añadir y quitar las cosas o servicios que se necesiten para la realización de un evento en específico.
- **CUS Gestionar Objetivos:** El líder podrá añadir los objetivos que se planean alcanzar en un evento en específico, así como darle una magnitud. También puede añadir actividades a un objetivo en particular.
- **CUS Gestionar Obstáculos:** El líder podrá especificar cuáles han sido los obstáculos de las actividades dentro del evento y si estos han hecho que el evento padezca de retrasos o no.
- **CUS Gestionar Parafernalia:** El líder podrá añadir los activos que serán necesarios para la realización de un evento en particular.
- **CUS Gestionar Participantes:** El líder podrá añadir participantes a un evento, así como asignarle distintos roles a los que así requieran.
- **CUS Participar en Evento:** El usuario podrá indicar que asistirá a un evento desde la página web siempre y cuando haya iniciado sesión.
- **CUS Planificar Evento:** El líder de un ministerio podrá acceder a la interfaz a planificación del evento, así como cerrarlo definitivamente y/o realizar un comentario acerca del mismo.

- **CUS Solicitar Evento:** El líder podrá solicitar un evento para su aprobación. Con todos los datos necesarios y también un comentario de la razón de la solicitud.
- **CUS Gestionar Eventos:** El miembro del consistorio podrá ver el calendario y todos los eventos que se hayan puesto en él, así como modificar sus datos y acceder a su planificación.
- **CUS Ver Log de Evento:** El consistorio podrá ver las acciones que se han realizado a lo largo de ciclo de vida de un evento, así como el usuario que las realizó, la fecha y hora de cada una de ellas.
- **CUS Consultar Evento:** El usuario podrá ver la información de un evento que vaya a realizarse o haya sido realizado, en la página web de la iglesia.
- **CUS Consultar Ministerio:** El usuario podrá ver en la página web la información de los ministerios, así como los líderes de estos.
- **CUS Gestionar CMS Banners:** La secretaria podrá elegir los banners que se han elegido para la página web de la iglesia.
- **CUS Gestionar Información Pública:** El consistorio podrá modificar la información pública de la iglesia, como la visión, misión, descripción, etc.
- **CUS Ver página Web:** El usuario podrá ver la página pública de la iglesia y enterarse de las novedades que hay en ella.
- **CUS Generar Reportes Iglesia:** Los diversos usuarios podrán ver los reportes que existen actualmente en el sistema.
- **CUS Administrar Ministerio:** El líder de su respectivo ministerio puede registrar miembros en dicho ministerio, así como otorgarle un rol dentro de este.
- **CUS Consultar Personas:** El consistorio, la secretaria y los líderes de los ministerios podrán acceder a un buscador de las distintas personas que hayan sido registradas en el sistema.
- **CUS Gestionar Activos:** La secretaria será capaz de registrar, actualizar, eliminar y ver los distintos activos que existen dentro de la iglesia de una forma básica.

- **CUS Gestionar Ministerios:** El miembro del consistorio podrá ser capaz de registrar ministerios y el líder de dicho ministerio podrá cambiar la información solo de su ministerio.
- **CUS Gestionar Personas:** La secretaria, el consistorio y los líderes de los ministerios, serán capaces de registrar, actualizar y ver información de las personas que asisten a la iglesia.
- **CUS Cambiar Contraseña:** El usuario será capaz de cambiar su contraseña cuando lo desee.
- **CUS Gestionar Perfiles:** El administrador del sistema será capaz de gestionar los roles de los usuarios que han sido registrados en el sistema.
- **CUS Iniciar sesión:** El usuario usará sus credenciales para ingresar al sistema y a las funcionalidades según el rol que le hayan asignado.
- **CUS Registrar Usuario:** El administrador del sistema podrá registrar usuario y asignarlos a un miembro registrado en el sistema.
- **CUS Gestionar Actividades de los Objetivos:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver las distintas actividades que han de realizarse dentro de los objetivos de los eventos.
- **CUS Gestionar Audiencia:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver el público a quien se dirigen los distintos eventos que se realizan.
- **CUS Gestionar Estado de los Activos:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los distintos estados que tienen los activos de la iglesia.
- **CUS Gestionar Necesidades en los Eventos:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver las distintas necesidades que tienen un evento y precios de dichas necesidades si es que lo tienen.
- **CUS Gestionar Objetivos en los Eventos:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los distintos objetivos que se planteen en los eventos.
- **CUS Gestionar Responsabilidades en los Eventos:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los distintos obstáculos que pueden tener las actividades de los eventos.

- **CUS Gestionar Responsabilidades en los Eventos:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver las distintas responsabilidades que puede tener los participantes en los eventos.
- **CUS Gestionar Servicios en los Locales:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los servicios que se brindan en los locales donde se realizan los eventos.
- **CUS Gestionar Departamento:** La secretaria será capaz de registrar, actualizar, eliminar y ver los distintos departamentos de donde vienen los miembros de la iglesia.
- **CUS Gestionar Distrito:** La secretaria será capaz de registrar, actualizar, eliminar y ver los distintos distritos donde viven los miembros de la iglesia.
- **CUS Gestionar Tipo de Evento:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los distintos tipos de eventos que maneja la iglesia.
- **CUS Gestionar Tipo de Persona:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los distintos tipos de persona que se maneja dentro de la iglesia.
- **CUS Gestionar Rol en Ministerio:** El Líder de un ministerio será capaz de registrar, actualizar, eliminar y ver los distintos roles que pueden tener los miembros de los ministerios.

5.2.4. Diagrama General de CUS

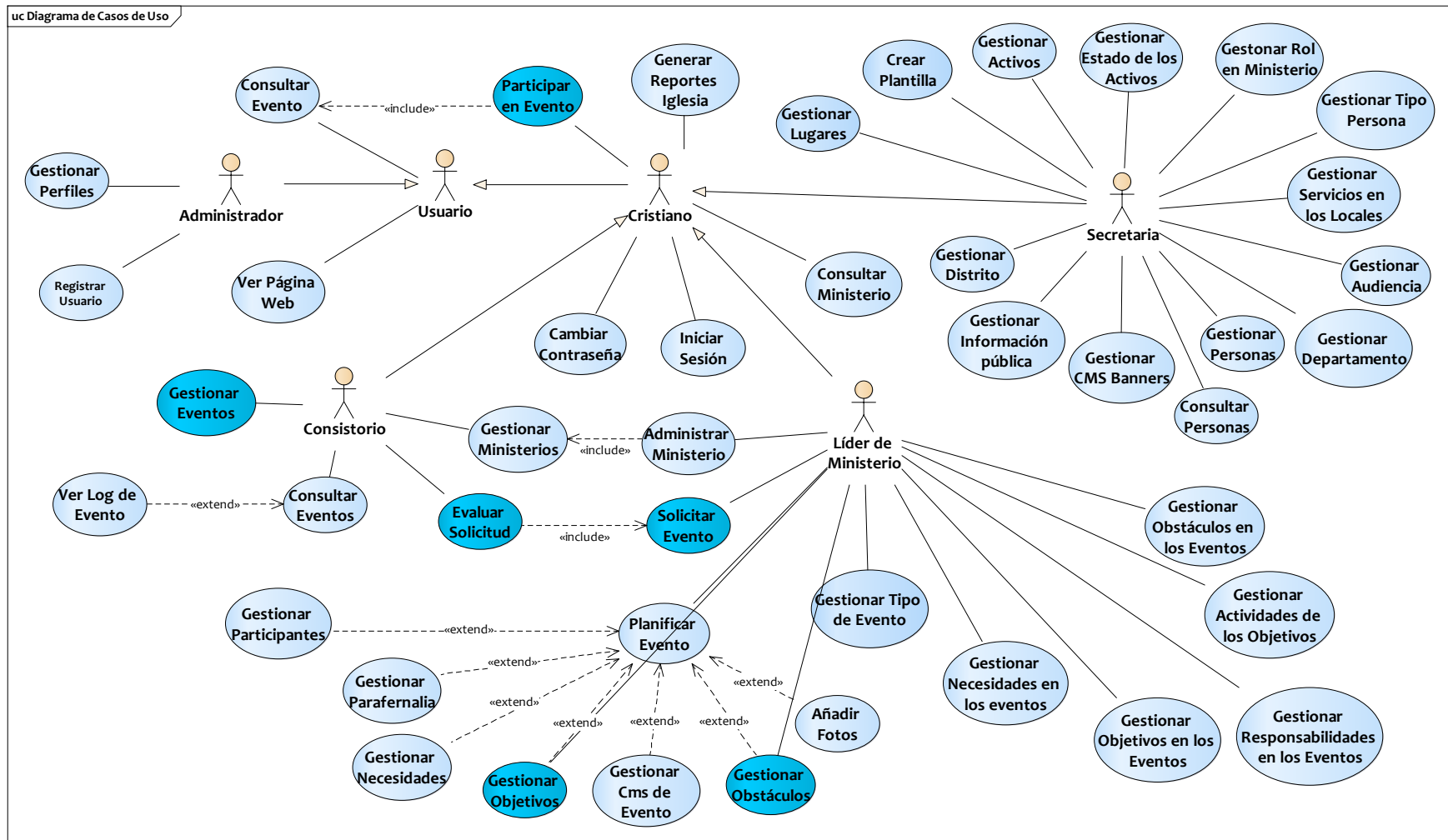


Figura 24 Diagrama general de CUS

Fuente: Elaboración propia

5.2.5. Diagrama de CUS por paquetes

- Paquete Eventos

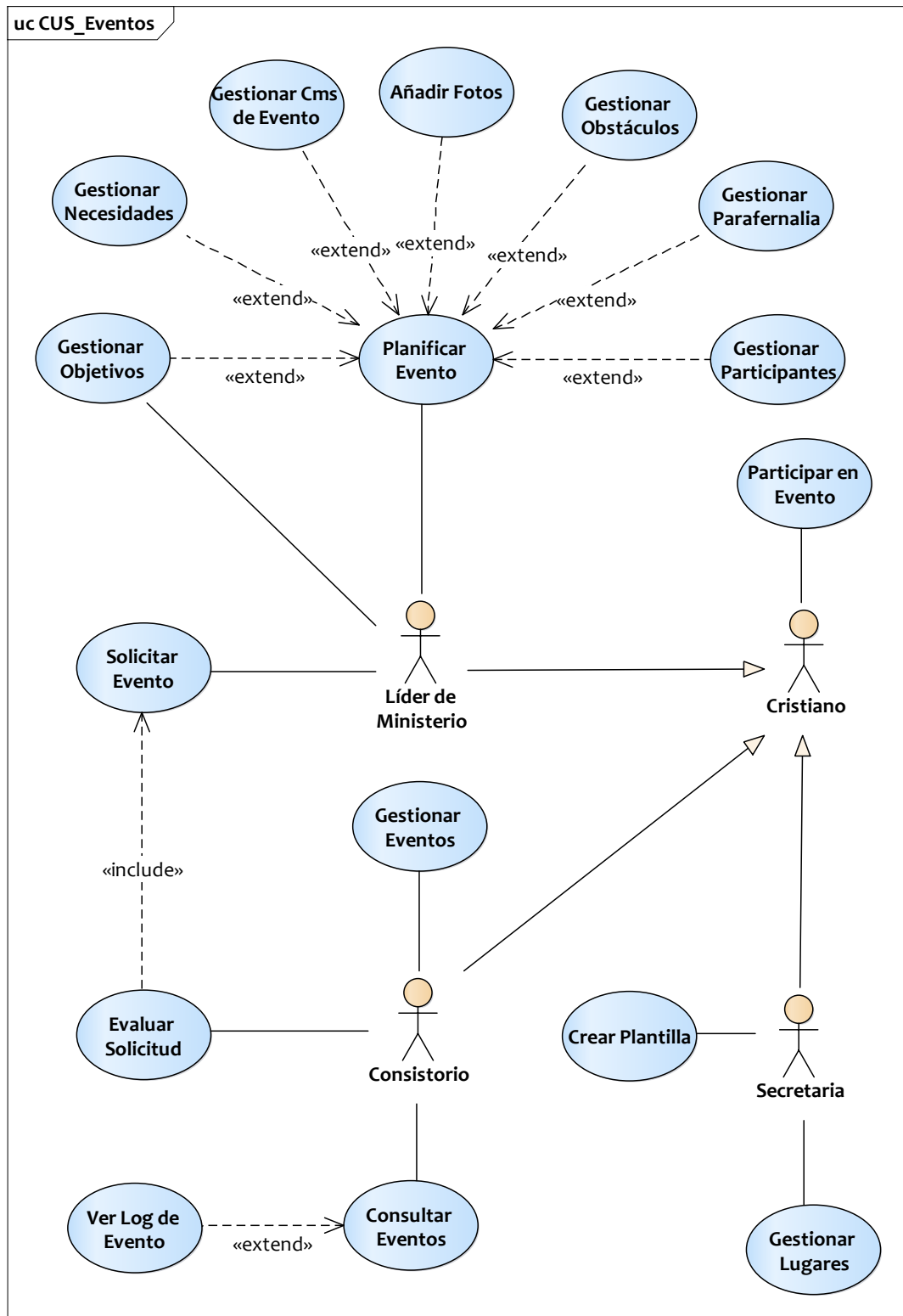


Figura 25 Diagrama de CUS del paquete Eventos

Fuente: Elaboración propia

- Paquete Seguridad

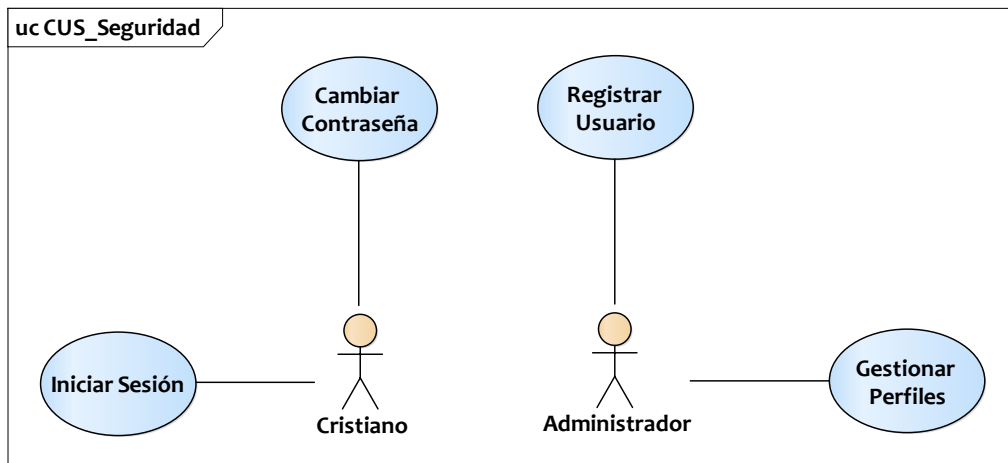


Figura 26 Diagrama de CUS del paquete Seguridad

Fuente: Elaboración propia

- Paquete Mantenimiento

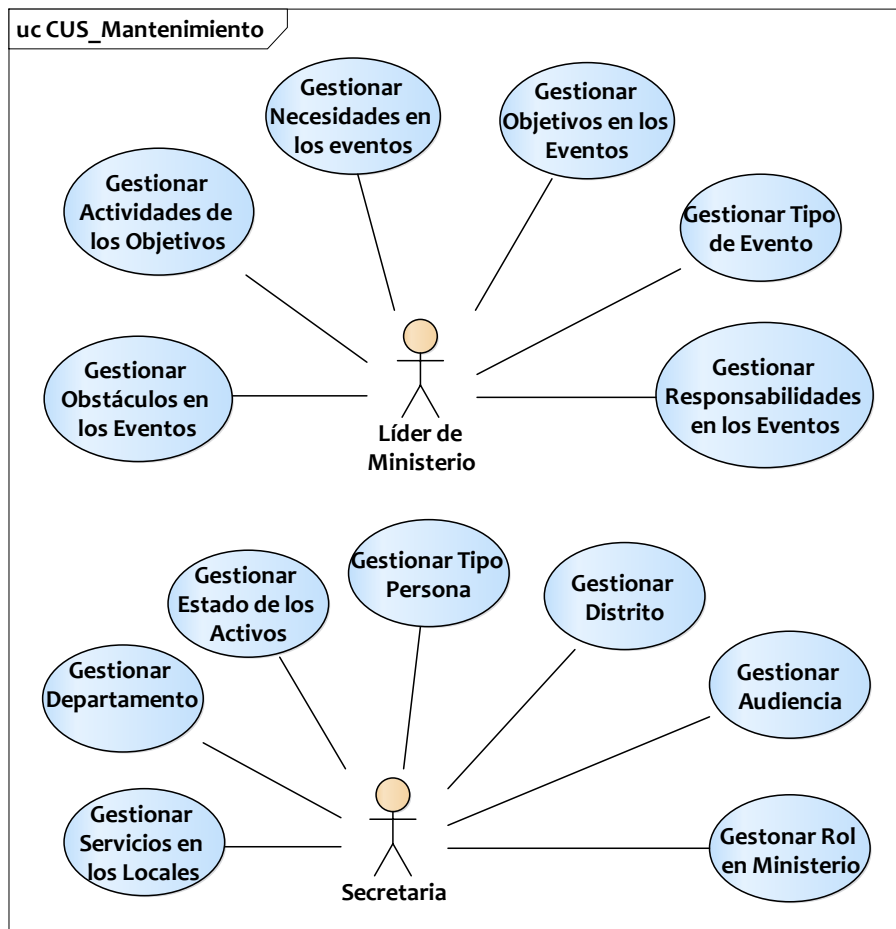


Figura 27 Diagrama de CUS del paquete Mantenimiento

Fuente: Elaboración propia

- Paquete Público

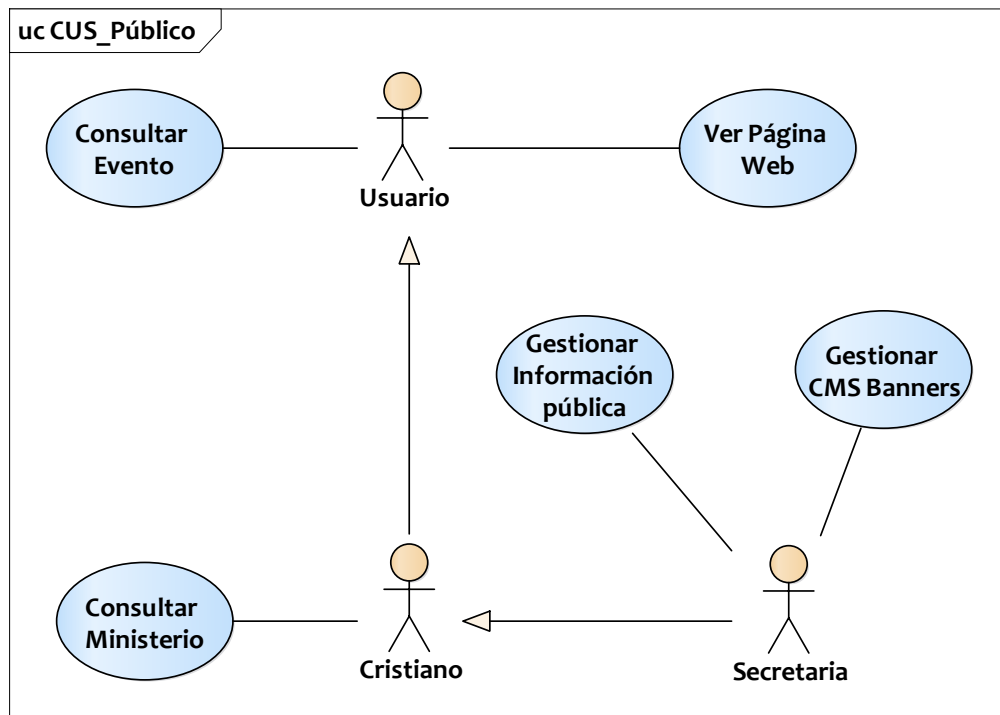


Figura 28 Diagrama de CUS del paquete Público

Fuente: Elaboración propia

- Paquete Congregación

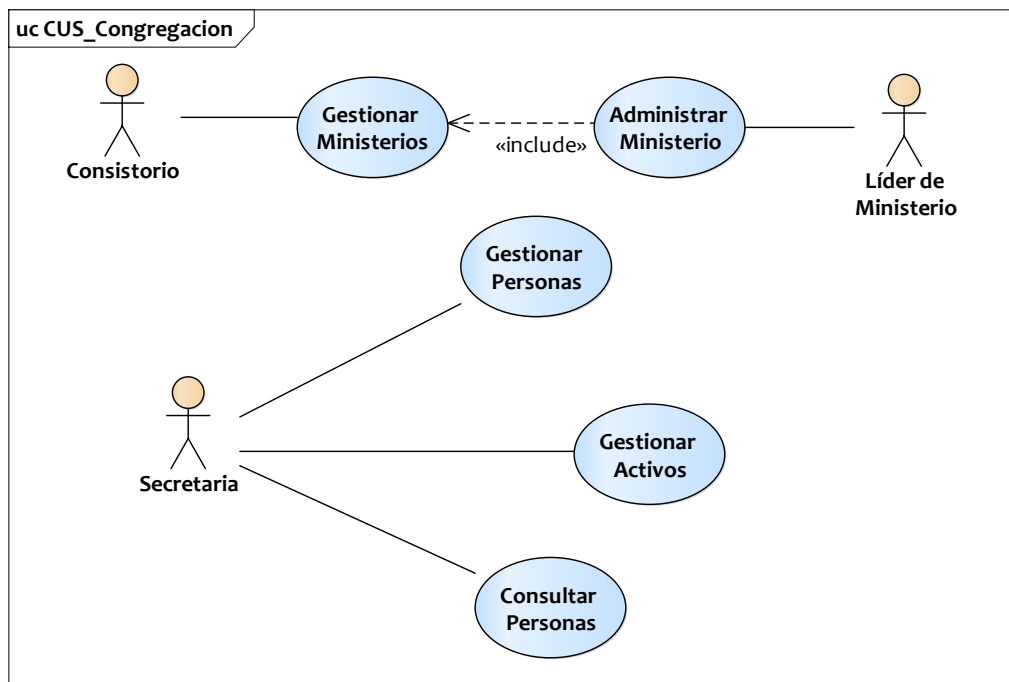


Figura 29 Diagrama de CUS del paquete Congregación

Fuente: Elaboración propia

- Paquete Reportes

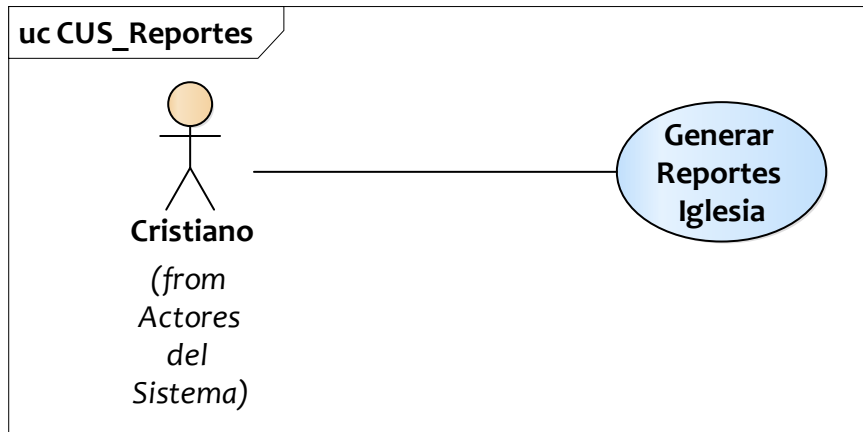


Figura 30 Diagrama de CUS del paquete Reportes

Fuente: Elaboración propia

5.2.6. Matriz CUN vs CUS

Tabla 3 Matriz CUN vs CUS

	Caso de Uso del Negocio		Caso de Uso del Sistema
1	CUN Seguimiento de miembros	1	CUS Gestionar Ministerios
		2	CUS Gestionar Personas
		3	CUS Consultar Personas
		4	CUS Generar Reportes Iglesia
		5	CUS Administrar Ministerio
		6	CUS Gestionar Tipo Persona
		7	CUS Gestionar Rol en Ministerio
		8	CUS Gestionar Activos
		9	CUS Gestionar Departamentos
		10	CUS Gestionar Distritos
2	CUN Gestión de Eventos	1	CUS Consultar Evento
		2	CUS Participar en Evento
		3	CUS Solicitar Evento
		4	CUS Evaluar Solicitud
		5	CUS Gestionar Participantes
		6	CUS Gestionar Lugares
		7	CUS Crear Plantilla
		8	CUS Gestionar Tipo de Evento
		10	CUS Gestionar Eventos
		11	CUS Gestionar Objetivos
		12	CUS Gestionar Necesidades
		13	CUS Gestionar Cms de Evento
		14	CUS Añadir Fotos
		15	CUS Gestionar Obstáculos
		16	CUS Gestionar Parafernalia
		17	CUS Planificar Evento
		18	CUS Consultar Eventos
		19	CUS Ver Log de Evento

5.4. Prototipos

5.4.1. Prototipo Solicitar Evento

El prototipo muestra un formulario modal 'Solicitar Evento' superpuesto sobre un calendario. El formulario contiene los siguientes campos:

- Nombre:** Campo de texto con el valor 'Campamento'.
- Descripción:** Campo de texto con el valor 'Campamento Por Fin de Año'.
- Tipo de Evento:** Menú desplegable con 'Campamento' seleccionado.
- Lugar:** Menú desplegable con 'Rumichaca' seleccionado.
- Fecha de Inicio:** Campo de texto con '01/12/2017'.
- Fecha de Fin:** Campo de texto con '04/12/2017'.
- Hora de Inicio:** Campo de texto con '08:00'.
- Hora de Fin:** Campo de texto con '16:00'.

En la parte inferior del formulario hay dos botones: 'Close' (gris) y 'Solicitar' (verde).

Figura 32 Prototipo del CUS Solicitar Evento

Fuente: Elaboración propia

5.4.2. Prototipo Gestionar Participantes

El prototipo muestra una interfaz para 'Gestionar Participantes'. Incluye un menú desplegable para seleccionar el evento ('Campamento').

PERSONAS:

Nombres	Apellidos	Telefono	Agregar
Julissa	Bobadilla	980617589	Añadir
Ricardo	Alama Ventocilla	980617263	Añadir
Julianne	Rosas Rojas	987654321	Añadir
Ivonne	Perez Estebia	978653421	Añadir

PARTICIPANTES:

Participante	Responsabilidad	Añadir
Julissa Bobadilla	Encargado de Juegos	Añadir
Ricardo Alama Ventocilla	Director de música	Añadir
Ivonne Perez Estebia	Ninguna	Añadir

RESPONSABILIDAD:

Responsabilidad: Encargado de Juegos (seleccionado), Encargado de Juegos, Director de música, Ninguna.

Hay un botón 'Añadir' asociado a esta sección.

Figura 33 Prototipo de CUS Gestionar Participantes

Fuente: Elaboración propia

5.4.3. Ver Calendario

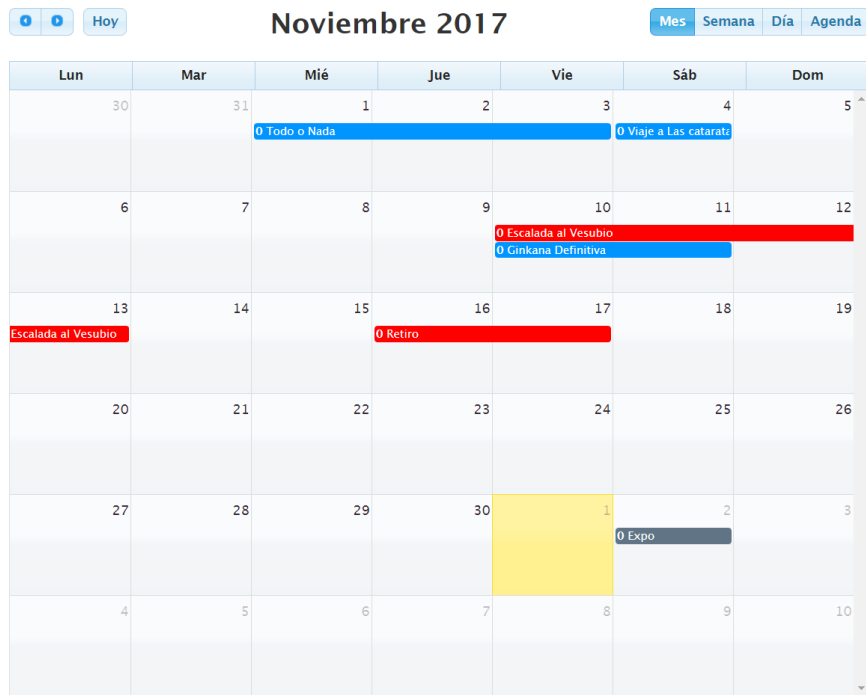


Figura 34 Prototipo del CUS Ver Calendario

Fuente: Elaboración propia

5.4.4. Reportes

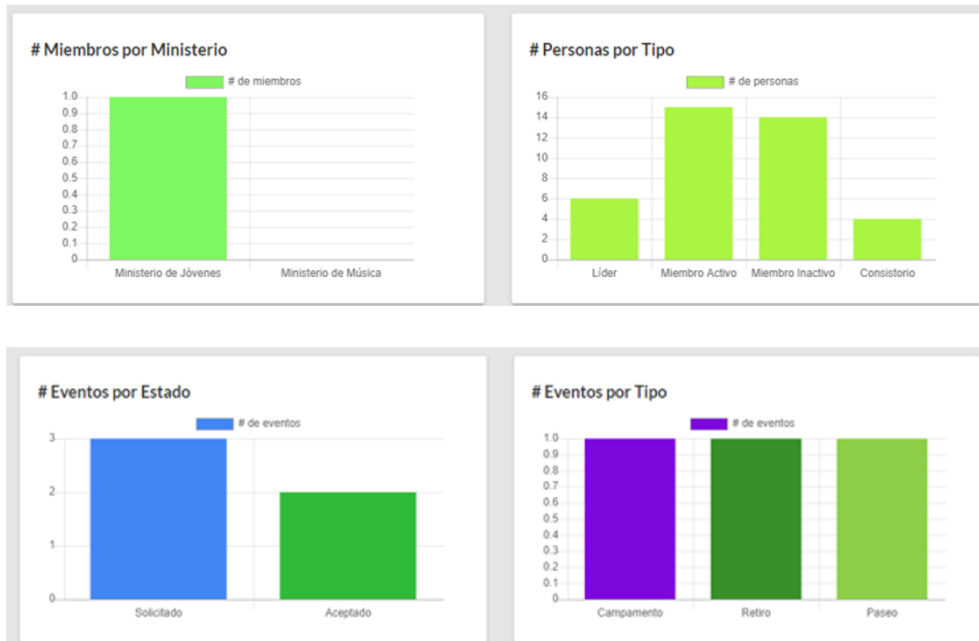


Figura 35 Prototipo de vista de reportes

Fuente: Elaboración propia

5.4.5. Gestionar Eventos

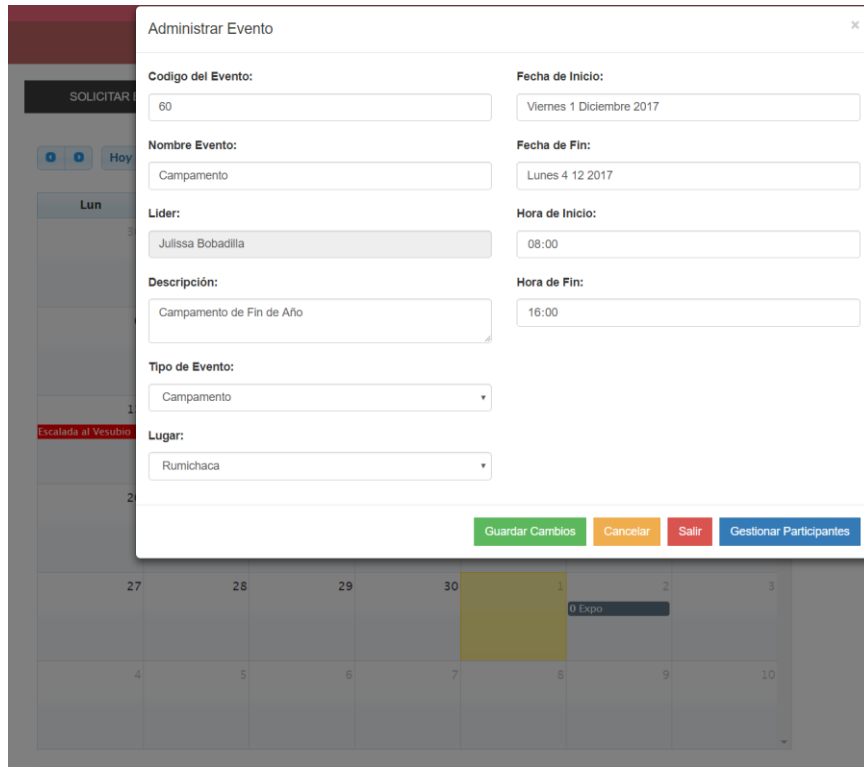


Figura 36 Prototipo del CUS Gestionar Eventos

Fuente: Elaboración propia

5.5. Matriz CUS vs FEATS

Source	CUS::Administrar Ministerio	CUS::Consultar Evento	CUS::Evaluar Solicitud	CUS::Gestionar Cms de Evento	CUS::Gestionar Eventos	CUS::Gestionar Ministerios	CUS::Gestionar Personas	CUS::Participar en Evento	CUS::Planificar Evento	CUS::Solicitar Evento	CUS::Ver Pagina Web	FEATS::UseCase1
FEATS::El sistema permitirá gestionar la información de los ministerios	↑					↑						
FEATS::La secretaria puede gestionar información de los miembros							↑					
FEATS::Los líderes podrán gestionar la información de los eventos				↑								
FEATS::Los líderes podrán planificar los eventos								↑				
FEATS::Los líderes serán capaces de solicitar eventos										↑		
FEATS::Los miembros de consistorio podrán ver el calendario con todos los eventos					↑							
FEATS::Los miembros del consistorio podrán evaluar las solicitudes de evento			↑									
FEATS::Los usuarios logeados puede participar en un evento							↑					
FEATS::Los usuarios podrán consultar la información pública de la iglesia											↑	
FEATS::Los usuarios podrán consultar los eventos en la página web	↑											

Figura 37 Matriz CUS vs Features

Fuente: Elaboración propia

CAPITULO VI: ARQUITECTURA

6.1. Casos de uso más significativos para la arquitectura

6.1.1. Propósito

La finalidad de mostrar los casos de uso más significativos recae en presentar un panorama global de cómo es el flujo principal de este proyecto. De esta manera no se detalla todos, sino que solo los altamente necesarios y se infiere que las funcionalidades detalladas en estos CUS ya están implementados o descritos en otros CUS complementarios.

6.1.2. Definiciones, Abreviaturas y Acrónimos

La definición completa de los conceptos y términos usados en la elaboración de documento se encuentra en el glosario de términos descrita en la parte final de tesis.

6.2. Representación de la arquitectura

6.2.1. Modelo Arquitectónico (4 + 1)

Las 4 vistas ubicadas en las esquinas se conectan por medio de la vista (+1) la cual tiene como función principal relacionar las 4 vistas.

Vista lógica

Se representa la función que el sistema proporcionara a los usuarios finales, es decir se representa lo que el sistema debe hacer y las funciones y servicios que ofrece.

Vista de despliegue

En esta vista se muestra el sistema desde la perspectiva de un programador y se ocupa de la gestión del software y se muestra como está dividido el sistema de software en componentes y la dependencia que hay entre esos componentes, para completar esta vista se puede incluir los Diagramas de paquetes y de componentes de UML.

Vista de procesos

En esta vista se muestra los procesos que hay en el sistema y la forma en la que se comunican estos procesos es decir se representan desde la perspectiva

de un integrador de sistema, el flujo de trabajo paso a paso de negocio, para completar esta vista se puede incluir el diagrama de actividad de UML.

Vista física

En esta vista se muestra de la perspectiva de un ingeniero de sistema como los componentes físicos del sistema, así como las conexiones físicas dentro de los componentes que conforman la solución incluyendo los servicios, para completar esta vista se puede incluir el diagrama de despliegue de UML.

Vista +1 o vista de escenario

Esta vista va a ser representada por los casos de uso del software y tiene la función de unir y relacionar las otras 4 vistas, esto quiere decir que desde un caso de uso podemos ver cómo se van relacionando las otras 4 vistas con lo que tendremos la trazabilidad de componentes, clases, equipos, paquetes, etc. para realizar cada caso de uso. Para completar esta vista se puede incluir los casos de uso de UML.



Figura 38 Diagrama del Modelo Arquitectónico 4 + 1

Fuente: ((Colombia), 2011)

6.3. Vista de Casos de Uso

6.3.1. Diagrama de Casos de Arquitectura

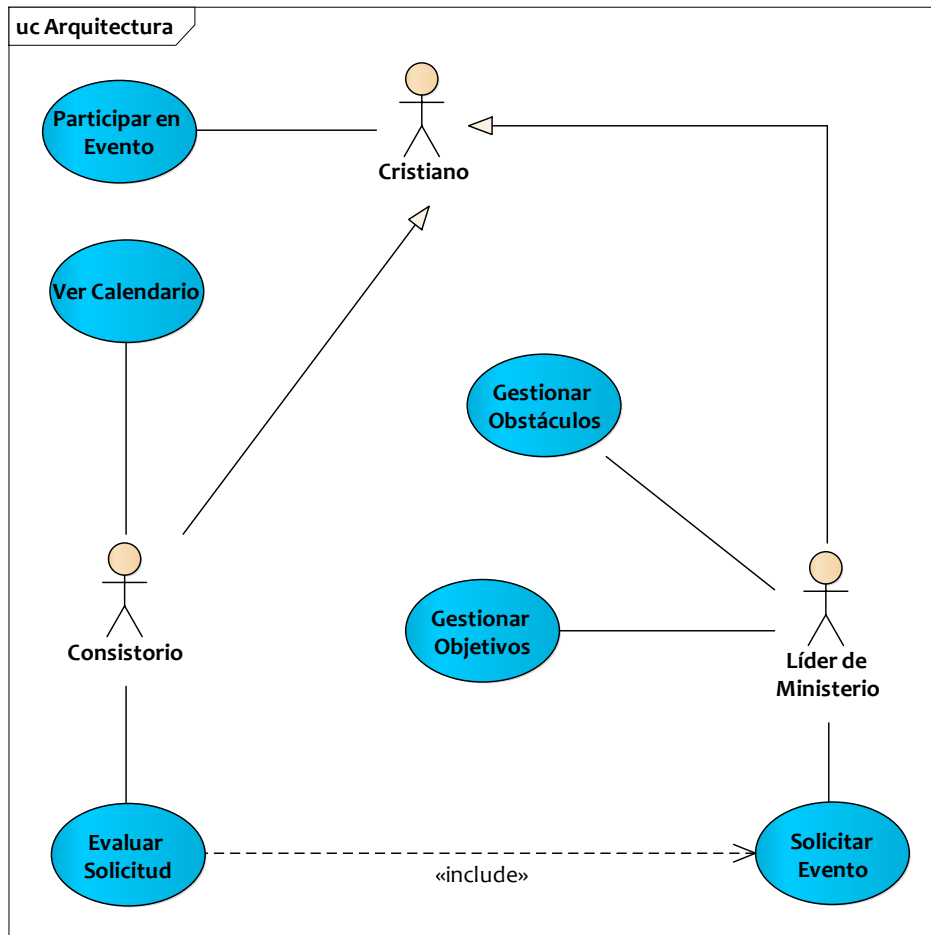


Figura 39 Diagrama de CUS de arquitectura

Fuente: Elaboración propia

6.3.2. Casos de uso más significativos

El caso de uso del negocio que identificamos para estos dos CUS más significativos fue el de “Organizar Evento”. EL CUN inicia con la petición al consistorio, por parte de un líder, para realizar un evento, siguiendo con la coordinación del lugar y los datos básicos. Se coordina con el lugar y luego se avisa a la congregación.

- **Gestionar Objetivos**

Nace del requerimiento “Planificar eventos”: este tiene un apartado de “Agregar objetivos”. El cumplimiento de objetivos es la base para iniciar una medición, generar reportes y empezar a tomar decisiones. Por esta razón fue que creamos un CUS específico para este requerimiento “Gestionar Objetivos”. Como pueden ver en la Figura 23 está dentro del paquete de eventos.

En la Tabla 4 se muestra la especificación del CUS. El líder se identificó como el actor principal y se han documentado los flujos correspondientes a cada una de las acciones que el usuario puede llevar a cabo.

Tabla 4 Especificación del CUS Gestionar Objetivos

Términos	Definición		
Caso de Uso	Gestionar Objetivos		
Pre-condición	El evento ha sido aceptado		
Post-condición	Se han agregado objetivos y actividades al evento		
Actor	Líder		
	Líder		Sistema
Flujo Principal	1	El CUS se inicia cuando, estando en “planificar” se activa la tarjeta de objetivos.	
			2 El sistema muestra la tarjeta con el siguiente contenido: <ul style="list-style-type: none"> • Una lista de los objetivos actuales y las actividades que le corresponden a cada uno. • Al margen derecho de cada actividad se encuentra el rango de fechas, un checkbox.

			<p>“alcanzado” y una “x” para eliminar dicha actividad.</p> <ul style="list-style-type: none"> • Junto al nombre de cada objetivo se puede el rango de fechas que abarca. • Al margen derecho de cada objetivo se muestra el porcentaje de magnitud respecto al propósito del evento, un checkbox. “alcanzado” en caso de no tener ninguna actividad y una “x” para eliminarlo junto con todas las actividades dentro del mismo. <p>También se muestran los siguientes botones:</p> <ul style="list-style-type: none"> • Agregar Objetivos • Agregar Actividades • Agregar Participantes • Nuevo Objetivo
3	El líder hace click en “Nuevo objetivo”.		
		4	Flujo principal del CUS Gestionar Responsabilidades en Evento.
5	El líder hacer click en “Agregar Objetivos”.		
		6	El sistema muestra un popup que contiene los siguientes

			<p>elementos:</p> <ul style="list-style-type: none"> • Un campo “Buscar”. • Una tabla con todos los objetivos que están registrados en el sistema que tiene las columnas: <ul style="list-style-type: none"> - Objetivo: nombre del objetivo. - Magnitud: el usuario ingresa la magnitud del objetivo (por defecto es 20). - Agregar: Botón para agregar el objetivo al evento.
	7	El líder llena el campo de magnitud.	
	8	El líder hace click en “Agregar”.	
			9 El sistema registra el objetivo en la BD.
			10 El sistema calcula el nuevo porcentaje de alcanzado que tiene el evento y lo actualiza.
			12 El sistema muestra la nueva lista de objetivos
			13 Se termina el CUS.
Flujo Alternativo Marcar Objetivo	3	El líder marca una casilla “Alcanzado”, al lado de un objetivo.	
			4 El sistema modifica el objetivo a “alcanzado” en la BD.

			5	El sistema calcula el nuevo porcentaje de alcanzado que tiene el evento y lo actualiza.
Flujo Alternativo Eliminar objetivo	3	El líder hace click en “x” al lado de un objetivo.		
			4	El sistema elimina el registro del objetivo de la BD junto con las actividades relacionadas.
			5	El sistema calcula el nuevo porcentaje de alcanzado que tiene el evento y lo actualiza.
			6	El sistema muestra la nueva lista de objetivos.
Flujo Alternativo Agregar Actividad	3	El Líder hace click en “Agregar Actividad”.		
			4	El sistema muestra un popup que contiene los siguientes elementos: <ul style="list-style-type: none"> • Un campo “Buscar”. • Una tabla con todos los objetivos que han sido agregados al evento, que tiene las columnas: <ul style="list-style-type: none"> - Objetivo: nombre del objetivo. - Actividad: Combobox que tiene las actividades que existen en el sistema.

				<ul style="list-style-type: none"> - Inicio: Fecha en la que se inicia la actividad (por defecto es el día de inicio del evento). - Fin: Fecha en la que termina la actividad (por defecto es el día de inicio del evento). - Agregar: Botón para agregar el objetivo al evento.
	5	El líder elige una actividad y llena los campos.		
	6	El líder hace click en “Agregar”.		
			7	El sistema registra la actividad en la BD.
			8	El sistema calcula el nuevo porcentaje de alcanzado que tiene el evento y lo actualiza.
			9	El sistema muestra la nueva lista de objetivos.
			10	El sistema muestra la nueva lista de obstáculos.
Flujo Alternativo Realizar Actividad	3	El líder marca una casilla “Alcanzado” al lado de una actividad.		
			4	El sistema modifica la actividad a “alcanzada” en la BD.
			5	El sistema calcula el nuevo porcentaje de alcanzado que

				tiene el evento y lo actualiza.
Flujo Alternativo Eliminar Actividad	3	El líder hace click en “x” al lado de una actividad.		
			4	El sistema elimina el registro de la actividad de la BD.
			5	El sistema calcula el nuevo porcentaje de alcanzado que tiene el evento y lo actualiza.
			6	El sistema muestra la nueva lista de objetivos.
			7	El sistema muestra la nueva lista de obstáculos.
Flujo Alternativo Actualizar Fechas	3	El líder hace click en la fecha de una actividad.		
			4	El sistema muestra un popup con dos campos: <ul style="list-style-type: none"> • Inicio • Fin Los cuales corresponden a la fecha de los mismos nombres.
	5	El líder llena los campos.		
	6	El líder hace click en Cerrar.		
			7	El sistema cierra el popup y actualiza los datos de dicha actividad en la BD
		8	El sistema muestra la nueva lista de objetivos	

El CUS en cuestión engloba algunas entidades que se han identificado y diagramado, así como podrán ver en la Figura 39. Todo apunta a la entidad Evento_Objetivo, que es la que guarda todos los objetivos que tiene el evento en cuestión. Esta nace la relación entre las entidades Evento y Objetivo. El Evento a su vez está relacionado con la entidad persona de cuya unión nace Participante, entidad que guarda los datos de cada participante del evento y sus responsabilidades (entidad Responsabilidad). Por otra parte, cada objetivo agregado al evento tiene una o muchas actividades asignadas, por esa razón nace la relación entre Actividad y Evento_Objetivo, la cual está relacionada con Participante. La utilidad de esta relación permite especificar el encargado de cada actividad.

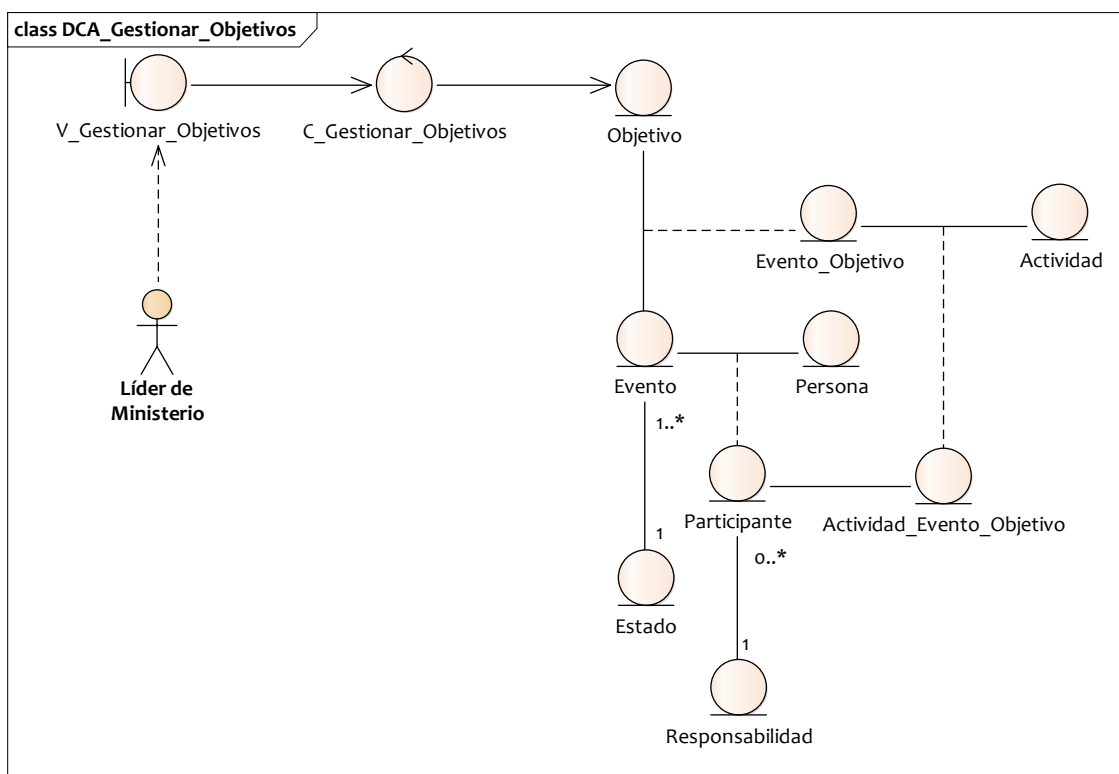


Figura 40 Diagrama de Clases Análisis del CUS Gestionar Objetivos

Fuente: Elaboración propia

Dado que el lenguaje de programación que usaré es php y el Framework que estoy usando para desarrollar el software es Laravel, el proyecto está construido con una arquitectura MVC, a continuación, muestro el resultado de la migración de un esquema de análisis a un producto ya hecho que viene a ser el software, representado en diagramas para este CUS. En las figuras 41 y

42 pueden verse los Diagramas de Clases de Diseño que son el resultado del Diagrama de Clases de Análisis anteriormente presentado.

En parte de la vista trabajamos con archivos constituidos básicamente por código html, css y javascript, por esa razón se ha representado la vista con una clase de tipo WebForm. El usuario es el que solicita la información a través del controlador, el cual realiza las llamadas a los Modelos que pueden verse en el diagrama. Estos 4 modelos van a realizar las llamadas a las tablas necesarias de la base de datos para que se pueda obtener la información correcta. Esta información es procesada devuelta y procesada por el controlador para que sea enviada a las distintas vistas para su organización y presentación en la pantalla.

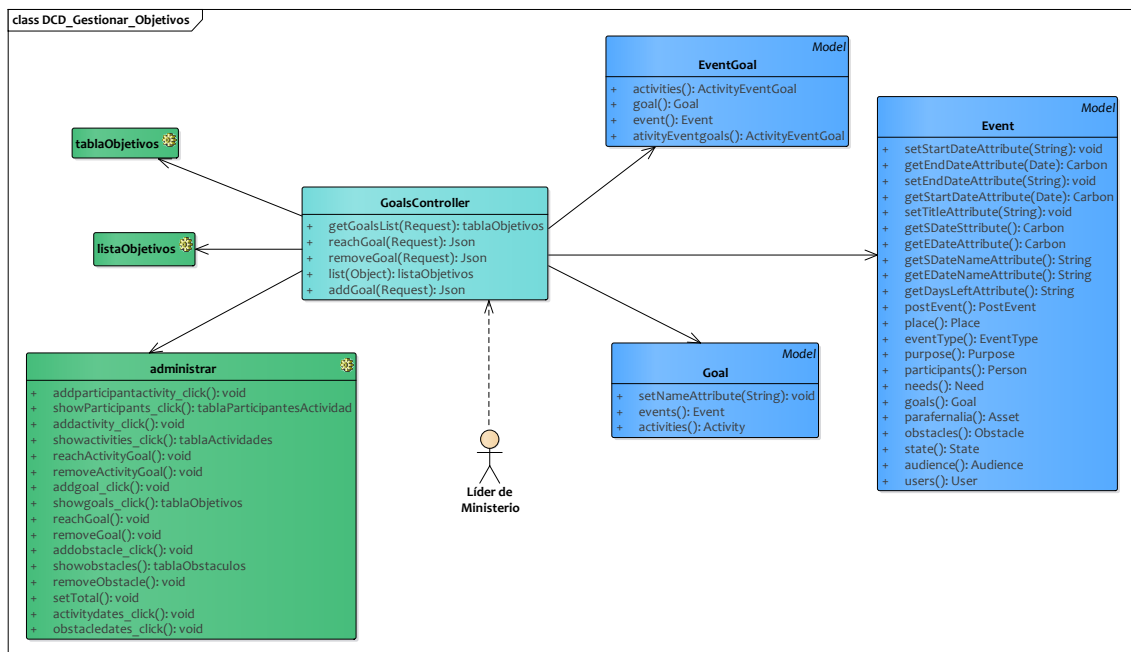


Figura 41 Diagrama de clases diseño del CUS Gestionar Objetivos

Fuente: Elaboración propia

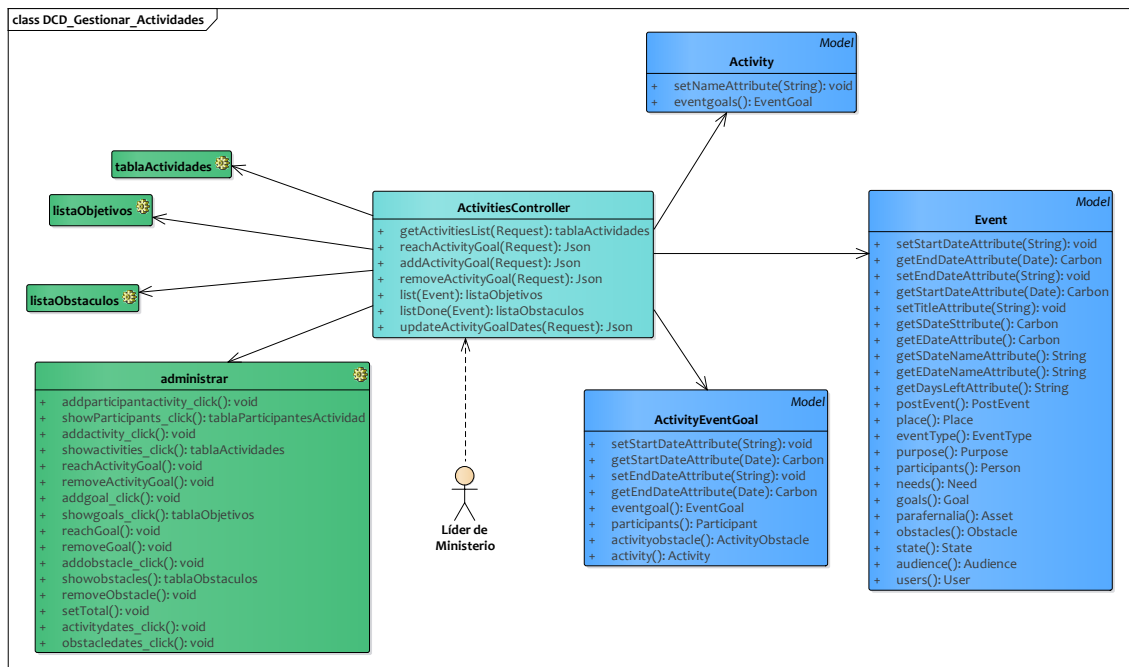


Figura 42 Diagrama de clases diseño del CUS Gestionar Actividades

Fuente: Elaboración propia

La estructura del código se ha dividido en clases como se puede observar en las figuras anteriores y cada una de esas clases se comunican entre sí para poder lograr el funcionamiento del software. A continuación, se muestran los Diagramas de Secuencia de Diseño en los que se puede observar los distintos métodos que hacen funcionar el CUS.

Figura 41: El usuario hace click en el botón "Agregar Objetivos". Esto dispara un evento que el controlador reconoce y ejecuta el método `getGoalsList` donde captura los parámetros que fueron enviados dentro del Request de la vista. Dentro de este método se obtiene el modelo del respectivo evento cuyo id ha sido reconocido, se obtienen todos los objetivos que no estén asignados a un ese evento, ordenados por id en forma ascendente. Se construye la vista `tablaObjetivos` que contiene la lista de objetivos devuelta por la consulta y finalmente se presenta en la pantalla. El usuario ingresa la magnitud que desea que tenga un objetivo en específico y le da click en agregar. EL controlador ejecuta la función `addGoal` con el Request capturado de la vista. El método obtiene el modelo del evento

respectivo y con la ayuda del modelo Objetivo, guarda en la base datos el objetivo elegido. Luego se ejecuta el método list con el evento actual como parámetro, que va a listar todos los objetivos del evento, para que posteriormente se construya la vista listaObjetivos para que sea enviada a la web.

Para poder marcar un objetivo como realizado, el usuario debe seleccionar la casilla en la lista de objetivos y el controlador ejecuta el método reachGoal. Lo que hace este método es obtener los datos del evento actual y para luego obtener el registro del objetivo marcado. Existe una variable con la magnitud de dicho objetivo. Si la casilla se ha activado el campo reached del registro se actualiza a 1, en caso contrario se actualiza con 0 y la magnitud pasa a convertirse en negativa. Finalmente se actualiza el valor en el modelo PostEvent que es el que accede al campo que guarda el porcentaje alcanzado del evento y suma el valor calculado de la magnitud.

Cuando el usuario desea eliminar un objetivo debe hacer click en la "x" que aparece en la lista de objetivos. Esto desencadena el método removeGoal del controlador. El método obtiene el modelo del evento actual para luego quitar el objetivo de ese evento. Se vuelve a hacer un cálculo de la suma de los porcentajes con todos los objetivos "alcanzados" y se actualiza la información en la base de datos. Luego se devuelve esta información a la vista.

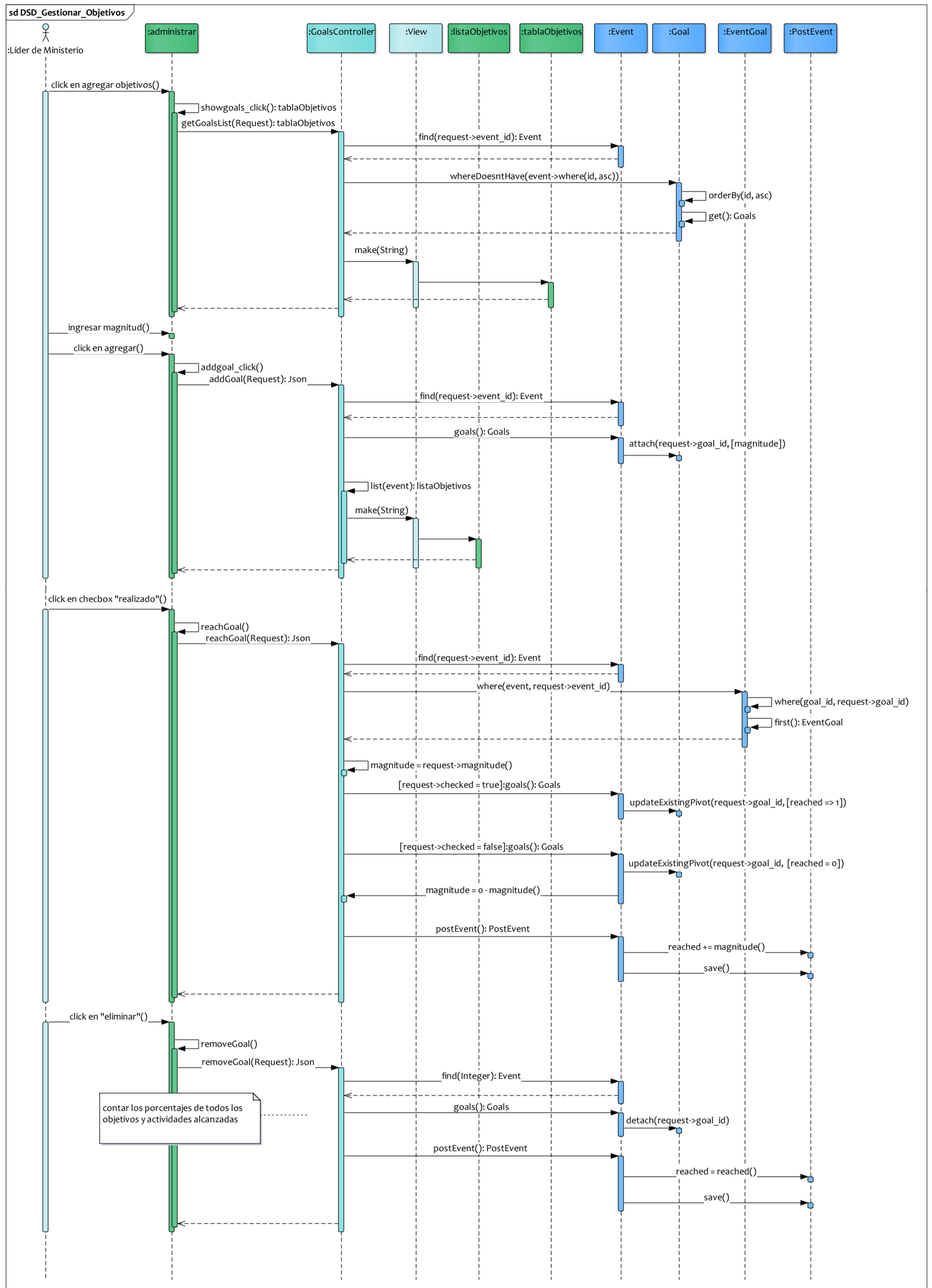


Figura 43 Diagrama de secuencia diseño del CUS Gestionar Objetivos

Fuente: Elaboración propia

Para la parte de Gestionar Actividades tenemos lo siguiente: el usuario hace click en el botón de agregar actividades, esto ejecuta una llamada a un método del controlador llamado `getActivitiesList`. Este método va a hacer una llamada al modelo `EventGoal` que realiza una consulta en la base de datos que retorna una lista de la tabla `event_goal` con todos los objetivos que pertenecen al evento actual. Hace otra llamada al Modelo `Activity` para traer una lista de las actividades que no están ligadas a ningún objetivo. Con todos esos datos, el controlador construye la vista `tablaActividades` y la devuelve a la vista. Con esta tabla en la pantalla, el usuario llena los campos correspondientes y hace click en el botón de agregar. Se llama al método `addActivityGoal` el cual consigue el modelo `EventGoal` de acuerdo con el objetivo al cual se desea agregar la actividad, también se obtiene el modelo del evento actual y se inserta un registro en la base de datos con el modelo `ActivityEventGoal`. Se obtiene el porcentaje alcanzado del evento y se realiza un cálculo que se puede observar en la Figura 44, con el resultado del cálculo se actualiza el porcentaje alcanzado el evento. Luego de la actualización de los datos se realiza una consulta de los objetivos pertenecientes al evento actual y con esos datos se construye la vista `listaObjetivos`. Seguidamente se realiza el mismo proceso, pero esta vez se construye la vista `listaObstaculos`. Ambas se retornan a la vista.

Para marcar una actividad como realizada el usuario debe hacer click en la casilla al lado de la actividad elegida. El controlador ejecuta el método `reachActivityGoal` dentro del cual va a obtener el modelo del objetivo al que pertenece la actividad en cuestión y también obtiene el modelo del evento actual. Si la casilla ha sido seleccionada, el campo "reached" del registro que contiene la actividad se actualiza a 1 y se actualiza el campo "reached" del objetivo correspondiente de acuerdo a: si la cantidad de actividades logradas es igual a la cantidad total de actividades. Si la casilla ha sido deseleccionada, el campo "reached" del registro que contiene la actividad se actualiza a 0 y se actualiza el campo "reached" del objetivo correspondiente a 0. Se guarda los datos del objetivo y se actualiza el porcentaje de logro del evento actual.

Para eliminar una actividad de un objetivo, se hace click en la "x" al lado de la actividad y se hace un llamado a `removeActivityGoal`, el cual obtiene el

modelo del Objetivo al que pertenece la actividad elegida, también obtiene el modelo del evento actual. Con esos datos elimina el registro de la actividad de la base de datos. Si la nueva cantidad de actividades es mayor que 0 entonces se actualiza el campo "reached" del objetivo correspondiente, dependiendo si la cantidad de actividades alcanzadas es igual a la cantidad total de actividades. Luego se realiza el cálculo que se muestra en la Figura 44 y se actualiza el porcentaje de logro del evento actual. Luego de eso, se construye la vista listaObjetivos, la lista Obstáculos y se retorna a la vista.

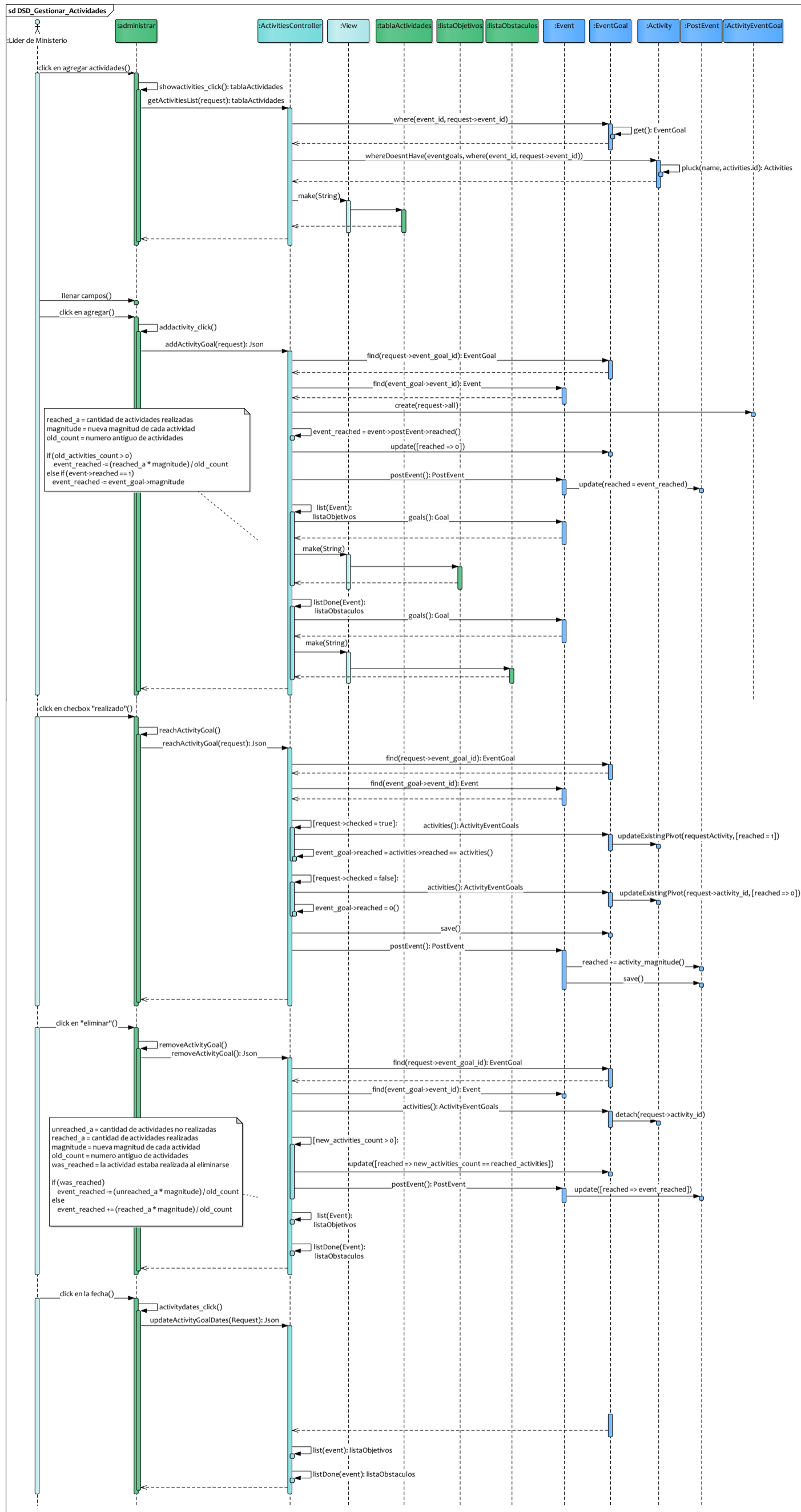


Figura 44 Diagrama de secuencia diseño del CUS Gestionar Actividades

Fuente: Elaboración propia

- **Gestionar Obstáculos**

Nace del requerimiento “Planificar Eventos”. Este tiene un apartado de “Asignarles obstáculos a las actividades”. La asignación de obstáculos que deliberadamente se han presentado durante la ejecución de un evento es de gran ayuda para poder buscar soluciones para evitar o mitigar dichos inconvenientes y como consecuencia mejorar la calidad de los eventos en la iglesia. Por esa razón la creación de este CUS dentro del paquete de eventos (Figura 23) fue considerada importante.

En la Tabla 5 se muestra la especificación del CUS. El líder se identificó como el actor principal del CUS y de han documentados los flujos correspondientes a cada una de las acciones que el usuario puede llevar a cabo.

Tabla 5 Especificación del CUS Gestionar Obstáculos

Términos	Definición		
Caso de Uso	Gestionar Obstáculos		
Pre-condición	El evento ha sido aceptado.		
Post-condición	Se han agregado obstáculos a las actividades de un evento.		
Actor	Líder		
	Líder		Sistema
Flujo Principal	1	El CUS se inicia cuando, estando en “planificar” se activa la tarjeta de obstáculos.	
			2 El sistema muestra la tarjeta con el siguiente contenido: <ul style="list-style-type: none"> • Una lista de los objetivos actuales y las actividades que le corresponden a cada uno con letras verdes si no hay

			<p>obstáculo alguno, azules si existe un obstáculo sin ningún retraso, amarillas si hay retraso en las fechas de la actividad y rojo si dicha actividad no ha podido realizarse.</p> <ul style="list-style-type: none"> Al margen derecho de cada actividad se encuentra el nombre del obstáculo que ocurrió en dicha actividad, el rango de fechas del obstáculo y una checkbox “Total”, que indica si el obstáculo ha impedido la realización de la actividad.. <p>También se muestran el siguiente botón:</p> <ul style="list-style-type: none"> Agregar Obstáculos
3	El líder hace click en “Agregar Obstáculos”		
		4	<p>El sistema muestra un popup que contiene los siguientes elementos:</p> <ul style="list-style-type: none"> Un campo “Buscar”. Una tabla con las actividades del evento que no tienen registrado un obstáculo. Tiene las siguientes columnas: <p>- Actividad: nombre de la</p>

				<p>actividad.</p> <ul style="list-style-type: none"> - Obstáculo Combobox que tiene los obstáculos registrados en el sistema. - Inicio: Fecha en la que se inicia el obstáculo (por defecto es el día de inicio de la actividad). - Fin: Fecha en la que termina la actividad (por defecto es el día de fin de la actividad). - Agregar: Botón para asignar el obstáculo a la actividad.
	5	El líder llena los campos.		
	6	El líder hace click en “Agregar”.		
			7	El sistema registra el obstáculo en la BD.
			8	El sistema muestra la nueva lista de obstáculos.
			9	Se termina el CUS.
Flujo Alternativo Marcar “Total”	3	El líder hace click en checkbox “Total” al lado de una actividad.		
			4.1	Si el checkbox no está seleccionado el sistema cambia el obstáculo a total = 1 y actualiza la BD.
			5.1	El sistema cambia el color de toda la actividad a rojo.

			6.1	Si la actividad a la que pertenece el obstáculo está registrada como “alcanzada”, se activa el flujo alternativo “realizar actividad” del CUS Gestionar Objetivos.
			4.2	Si es checkbox está seleccionado, el sistema cambia el obstáculo a total = 0 y actualiza la BD.
			5.2	El sistema cambia el color de toda la actividad a su respectivo color.
Flujo Alternativo Actualizar Fechas	3	El líder hace click en la fecha de un obstáculo.		
			4	El sistema muestra un popup con dos campos: <ul style="list-style-type: none"> • Inicio • Fin Los cuales corresponden a las fecha de los mismos nombres.
	5	El líder llena los campos.		
	6	El líder hace click en Cerrar.		
			7	El sistema cierra el popup y actualiza los datos de dicha actividad en la BD.
			8	El sistema muestra la nueva lista de obstáculos.

Las entidades que engloba el presente CUS se han identificado, diagramado y se muestran en la Figura 45. La entidad central del CUS es aquella llamada Actividad_Obstáculo. Esta es la que guarda los datos de los obstáculos que se han presentado por cada actividad. Esta entidad está ligada a la entidad “Obstáculo” que es de donde obtiene la información básica del obstáculo. La entidad principal también tiene una relación con otra entidad importante llamada Actividad_Evento_Objetivo; dicha relación indica que la actividad de un evento puede tener asociado un obstáculo o ninguno. Esta última entidad nace de una relación entre las entidades de Actividad y Evento_Objetivo, donde, como se especifica en el CUS anterior, cada objetivo de un evento se relaciona con una o más actividades.

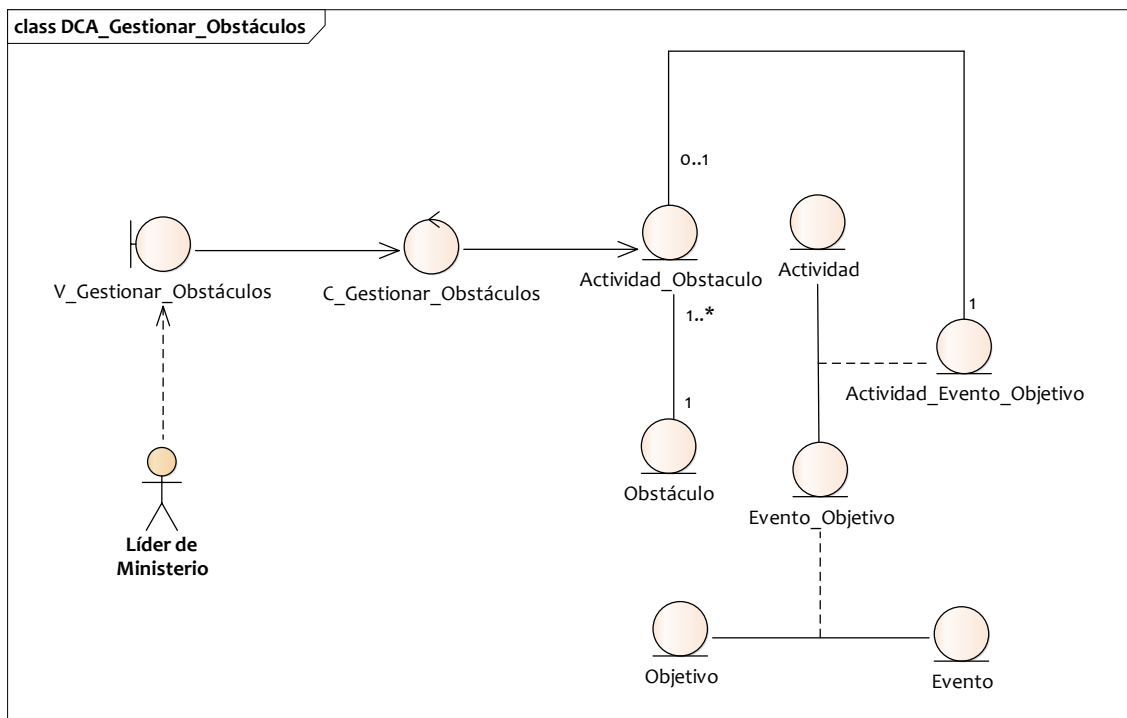


Figura 45 Diagrama de Clases Análisis del CUS Gestionar Obstáculos

Fuente: Elaboración propia

Como se mencionó en el apartado anterior, el lenguaje de programación y Framework usados, son PHP y Laravel respectivamente. Partiendo de ese punto y dado que la arquitectura de dicho Framework sigue un MVC, muestro la migración del diagrama de análisis, presentado anteriormente, al diagrama de diseño que representa el funcionamiento del software, en la Figura 46.

Para las vistas trabajamos con archivos compuestos básicamente por código html, css y javascript, pro está razón también se utilizaron clases de tipo WebForm. El usuario activa los métodos dentro del controlador que llama las vistas necesarias para realizar cada actividad en específico. Por otro lado, el controlador también se encarga de obtener información los 4 modelos que pueden verse en el diagrama. La información es procesada y devuelta a las vistas para su organización y presentación en la pantalla.

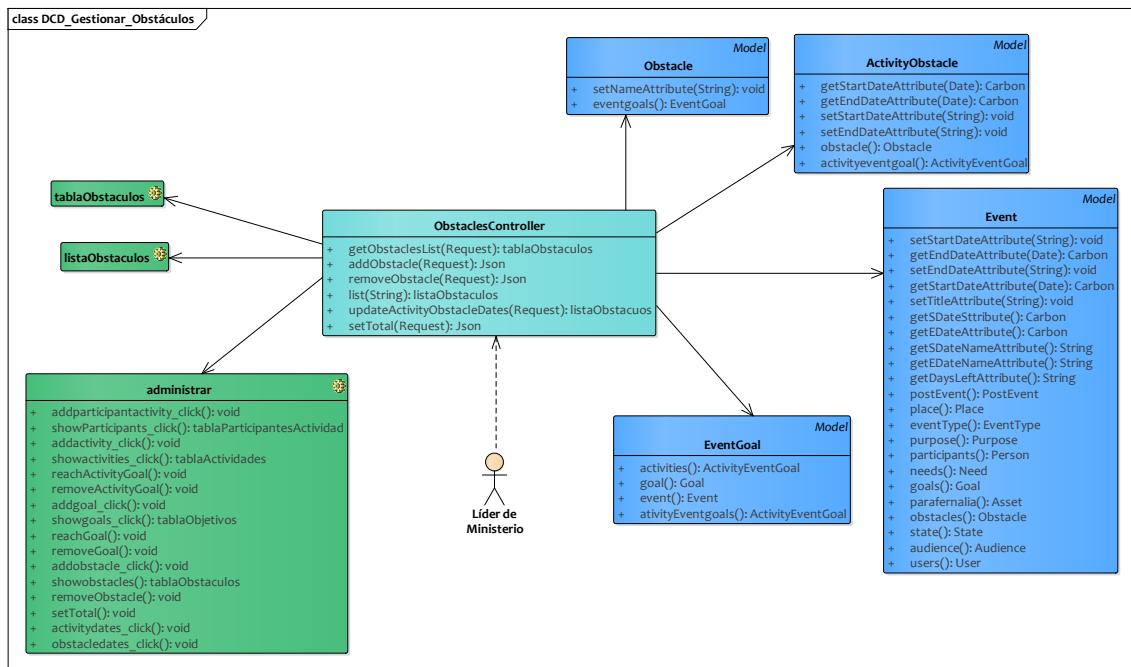


Figura 46 Diagrama de clases diseño del CUS Gestionar Obstáculos

Fuente: Elaboración propia

Como se ha mencionado, las clases están divididas en 3 niveles que se comunican entre sí para lograr que el software funcione correctamente. A continuación, en la Figura 47 se muestra el diagrama de Secuencia de Diseño en que se puede observar los métodos que hacen que el CUS funcione.

El usuario hace click en el botón “Agregar Obstáculos” de la vista. Esto dispara un evento que el controlador reconoce y ejecuta el método `getObstaclesList`, el cual captura los parámetros enviados desde el Request (proporcionado por la vista). El método comienza por realizar una consulta a la Base de Datos, por medio del modelo `Obstáculo`, de todos los obstáculos registrados en el sistema ordenados de forma ascendente por nombre. Posteriormente se realiza una

consulta para obtener las actividades relacionadas al evento que no tengan asociadas ningún obstáculo. La información es devuelta a los modelos respectivos y estos últimos mandan dichos al controlador, para que a continuación se construya la vista listaObstaculos y sea enviada a la web.

El usuario llena los campos correspondientes y hace click en el botón agregar de lista de Obstáculos. El controlador detecta la llamada a la función addObstacle y realiza una inserción en la Base de datos con la ayuda del modelo ActivityObstacle, con los datos enviados por medio del Request. Luego de realizada dicha operación se llama al método list, el cual, a su vez, por medio del método goals del Modelo Event, consigue la lista actualizada de los objetivos del evento (con sus respectivas actividades y obstáculos) y con todos esos datos construye la lista actualizada de Obstáculos del evento.

Para poder eliminar un obstáculo el usuario debe hacer click en el botón Eliminar. El controlador detecta la llamada y con el identificador de dicho obstáculo (devuelto por el Request) ejecuta el método destroy del modelo ActivityObstacle. Luego de eso Se hace nuevamente la llamada al método list para conseguir y mostrar la lista actualizada de obstáculos del evento.

Si usuario desea especificar que un obstáculo ha podido arruinar o saltar una actividad por completo, debe seleccionar la casilla “Total”. Al momento de darle click el controlador detecta la llamada y ejecuta el método setTotal. El presente método va a obtener los datos respectivos del Obstacle del evento a través del Modelo ActivityObstacle y va a verificar si es que ha seleccionado o deseleccionado la casilla de verificación. En el primer caso, se cambia el campo total a 1, mientras que el segundo caso se cambia el valor a 0. Se guardan los cambios y se devuelve al resultado a la vista. La vista detecta el cambio y verifica el estado de la actividad relacionada a dicho obstáculo. En caso de tener un estado realizado o “reached”, se realiza un click automático en la casilla “Realizado” de la lista de actividades y se ejecuta el método descrito en el punto anterior; en caso contrario, no se realiza ninguna acción extra.

Si se desea actualizar la fecha de los obstáculos, el usuario debe hacer click en la fecha y eso dispara un evento que es reconocido por el controlador para ejecutar el método updateActivityObstacleDates que recibe los datos a través

de un Request. El método encuentra el modelo del obstáculo determinado y actualiza la información con el método update del Modelo ActivityObstacle. Luego de eso, realiza los pasos para poder obtener la lista de Obstáculos Actualizada y mostrarla en la vista.

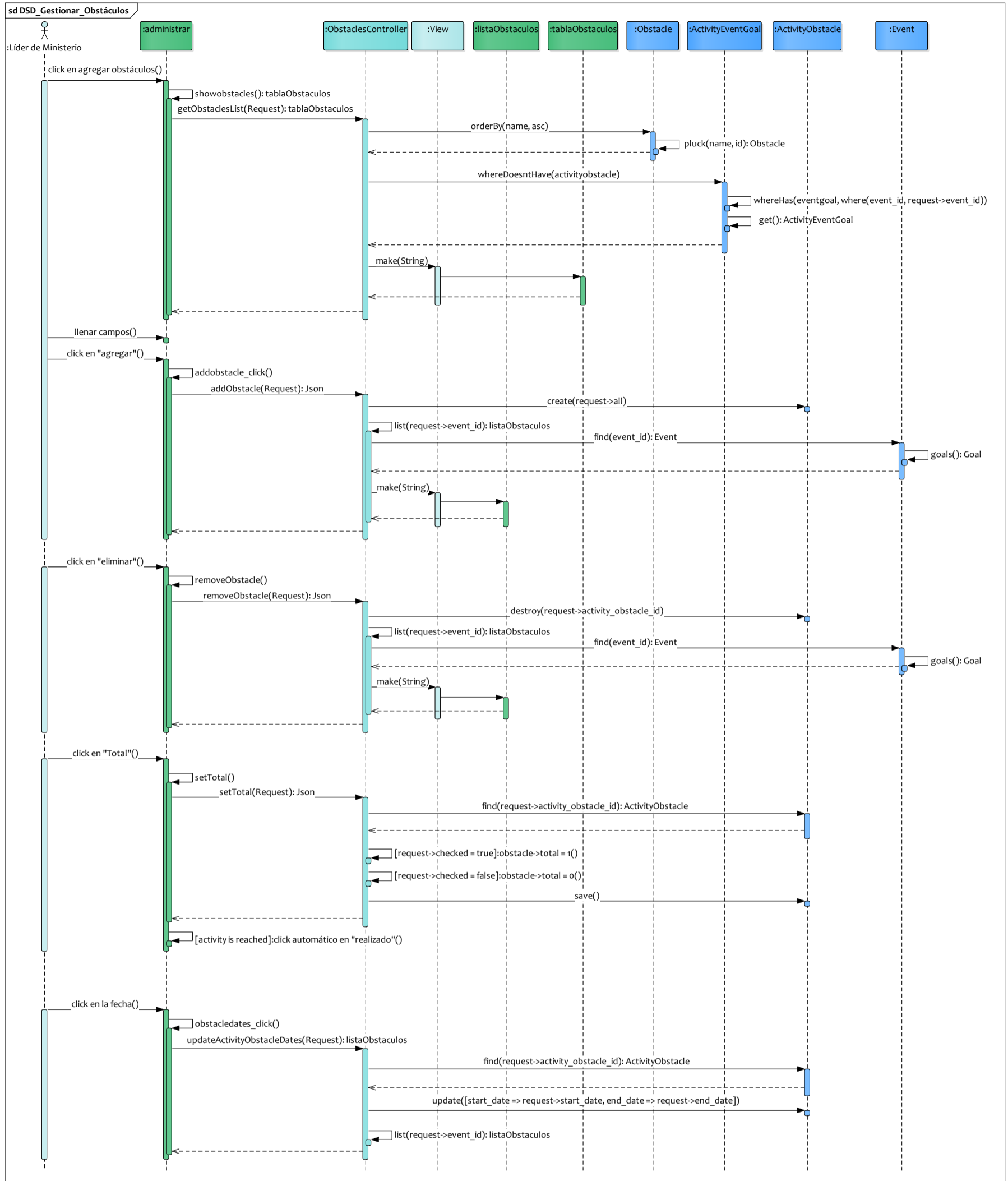


Figura 47/ Diagrama de secuencia diseño del CUS Gestionar Obstáculos

Fuente: Elaboración propia

6.3.3. Especificaciones de Casos de Uso

- **Solicitar Evento**

Tabla 6 Especificación del CUS Solicitar Evento

Términos	Definición		
Caso de Uso	Solicitar Evento		
Pre-condición	El usuario debe haber iniciado sesión en el sistema.		
Post-condición	La solicitud debe aparecer en la lista de pendientes de la página del consistorio.		
Actor	Líder		
	Líder		Sistema
Flujo Principal	1	El CUS se inicia cuando el líder selecciona “Solicitar” en el menú de opciones.	
			2 El sistema muestra una pantalla donde se puede ver lo siguiente: <ul style="list-style-type: none"> • Un calendario con los eventos que se han solicitado, aprobado y rechazado. • Una lista con las plantillas de los tipos de eventos que se han registrado en el sistema. • Una lista de los eventos aprobados. • Una lista de los eventos en curso. • Una lista de los eventos finalizados.
	3	El líder hace click y	

	arrastra la plantilla del eventos q quiere solicitar hasta la fecha que desea y lo suelta.		
		4	El sistema muestra dicha plantilla en la fecha del calendario elegida.
5	El líder hace click en la plantilla del calendario.		
		6	El sistema abre un modal en el que se muestra un formulario que contiene los siguientes campos: <ul style="list-style-type: none"> • Nombre del Evento • Propósito • Descripción • Imagen • Dirigido A • Lugar • Día de Inicio • Día de Fin • Hora de Inicio • Hora de Fin • Comentario • A través
7	El líder llena los campos obligatorios.		
8	El líder hace click en el botón guardar.		
		9	El sistema registra los datos del evento en la BD con el estado "Solicitado".
		10	El sistema registra en la BD

				la operación que se ha realizado.
			11	El sistema cierra el modal, borra la plantilla del calendario y coloca los eventos en color celeste.
			12	Se termina el CUS
Flujo Alternativo: Guardar “A Través”	7	El líder llena los campos obligatorios y hace click en la opción “A través”.		
	8	El líder hace click en el botón guardar.		
			9	El sistema registra los datos del evento en la BD con el estado “Aceptado”.
			10	El sistema registra la operación en la BD, con el mensaje que ha ingresado el usuario.
			11	El sistema crea una tarea para que evento cambie a estado “En Curso” en el momento correcto.
			12	El sistema crea una tarea para que evento cambie a estado “Finalizado” en el momento correcto.
			13	El sistema cierra el modal y borra la plantilla que se veía en el calendario y coloca el evento en color verde.
Flujo	7	El líder hace click en		

Alternativo:		borrar.		
Borrar				
			8	El sistema cierra el modal y borra la plantilla que se veía en el calendario.

- **Evaluar Solicitud**

Tabla 7 Especificación del CUS Evaluar Solicitud

Términos	Definición			
Caso de Uso	Evaluar Solicitud			
Pre-condición	El líder ha solicitado un evento.			
Post-condición	El evento debe ser aceptado o rechazado.			
Actor	Consistorio			
		Miembro del consistorio		Sistema
Flujo Principal	1	El CUS se inicia cuando el consistorio elige “Solicitar” en el menú de opciones.		
			2	El sistema muestra una pantalla donde se puede ver lo siguiente: <ul style="list-style-type: none"> • Un calendario con los eventos que se han solicitado, aprobado y rechazado. • Una lista de eventos que falta evaluar y la persona que lo ha solicitado.
	3	El consistorio hace click en “Info”.		
			4	El sistema muestra un popup donde se podrá ver la

			información básica del evento, así como el comentario de la persona que lo ha solicitado.
5	El consistorio hace click en cerrar.		
6	El consistorio hace click en el evento que desea evaluar.		
		7	El sistema abre un modal que contiene: <ul style="list-style-type: none"> • Campo para ingresar la razón de la acción que tomará, así como cualquier comentario que desee. • Botón “Aceptar” • Botón “Rechazar”
8	El consistorio ingresa su comentario.		
9	El consistorio hace click en el botón “Aceptar”.		
		10	El sistema cambia el estado del eventos a “Aceptado”.
		11	El sistema registra la operación en la BD con el mensaje que ingresado.
		12	El sistema crea una tarea para que evento cambie a estado “En Curso” en el momento correcto.
		13	El sistema crea una tarea para que evento cambie a

				estado “Finalizado” en el momento correcto..
			14	El sistema cierra el modal, borra el evento de la lista de pendientes y cambia el color del evento en el calendario a verde.
			13	Se termina el CUS
Flujo Alternativo: Rechazar	9	El consistorio hace click en “Rechazar”.		
			10	El sistema cambia el estado del eventos a “Rechazado”.
			11	El sistema registra la operación en la BD con el mensaje que ha ingresado.
			12	El sistema cierra el modal, borra el evento de la lista de pendientes y cambia el color del evento en el calendario a rojo.
Flujo Alternativo: Click en “Aceptar”	6	El consistorio hace click en el botón “aceptar” de la lista de pendientes.		
			7	El sistema cambia el estado del evento a “Aceptado”.
			8	El sistema registra la operación en la BD con el mensaje “Aceptado automáticamente”.
			9	El sistema crea una tarea para que evento cambie a estado “En Curso” en el momento correcto.

			10	El sistema crea una tarea para que evento cambie a estado “Finalizado” en el momento correcto.
			11	El sistema borra el evento de la lista y cambia el color del evento en el calendario a verde.
Flujo Alternativo: Click en “Rechazar”	6	El consistorio hace click en el botón “rechazar” de la lista de pendientes.		
			7	El sistema cambia el estado del evento a “Rechazado”.
			8	El sistema registra la operación en la BD con el mensaje “Rechazado automáticamente”.
			9	El sistema borra el evento de la lista y cambia el color del evento en el calendario a rojo.

- Gestionar Eventos

Tabla 8 Especificación del CUS Gestionar Eventos

Términos	Definición		
Caso de Uso	Gestionar Eventos		
Pre-condición	Existen Eventos registrados en el sistema.		
Post-condición	Los datos de uno o más han sido modificados.		
Actor	Consistorio		
	Miembro del consistorio		Sistema
Flujo Principal Estado Evento: <ul style="list-style-type: none"> • Aceptado 	1	El CUS se inicia cuando el consistorio elige 'Calendario' en el menú de opciones.	
			2 El sistema muestra un calendario donde se ven todos los eventos del sistema.
	3	El consistorio hace click en un evento del calendario.	
			4 El sistema abre un modal en el que se muestra un formulario que contiene los siguientes campos: <ul style="list-style-type: none"> • Nombre del Evento • Propósito • Descripción • Imagen • Dirigido A • Lugar • Día de Inicio • Día de Fin • Hora de Inicio • Hora de Fin

				También cuenta con los siguientes botones: <ul style="list-style-type: none"> • Guardar. Inicialmente aparece desactivado • Planificar • Cancelar
	5	El consistorio realiza los cambios que necesite.		
			6	El sistema activa el botón "Guardar".
	7	El consistorio hace click en "Guardar".		
			8	El sistema guarda los cambios en la BD.
			9	El sistema realiza los cambios necesarios en la interfaz.
			10	Se termina el CUS.
Flujo Alternativo Cancelar	5	El consistorio hace click en "Cancelar".		
			6	El sistema abre un pop un con un campo para ingresar la razón de la acción y un botón para confirmar la cancelación del evento.
	7	El consistorio llena el campo y hace click en "Cancelar".		
			8	El sistema cambia el estado del evento a "Cancelado".
			9	El sistema registra la operación en la BD con el mensaje que ha puesto el

				consistorio.
			10	El sistema cierra todos los modales y cambia el color del evento en el calendario a rojo.
Flujo Alternativo Estado Evento:			4	El sistema muestra un popup donde se podrá ver la información básica del evento, así como el comentario de la persona que lo ha solicitado o rechazado.
<ul style="list-style-type: none"> • Solicitado • Rechazado 				
Flujo Alternativo Estado Evento:			4	El sistema abre una nueva pestaña en el navegador con la pantalla de “planificar”. <i>Se inicia el CUS Planificar.</i>
<ul style="list-style-type: none"> • En Curso • Cancelado • Finalizado • Cerrado 				

- Gestionar Objetivos

Ver Tabla 4 en página 63

- Gestionar Obstáculos

Ver Tabla 5 en página 78

- Participar en Evento

Tabla 9 Especificación de CUS Participar en Evento

Términos		Definición	
Caso de Uso		Participar en Evento	
Pre-condición		Existen Eventos Aceptados.	
Post-condición		La persona ha sido registrada como un participante del evento.	
Actor		Cristiano	
		Cristiano	Sistema
Flujo Principal	1	El CUS se inicia cuando el cristiano (logueado) ingresa a la página “Evento” de la página web de la iglesia.	
			2 El sistema muestra una página con los datos del Evento: <ul style="list-style-type: none"> • La imagen del evento • Nombre • Descripción • Público Dirigido • Fecha de inicio • Hora de inicio • Fecha de fin • Hora de fin • Lugar <ul style="list-style-type: none"> • Descripción • Dirección • Servicios • Teléfonos Muestra un botón por cada escenario cuando el evento aún no haya iniciado:

			<ul style="list-style-type: none"> • Usuario: no mostrará el botón de participación. • Usuario Logueado: muestra un botón para que pueda participar. • Participante: muestra un botón con el mensaje: “Ya eres un participante!” <p>En caso de que el evento se encuentre en curso, mostrará un botón con el mensaje: “Evento en curso”. En caso de que el evento se encuentre finalizado o cerrado, mostrará un botón con el mensaje: “Evento terminado”.</p>
	3	El cristiano hace click en “Participar”.	
			4 El sistema registra a la persona como un participante del evento, en la BD.
			5 El sistema recarga la página.
			6 Se termina el CUS.

6.3.4. Diagramas de clases Análisis

- **Solicitar Evento**

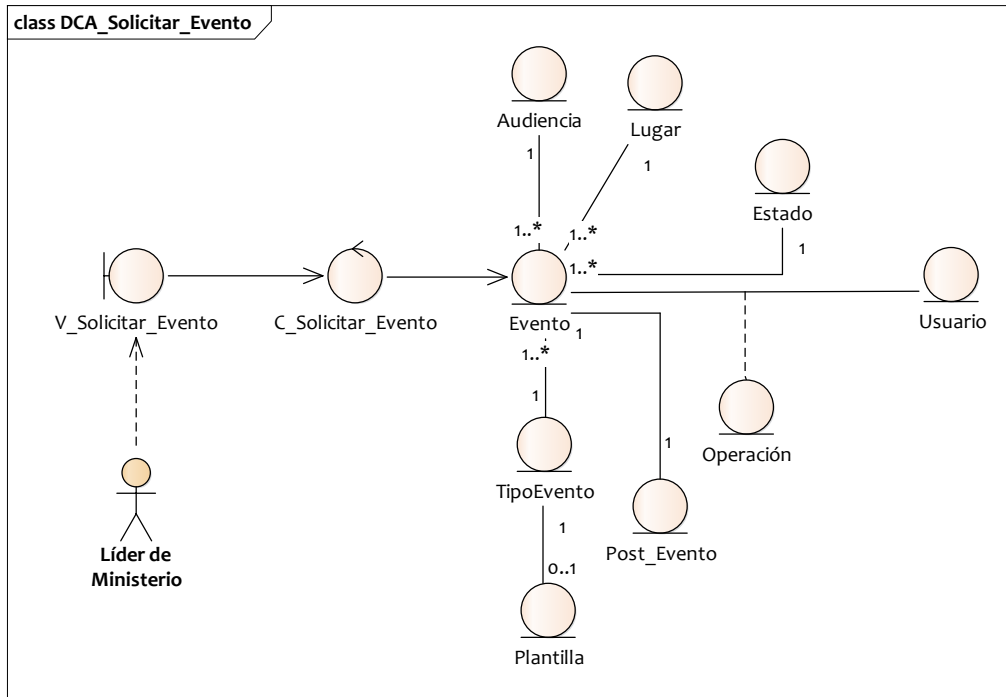


Figura 48 Diagrama de clases análisis del CUS Solicitar Evento

Fuente: Elaboración propia

- **Evaluar Evento**

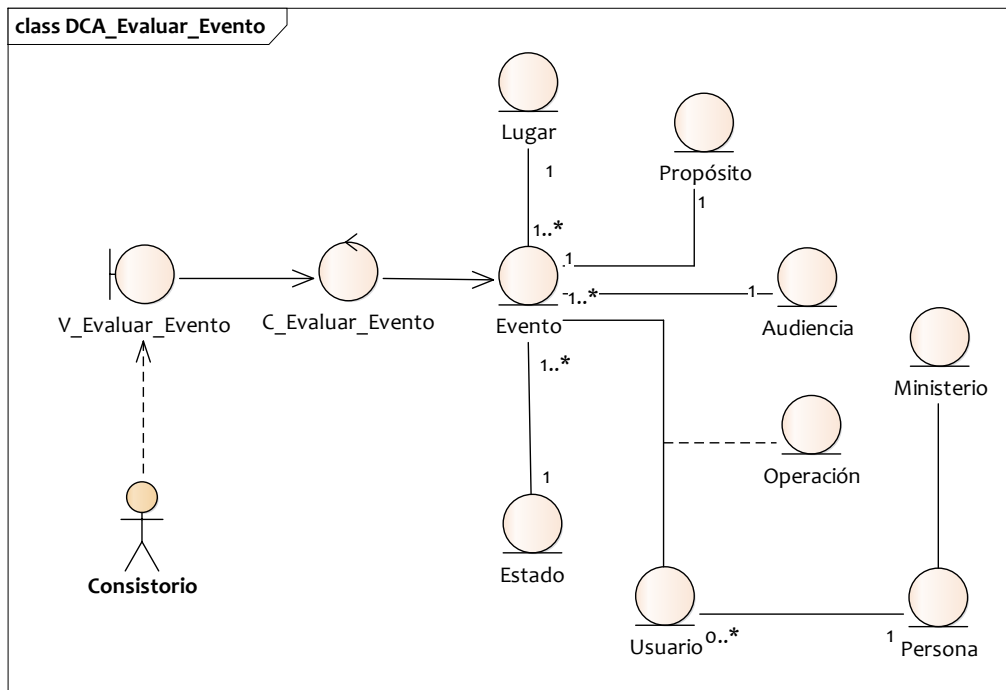


Figura 49 Diagrama de clases análisis del CUS Evaluar Evento

Fuente: Elaboración propia

- **Gestionar Eventos**

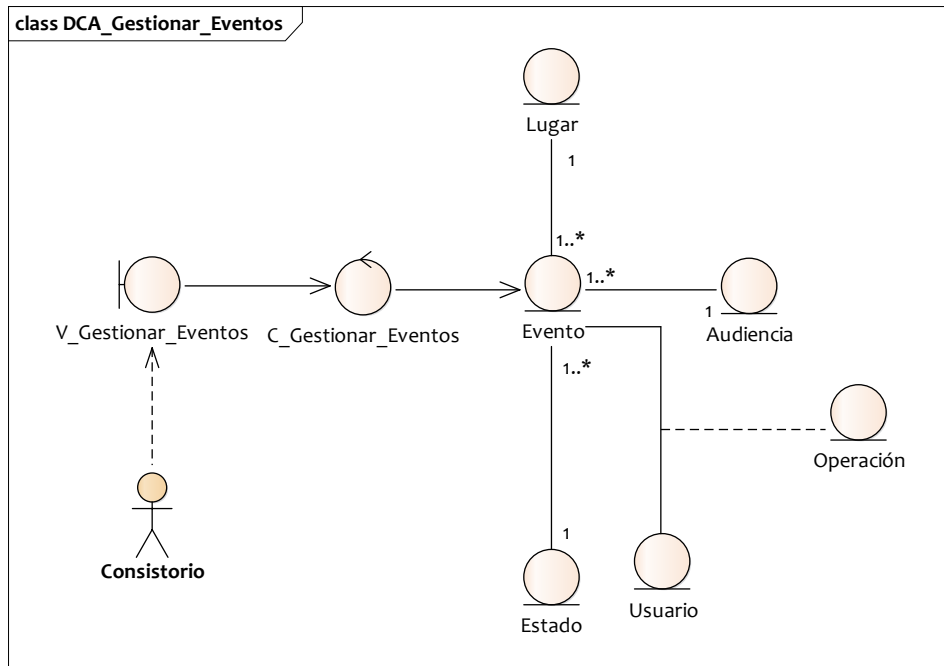


Figura 50 Diagrama de clases análisis del CUS Gestionar Eventos

Fuente: Elaboración propia

- **Gestionar Objetivos**

Referencia Figura 40 página 70

- **Gestionar Obstáculos**

Referencia Figura 45 página 82

- **Participar en Evento**

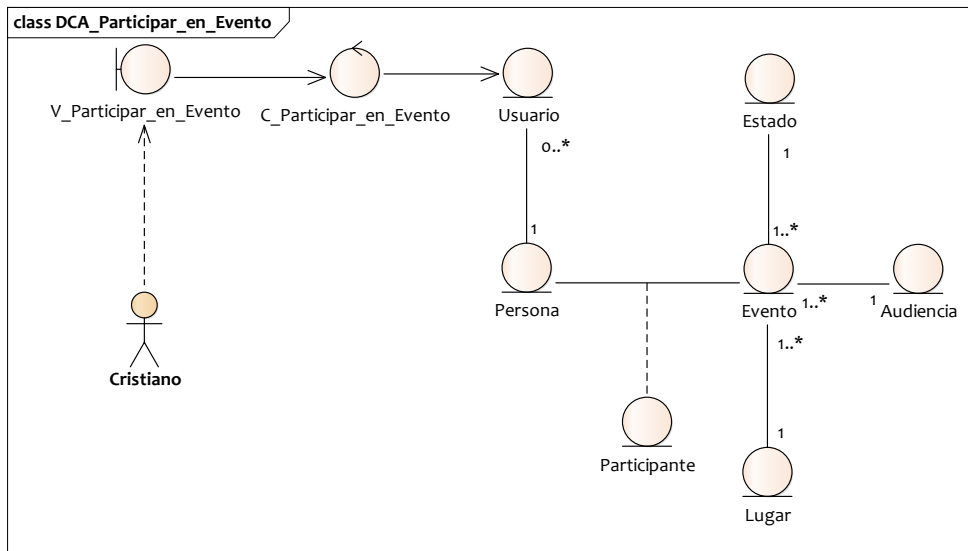


Figura 51 Diagrama de clases análisis Participar en Evento

Fuente: Elaboración propia

6.3.5. Diagramas de Diseño

- **Solicitar Evento**

Diagrama de Clases

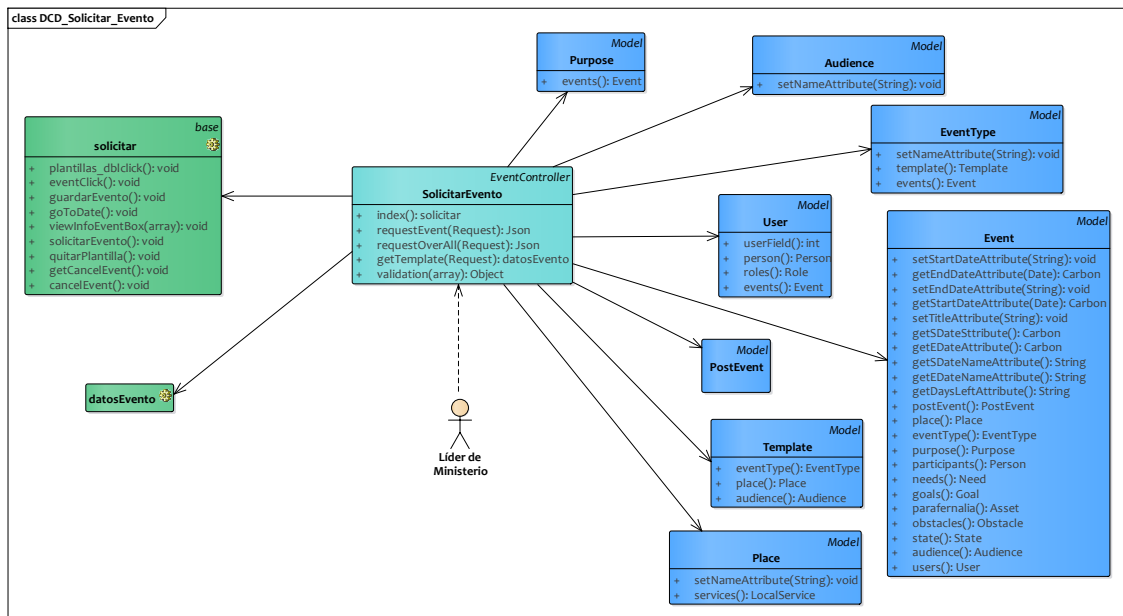


Figura 52 Diagrama de clases diseño del CUS Solicitar Evento

Fuente: Elaboración propia

Diagrama de Secuencia

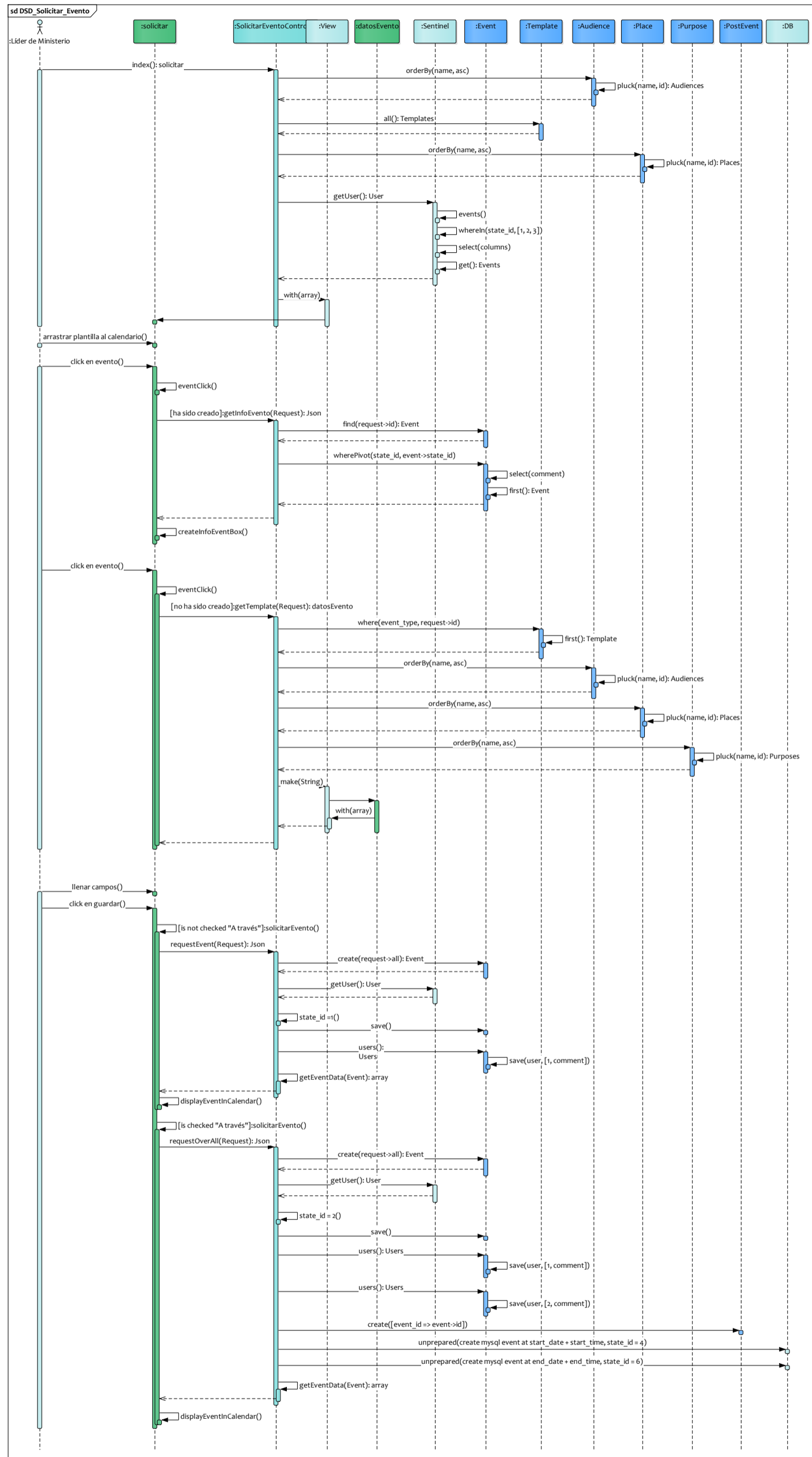


Figura 53 Diagrama de secuencia diseño del CUS Solicitar Evento

Fuente: Elaboración propia

- **Evaluar Evento**

Diagrama de Clases

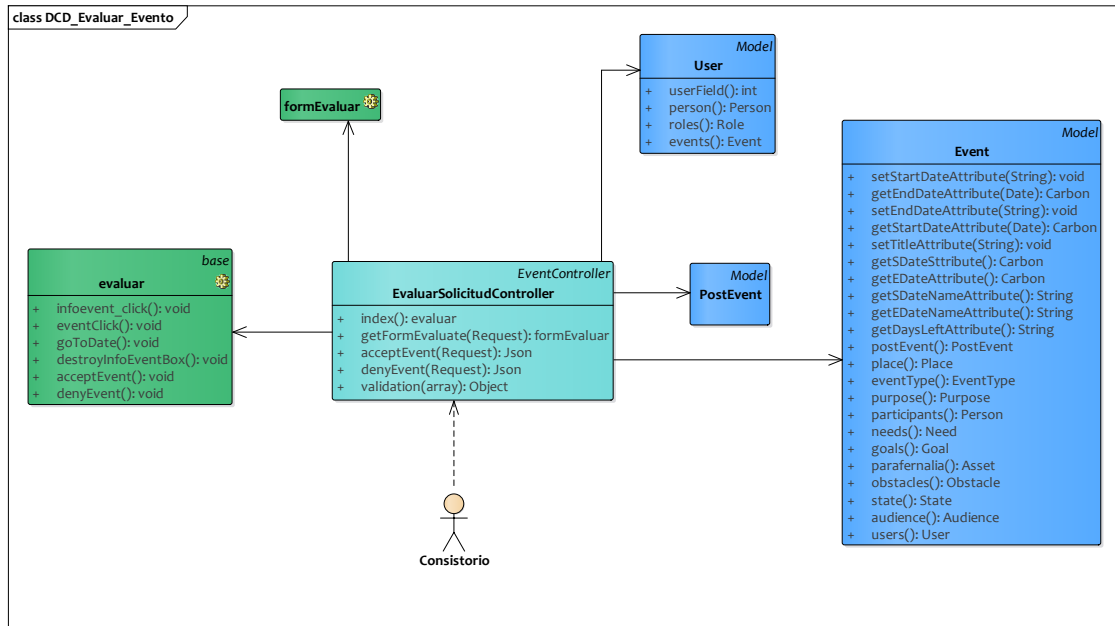


Figura 54 Diagrama de clase diseño del CUS Evaluar Evento

Fuente: Elaboración propia

Diagrama de Secuencia

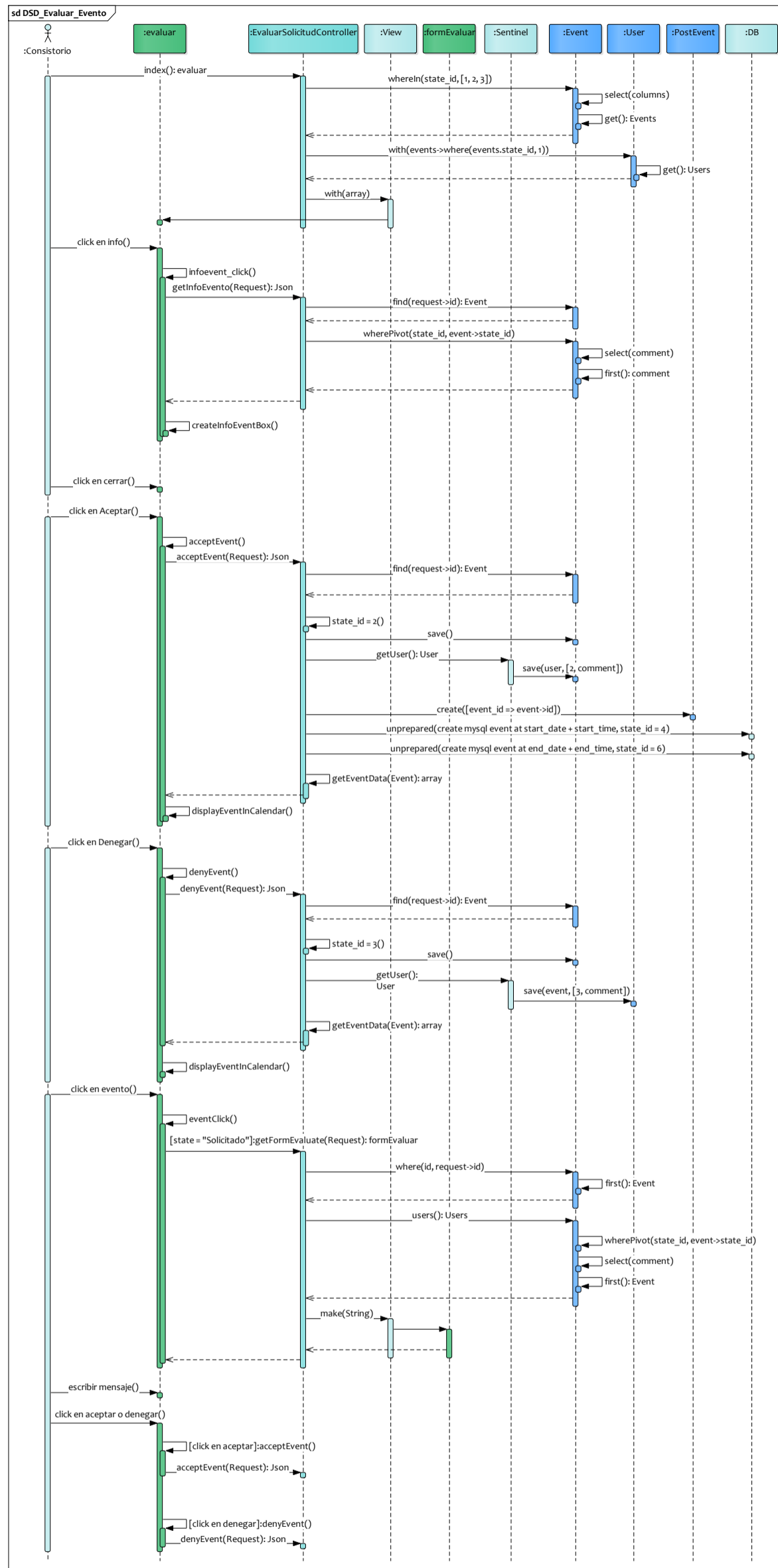


Figura 55 Diagrama de secuencia diseño del CUSEvaluar Evento

Fuente: Elaboración propia

- **Gestionar Eventos**

Diagrama de Clases

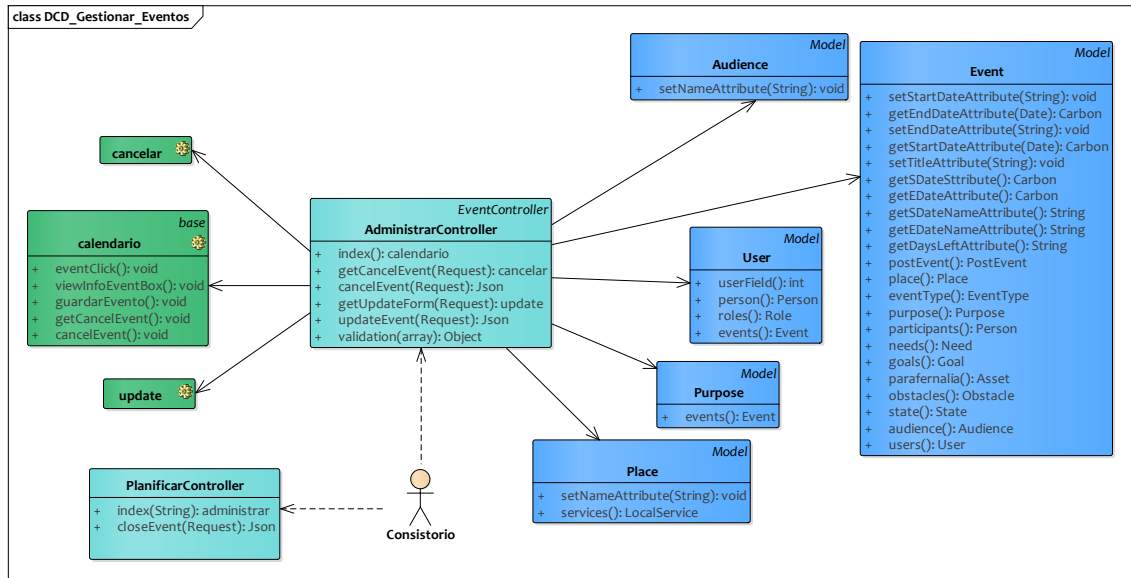


Figura 56 Diagrama de clases diseño del CUS Gestionar Eventos

Fuente: Elaboración propia

Diagrama de Secuencia

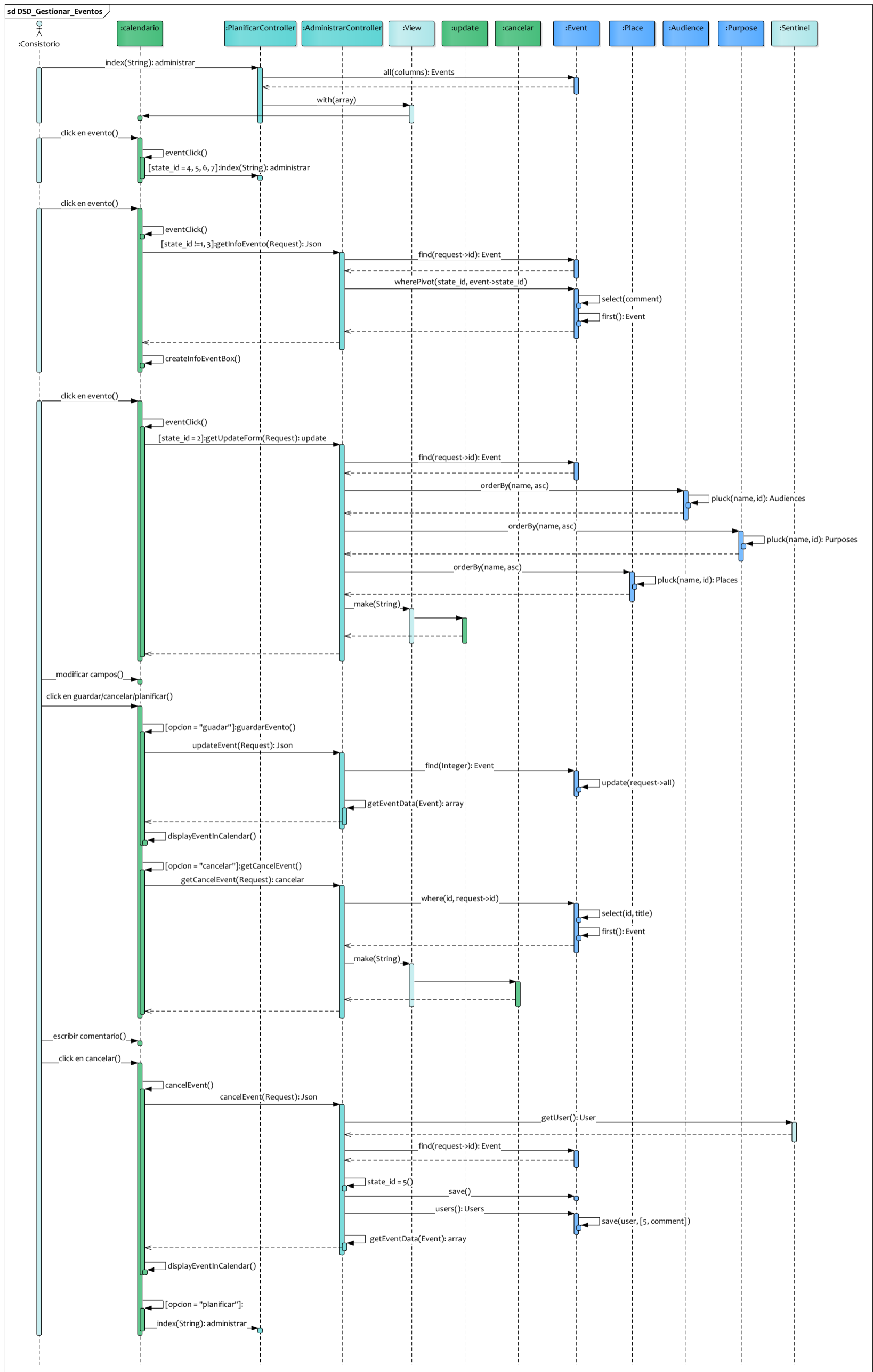


Figura 57 Diagrama de secuencia diseño del CUS Gestionar Eventos

Fuente: Elaboración propia

- **Gestionar Objetivos**

Diagrama de Clases

Referencia Figura 41 página 71

Referencia Figura 42 página 71

Diagrama de Secuencia

Referencia Figura 43 página 74

Referencia Figura 44 página 77

- **Gestionar Obstáculos**

Diagrama de Clases

Referencia Figura 46 Página 83

Diagrama de Secuencia

Referencia Figura 47 página 86

- **Participar en Evento**

Diagrama de Clases

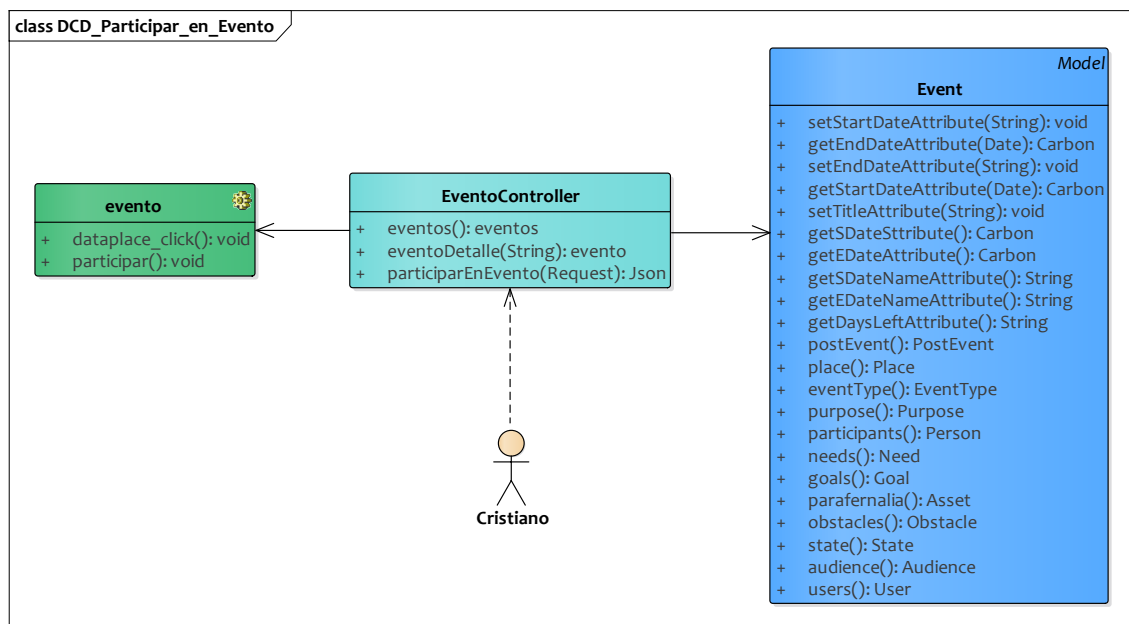


Figura 58 Diagrama de clases diseño del CUS Participar en Evento

Fuente: Elaboración propia

Diagrama de Secuencia

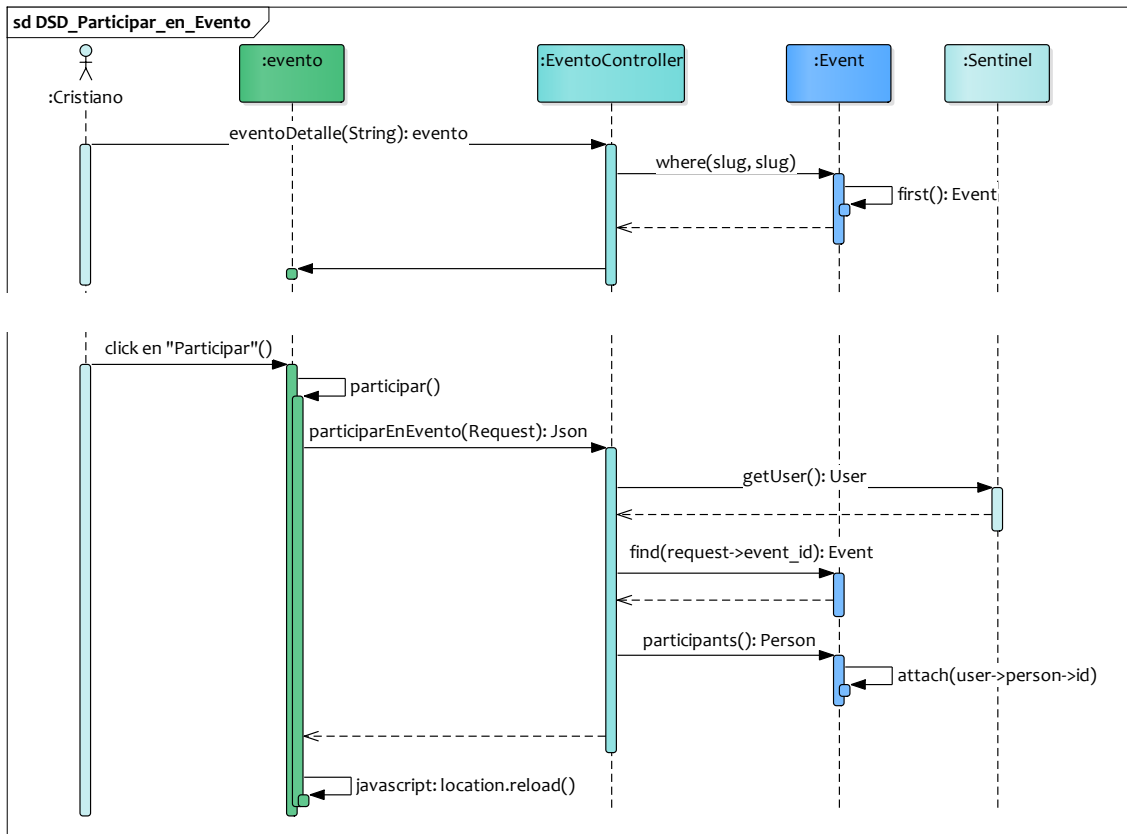


Figura 59 Diagrama de secuencia de diseño del CUS Participar en Evento

Fuente: Elaboración propia

- **Diagrama de Estados para la clase Evento**

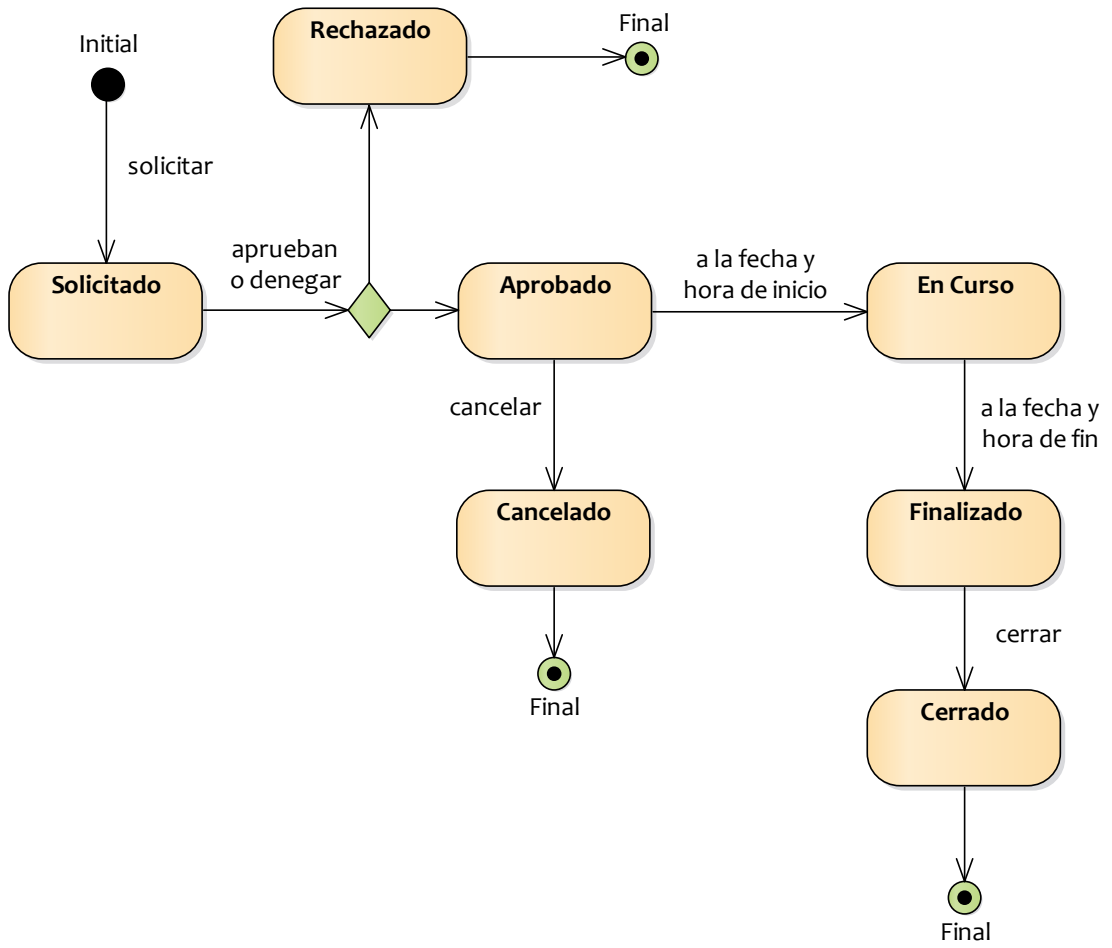


Figura 60 Diagrama de Estados

Fuente: Elaboración propia

6.4. Vista Lógica

6.4.1. Vista de Componentes

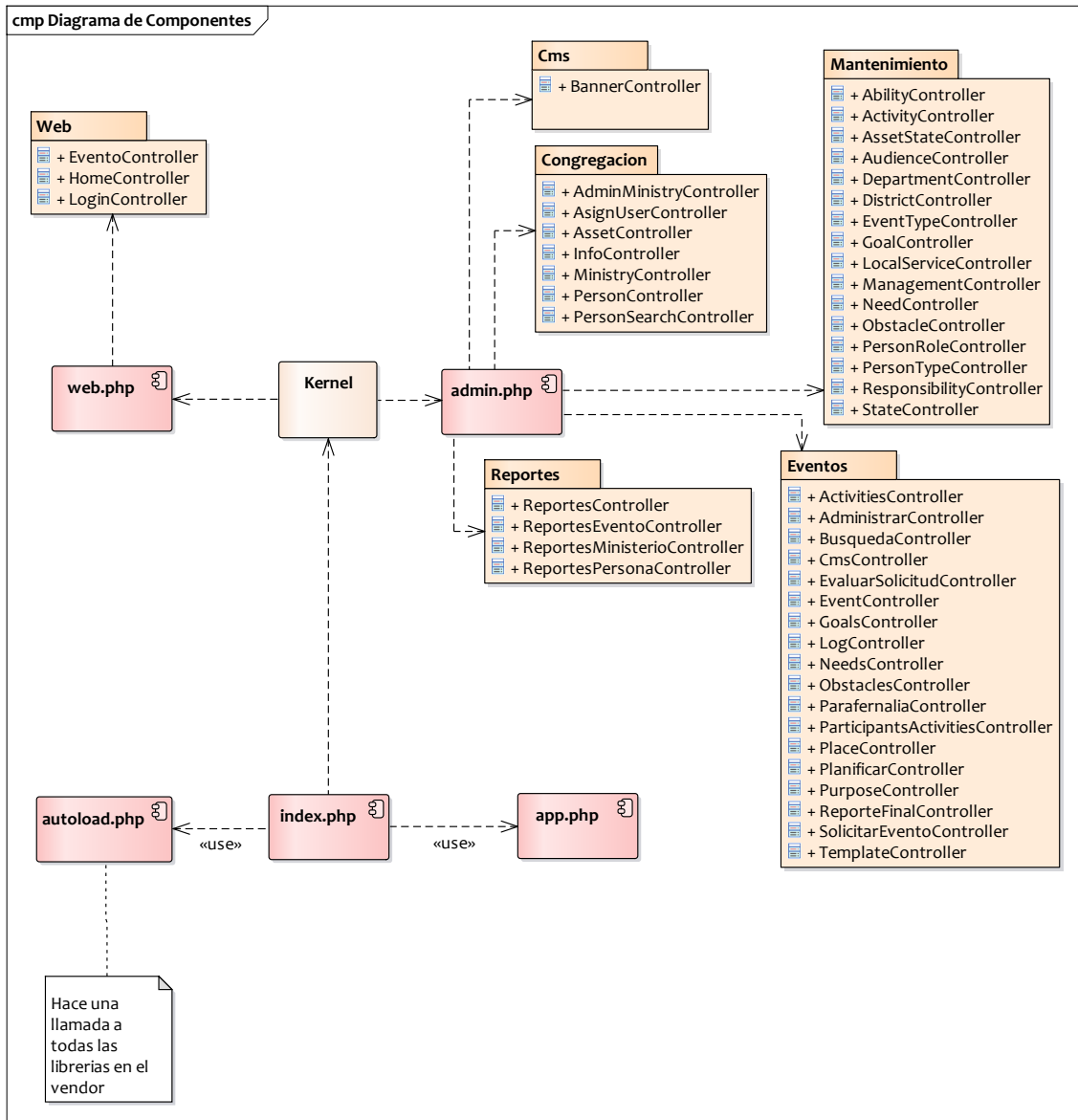


Figura 61 Diagrama de Componentes

Fuente: Elaboración propia

6.4.2. Vista de Despliegue

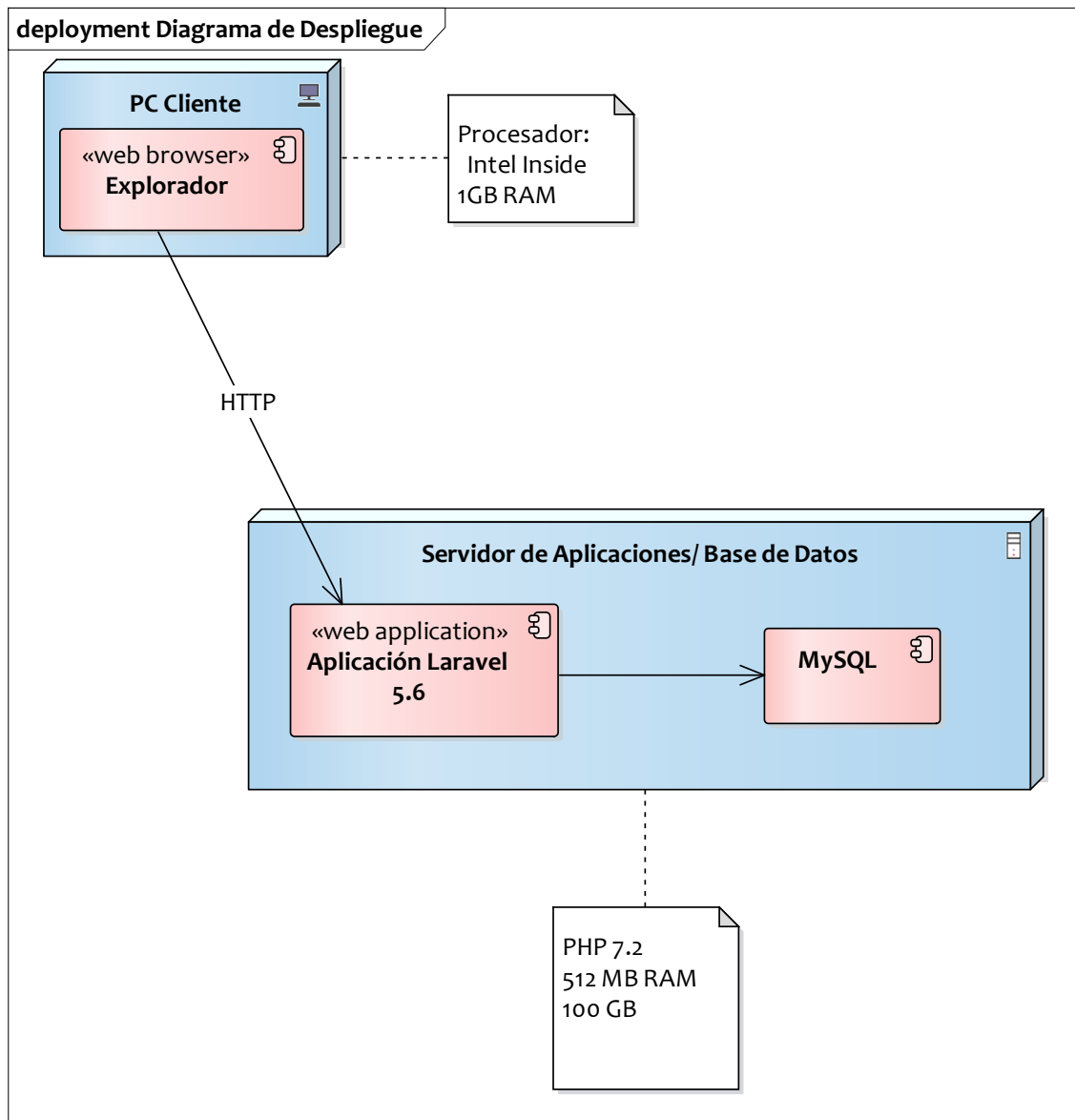


Figura 62 Diagrama de despliegue

Fuente: Elaboración propia

643. Modelo Físico

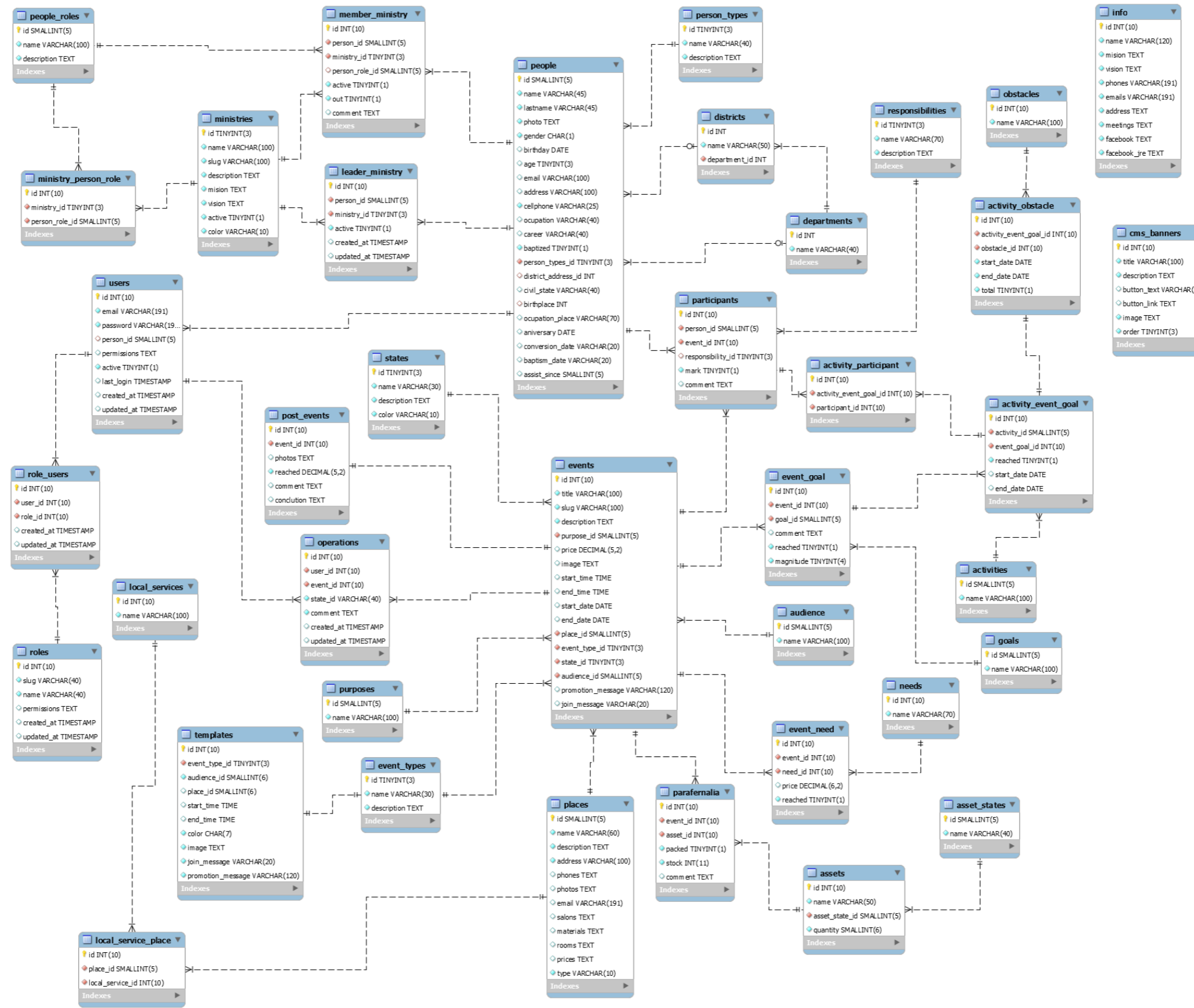


Figura 63 Modelo físico de la base de datos

Fuente: Elaboración propia

6.4.4. Diccionario de Datos

Tabla 10 Diccionario de datos

Activities			
id	smallint	not null	Autoincrement
name	varchar(100)	not null	
activity_event_goal			
id	int	not null	autoincrement
activity_id	smallint	not null	FK: activities
event_goal_id	int	not null	FK: event_goal
reached	tinyint	not null	Default: 0
start_date	date	null	
end_date	date	null	
activity_obstacle			
id	int	not null	autoincrement
activity_event_goal_id	int	not null	FK: activity_event_goal
obstacle_id	int	not null	
start_date	date	not null	
end_date	date	not null	
total	tinyint	not null	Default: 0
activity_participant			
id	int	not null	autoincrement
activity_event_goal_id	int	not null	FK: activity_event_goal
participant_id	int	not null	FK: participants
assets			
id	int	not null	Autoincrement
name	varchar(100)	not null	
asset_state_id	smallint	not null	FK: asset_states
quantity	smallint	not null	
asset_states			
id	smallint	not null	Autoincrement
name	varchar(40)	not null	
audience			
id	smallint	not null	Autoincrement

name	varchar(100)	not null	
cms_banners			
id	smallint	not null	Autoincrement
title	varchar(100)	not null	
description	text	not null	
button_text	varchar(30)	null	
button_link	text	null	
image	text	not null	
order	tinyint	not null	
departments			
id	int	not null	Autoincrement
name	varchar(40)	not null	
districts			
id	int	not null	Autoincrement
name	varchar(50)	not null	
department_id	int	not null	FK: departments
events			
id	int	not null	Autoincrement
title	varchar(100)	not null	
slug	varchar(100)	not null	
description	text	not null	
purpose_id	smallint	not null	FK: purposes
price	decimal(5,2)	null	
image	text	null	
start_time	time	null	
end_time	time	null	
start_date	date	null	
end_date	date	null	
place_id	smallint	not null	FK: places
event_type_id	tinyint	not null	FK: event_types
state_id	tinyint	not null	Default: 1
audience_id	smallint	not null	
promotion_message	varchar(120)	null	

join_message	varchar(20)	null	
event_goal			
id	int	not null	Autoincrement
event_id	int	not null	FK: events
goal_id	smallint	not null	FK: goals
reached	tinyint	not null	Default: 0
comment	text	null	
magnitude	tinyint	not null	Default: 20
event_need			
id	int	not null	Autoincrement
event_id	int	not null	FK: events
need_id	int	not null	FK: needs
price	decimal(6,2)	null	
reached	tinyint	not null	Default: 0
event_types			
id	tinyint	not null	Autoincrement
name	varchar(30)	not null	
description	text	not null	
goals			
id	smallint	not null	Autoincrement
name	varchar(100)	not null	
info			
id	int	not null	Autoincrement
name	varchar(120)	not null	
mision	text	not null	
vision	text	not null	
phones	varchar(191)	not null	
emails	varchar(191)	not null	
address	text	not null	
meetings	text	not null	
facebook	text	not null	
facebook_jre	text	not null	

leader_ministry			
id	int	not null	Autoincrement
person_id	smallint	not null	FK: people
ministry_id	tinyint	not null	FK: ministries
local_services			
id	smallint	not null	Autoincrement
name	varchar(100)	not null	
local_service_place			
id	Int	not null	Autoincrement
place_id	smallint	not null	FK: places
local_service_id	Int	not null	FK: local_services
member_ministry			
id	Int	not null	Autoincrement
person_id	smallint	not null	FK: people
ministry_id	Tinyint	not null	FK: ministries
person_role_id	smallint	null	FK: people_roles
active	Tinyint	not null	Default: 1
ministries			
id	tinyint	not null	Autoincrement
name	varchar(100)	not null	
slug	varchar(120)	not null	
description	text	not null	
mision	text	not null	
vision	text	not null	
active	tinyint	not null	Default: 0
color	varchar(10)	not null	
ministry_person_role			
id	int	not null	Autoincrement
ministry_id	tinyint	not null	FK: ministries
person_role_id	smallint	not null	FK: people_roles
needs			
id	int	not null	Autoincrement

name	varchar(70)	not null	
obstacles			
id	int	not null	Autoincrement
name	varchar(100)	not null	
operations			
id	int	not null	Autoincrement
user_id	int	not null	FK: users
event_id	int	not null	FK: events
state_id	varchar(40)	not null	
comment	text	not null	
created_at	date	null	
updated_at	date	null	
parafernalialia			
id	int	not null	Autoincrement
event_id	int	not null	FK: events
asset_id	int	not null	FK: assets
packed	tinyint	not null	Default: 0
stock	int	not null	
comment	text	null	
participants			
id	int	not null	Autoincrement
person_id	smallint	not null	FK: people
event_id	int	not null	FK: events
responsibility_id	tinyint	Null	FK: responsibilities
mark	tinyint	not null	Default: 0
comment	text	Null	
people			
id	Smallint	not null	Autoincrement
name	varchar(45)	not null	
lastname	varchar(45)	not null	
photo	text	not null	
gender	char(1)	not null	
birthday	date	Null	

age	tinyint	null	
email	varchar(100)	null	
address	varchar(100)	null	
cellphone	varchar(25)	null	
ocupation	varchar(40)	null	
career	varchar(40)	null	
baptized	tinyint	not null	
person_types_id	tinyint	not null	FK: person_types
district_address_id	int	null	FK: districts
civil_state	varchar(40)	null	
birthplace	int	null	FK: departments
ocupation_place	varchar(70)	null	
aniversary	date	null	
conversion_date	varchar(20)	null	
baptism_date	varchar(20)	null	
assist_since	smallint	null	
people_roles			
id	smallint	not null	Autoincrement
name	varchar(100)	not null	
description	text	not null	
person_types			
id	smallint	not null	Autoincrement
name	varchar(100)	not null	
description	text	not null	
places			
id	smallint	not null	Autoincrement
name	varchar(60)	not null	
description	text	not null	
address	text	not null	
phones	text	null	
photos	text	null	
email	varchar(191)	null	
salons	text	null	

materials	Text	null	
rooms	text	null	
prices	text	null	
type	varchar(10)	not null	
post_events			
id	int	not null	Autoincrement
event_id	int	not null	FK: events
photos	text	null	
reached	decimal(5,2)	not null	Default: 0
comment	text	null	
conclusion	text	null	
purposes			
id	smallint	not null	Autoincrement
name	varchar(100)	not null	
responsibilities			
id	tinyint	not null	Autoincrement
name	varchar(70)	not null	
description	text	not null	
roles			
id	int	not null	Autoincrement
Slug	varchar(40)	not null	
Name	varchar(40)	not null	
Permissions	text	null	
created_at	timestamp	null	
updated_at	timestamp	null	
role_users			
Id	int	not null	Autoincrement
user_id	int	not null	FK: users
role_id	int	not null	FK: roles
created_at	timestamp	null	
updated_at	timestamp	null	
states			
Id	tinyint	not null	Autoincrement

Name	varchar(30)	not null	
Description	text	not null	
Color	varchar(10)	not null	
templates			
Id	int	not null	Autoincrement
event_type_id	tinyint	not null	FK: event_types
audience_id	smallint	not null	
place_id	smallint	null	
start_time	time	null	
end_time	time	null	
Image	text	not null	
join_message	varchar(20)	not null	
promotion_message	varchar(20)	not null	
users			
Id	int	not null	Autoincrement
Email	varchar(191)	not null	
Password	varchar(191)	not null	
person_id	smallint	null	FK: people
Permissions	text	null	
Active	tinyint	not null	Default: 1
last_login	timestamp	null	
created_at	timestamp	null	
updated_at	timestamp	null	

6.5. Calidad

La implementación del sistema debe estar constituida dentro de los rangos de calidad establecidos, por esa razón el mismo es compatible con los siguientes requisitos:

- El sistema puede ejecutarse en los siguientes navegadores: Chrome, Firefox, Microsoft Edge, Edge Chromium.
- El sistema estará disponible las 24 horas, los 365 días del año, debido a que se encuentra alojado en un servidor en internet.

En el capítulo siguiente se presentará los resultados de las validaciones en las pruebas unitarias y de integración que se ha llevado a cabo.

6.6. Tecnologías a utilizar para la implementación

Los lenguajes de programación, los frameworks, gestores, etc., utilizados para el desarrollo e implementación del sistema son los siguientes:

- Lenguajes de programación: PHP, Javascript
- Entorno de programación: Visual Studio Code
- Framework frontend: Bootstrap
- Framework backend: Laravel
- Gestor de base de datos: Mysql
- Gestor de versiones: Git (Bitbucket)

CAPÍTULO VII

7.1. Desarrollo

Cada tarea que se ha desarrollado para la culminación del presente proyecto ha sido dirigida por estándares de documentación, pruebas e implementación, para que la lectura y escritura de estas, sean más legibles y entendibles para diversas personas. A continuación, les presento los estándares presentes en el proyecto.

7.1.1. Estándares de codificación

La importancia de los estándares dentro del código del sistema radica en la dificultad que presenta el mismo a la hora de leer el mismo. En un futuro, cercano o lejano, el código escrito necesitará ser modificado, para esto primero debe ser entendido por el programador y el orden tiene un factor importante en esta tarea, ya que un código ordenado y estandarizado en todo el sistema, es fácil de leer.

Tabla 11 Estándares de codificación

Tipo	Lenguaje	Descripción
Variables	Javascript	Escritas en Snake Case
	PHP	Escritas en Camel Case
	Json	Escritas en Snake Case
		Para ids, se aplica la siguiente forma: <i>campo_id</i> . Ejmp: evento_id
Funciones	General	Nombres en inglés
		Escritas en Camel Case
Clases PHP	General	Escritas en Pascal Case
		Nombres en Inglés
	Controladores	Para controladores, se aplica la siguiente forma: <i>ModeloOFuncionalidadController</i> . Ejmp: EventController, RequestEventController

7.1.2. Estándares de campos de tablas en la Base de Datos

Los estándares para los campos de todas las tablas de la base se muestran según la siguiente tabla.

Tabla 12 Estándares en la base de datos

Tipo	Nomenclatura	Ejemplo
Tabla	name_table (en plural)	events
Tabla Muchos a Muchos	name_table (en singular)	event_goal
Primary Key	id	id
Foreign Key	foreign_table_name_id (en singular)	event_goal_id

7.1.3. Estándares de documentación

Los estándares de la documentación en cuando a fuentes como a nombres de los documentos, se describen en las siguientes tablas:

Tabla 13 Estándares en la documentación

Clasificación	Tipo de Letra	Tamaño	Otros	Interlineado
Contenido	Arial	12	Normal	Sencillo
Títulos	Arial	14	Negrita	Sencillo
Subtítulos	Arial	12	Negrita	Sencillo
Pie de Página	Arial	12	Normal	Sencillo

Tabla 14 Estándares de los títulos de los documentos

Tipo de documento	Nomenclatura
Documento del Negocio	Neg_Nombre_De_Plantilla
Especificaciones de CUS	ECUS_Nombre_del_CUS
Documentos del sistema	RA_Nombre_de_Plantilla

7.2. Pruebas

7.2.1. Plan de pruebas

El presente plan de pruebas describe la validación de las funciones principales del presente sistema.

Tabla 15 Plan de pruebas

Plan de Pruebas: Rumbo Ágil						
#	Tipo de Unidad	Unidad de prueba	Descripción	Fecha Planificada	Tester	Responsable de la Unidad
1	PCUS	Solicitar Evento	El usuario será capaz de solicitar un evento para su evaluación.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
2	PCUS	Solicitar Evento	El usuario será capaz de solicitar un evento para su aprobación automática.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
3	PCUS	Evaluar Solicitud	El usuario será capaz de aceptar o rechazar una solicitud de evento, rápidamente.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
4	PCUS	Evaluar Solicitud	El usuario será capaz de aceptar o rechazar una solicitud de evento y adjuntar un mensaje a la acción.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
5	PCUS	Gestionar Eventos	El usuario será capaz de actualizar los datos y/o el estado de los eventos que elija	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
6	PCUS	Gestionar Objetivos	El usuario será capaz de agregar objetivos a un evento aceptado o en curso.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
7	PCUS	Gestionar Objetivos	El usuario podrá agregar actividades a los objetivos de un evento.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
8	PCUS	Gestionar Obstáculos	El usuario podrá asignar un obstáculo a una actividad del evento.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla

9	PCUS	Participar en Evento	El usuario será capaz de confirmar su participación a un evento mediante la web.	05/01/2019	Kevin Narváez	Ricardo Alama Ventocilla
---	------	----------------------	--	------------	---------------	--------------------------

7.2.2. Informe de pruebas: Solicitar Evento

Tabla 16 Informe de pruebas del CUS Solicitar Evento

Informe de prueba						
Unidad de Prueba		PCUS_Solicitar_Evento				
Fecha:	15/012019	Avance	100%	Tester	Kevin Narvaez	
Descripción de la prueba:						
El objetivo de la prueba es verificar la funcionalidad de todas las acciones posibles a momento de solicitar un evento.						
#	Tipo	Descripción	Resultado esperado	Resultado	Detalles	
1	Prueba Unitaria	Se solicita un evento de forma regular. Arrastrando una plantilla al calendario y llenando todos los campos obligatorios, excepto la casilla "A través".	El evento se registra con estado solicitado y se muestra en el calendario con el color respectivo.	Exitoso	El registro fue un éxito y se mostró el evento.	
2	Prueba Unitaria	Se solicita un evento de forma rápida, arrastrando una plantilla al calendario, llenando todos los campos obligatorios y activando la casilla "A través".	El evento se registra con estado aceptado y se muestra en el calendario con el color respectivo.	Exitoso	El registro fue un éxito y el eventos está listo para su planificación.	
3	Prueba Unitaria	Aparece un popup de información sobre un evento en particular, cuando se hace click sobre el mismo ubicado en el calendario.	El popup aparece en la pantalla con los datos básicos del evento elegido.	Exitoso	El popup se mostró exitosamente y se vio la información del evento.	

7.2.3. Informe de pruebas: Evaluar Solicitud

Tabla 17 Informe de pruebas del CUS Evaluar Solicitud

Informe de prueba					
Unidad de Prueba		PCUS_Evaluar_Solicitud			
Fecha:	15/012019	Avance	100%	Tester	Kevin Narvaez
Descripción de la prueba:					
El objetivo de la prueba es verificar la funcionalidad de todas las acciones posibles a momento de evaluar la solicitud de un evento.					
#	Tipo	Descripción	Resultado esperado	Resultado	Detalles
1	Prueba Unitaria	Se acepta o se rechaza la solicitud de un evento, haciendo click en un evento del calendario, escribiendo un comentario y haciendo click en el botón de aceptar o rechazar.	El evento cambia al estado aceptado o rechazado, el comentario ingresado ha sido registrado en el sistema, cambia de color en el calendario y se elimina el registro en la lista de pendientes.	Exitoso	El evento ha cambiado al estado elegido, el color en el calendario cambió y el registro en la lista de pendientes desapareció.
2	Prueba Unitaria	Se acepta o se rechaza la solicitud de un evento rápidamente, haciendo click en el botón de la lista en la tarjeta de pendientes.	El evento cambia al estado aceptado o rechazado, el color del evento cambia y el registro en la lista de pendientes desaparece.	Exitoso	El evento ha cambiado al estado elegido, el color en el calendario cambió y el registro en la tarjeta de pendientes desapareció.
3	Prueba Unitaria	Aparece un popup de información sobre un evento en particular, cuando se hace click sobre el mismo ubicado en el calendario.	El popup aparece en la pantalla con los datos básicos del evento elegido.	Exitoso	El popup se mostró exitosamente y se vio la información del evento.

7.2.4. Informe de pruebas: Gestionar Eventos

Tabla 18 Informe de pruebas del CUS Gestionar Eventos

Informe de prueba					
Unidad de Prueba		PCUS_Gestionar_Eventos			
Fecha:	15/012019	Avance	100%	Tester	Kevin Narvaez
Descripción de la prueba:					
El objetivo de la prueba es verificar la funcionalidad de todas las acciones posibles a momento de actualizar la información de los eventos					
#	Tipo	Descripción	Resultado esperado	Resultado	Detalles
1	Prueba Unitaria	Se hace click en un evento del calendario y se realizan cambios en la información básica de los eventos y se actualiza en el sistema.	La información de un evento se modificará en el sistema.	Exitoso	La información básica del evento ha sido modificada exitosamente.
2	Prueba Unitaria	Se hace click en un evento del calendario y se cancela un evento.	El evento cambia al estado cancelado y el color del mismo en el calendario cambia.	Exitoso	El evento ha cambiado al estado cancelado y su color en el calendario cambió.
3	Prueba Unitaria	Aparece un popup de información sobre un evento en particular, cuando se hace click sobre el mismo ubicado en el calendario.	El popup aparece en la pantalla con los datos básicos del evento elegido.	Exitoso	El popup se mostró exitosamente y se vio la información del evento.
4	Prueba Unitaria	Se hace click en un evento correspondiente y se accede a la planificación del mismo.	La página de planificación aparecerá lista para los cambios.	Exitoso	Se abrió en una pestaña nueva la página de planificación y de acuerdo al estado del evento, algunas opciones se activarán o desactivarán.

7.2.5. Informe de pruebas: Gestionar Objetivos

Tabla 19 Informe de pruebas de CUS Gestionar Objetivos

Informe de prueba					
Unidad de Prueba		PCUS_Gestionar_Objetivos			
Fecha:	15/012019	Avance	100%	Tester	Kevin Narvaez
Descripción de la prueba:					
El objetivo de la prueba es verificar la funcionalidad de todas las acciones posibles a momento de agregar, eliminar, objetivos y actividades					
#	Tipo	Descripción	Resultado esperado	Resultado	Detalles
1	Prueba Unitaria	Se hace click en la etiqueta Objetivos y aparece la tarjeta de objetivos.	Aparece la tarjeta de objetivos.	Exitoso	La tarjeta de objetivos se mostró en la página con todos los objetivos y actividades registradas.
2	Prueba Unitaria	Se hace click en la lista de objetivos, se elige un objetivo, se indica la magnitud y se hace click en el botón de agregar.	El objetivo se agrega a la lista de objetivos del evento, con el la magnitud indicada.	Exitoso	La lista de objetivos se actualiza y se muestra toda la información.
3	Prueba Unitaria	Se hace click en la casilla “alcanzado” para indicar la realización del objetivo.	El objetivo cambia a estado realizado y el porcentaje de realización del evento se actualiza.	Exitoso	El objetivo cambió de estado a realizado y el porcentaje total del evento aumenta la magnitud del objetivo.
4	Prueba Unitaria	Se hace click en el botón “X” al costado de un objetivo realizado para eliminarlo del evento.	El objetivo se elimina del evento y el porcentaje se reduce.	Exitoso	El objetivo ha sido eliminado con éxito y el porcentaje total de evento se redujo.
5	Prueba	Se hace click en el botón	El objetivo se elimina	Exitoso	El objetivo ha

	Unitaria	“X” al costado de un objetivo no realizado para eliminarlo del evento.	del evento.		sido eliminado con éxito y la lista de objetivos se actualiza.
6	Prueba Unitaria	Se hace click en la lista de actividades, se elige el objetivo, se indica la fecha de inicio y fin de la actividad y se hace click en el botón de añadir.	La actividad se añade al objetivo elegido.	Exitoso	La actividad se ha añadido al objetivo correspondiente, el porcentaje total del evento se actualiza si así se requiere y la lista de objetivos se actualizó.
7	Prueba Unitaria	Se hace click en el botón “X” al costado de una actividad realizada para eliminarla del objetivo.	La actividad se elimina con éxito del objetivo.	Exitoso	La actividad se ha eliminado del objetivo, el porcentaje total del evento se actualizó y la lista de objetivos se actualizó.
8	Prueba Unitaria	Se hace click en las fechas de un actividad, y se modifica las fechas de inicio y fin.	Las fechas de la actividad se actualizan en el sistema.	Exitoso	Las fechas de la actividad se actualizaron en el sistema y la lista de objetivos se actualizó.

7.2.6. Informe de pruebas: Gestionar Obstáculos

Tabla 20 Informe de pruebas del CUS Gestionar Obstáculos

Informe de prueba					
Unidad de Prueba		PCUS_Gestionar_Obstáculos			
Fecha:	15/012019	Avance	100%	Tester	Kevin Narvaez
Descripción de la prueba:					
El objetivo de la prueba es verificar la funcionalidad de todas las acciones posibles al momento de gestionar obstáculos.					
#	Tipo	Descripción	Resultado esperado	Resultado	Detalles
1	Prueba Unitaria	Se hace click en la etiqueta Obstáculos y aparece la tarjeta de objetivos.	Aparece la tarjeta de obstáculos.	Exitoso	La tarjeta de obstáculos se mostró en la página con todos los objetivos y actividades registradas con los obstáculos que presentaron.
2	Prueba Unitaria	Se hace click en el botón de lista de obstáculos, se elige un obstáculo para la actividad, se indica la fecha de inicio y fin de obstáculo y se hace click en el botón de agregar.	El obstáculo se agrega a la actividad respectiva y se actualiza la tarjeta de obstáculos.	Exitoso	El obstáculo se añadió a la actividad exitosamente y la tarjeta de obstáculos se actualizó.
3	Prueba Unitaria	Se hace click en la casilla “total” para indicar que el obstáculo ha impedido la realización de la actividad a la que está asociado.	El obstáculo cambia de estado a total y la actividad a la que está asociado ya no puede ser “alcanzada”.	Exitoso	El obstáculo cambió al estado total y es imposible que la actividad sea “realizada”.
4	Prueba Unitaria	Se hace click en las fechas de un obstáculo, y se modifica las fechas de inicio y fin.	Las fechas del obstáculo se actualizan en el sistema.	Exitoso	Las fechas del obstáculo se actualizaron en el sistema y la tarjeta de obstáculos se

					actualizó.
5	Prueba Unitaria	Se hace click en el botón “X” al costado de un obstáculo para eliminarlo de la actividad.	El obstáculo se elimina de la actividad.	Exitoso	El obstáculo ha sido eliminado con éxito y se actualizó la tarjeta de obstáculos.

7.2.7. Informe de pruebas: Participar en un evento

Tabla 21 Informe de prueba del CUS Participar en Evento

Informe de prueba					
Unidad de Prueba		PCUS_Participar_en_Evento			
Fecha:	15/012019	Avance	100%	Tester	Kevin Narvaez
Descripción de la prueba:					
El objetivo de la prueba es verificar la funcionalidad de todas las acciones posibles a momento de participar en un evento aceptado mediante la página web.					
#	Tipo	Descripción	Resultado esperado	Resultado	Detalles
1	Prueba Unitaria	Se hace click en el nombre de un evento que, en la página donde están todos los eventos, en la sección próximos.	Aparece la información básica del evento seleccionado y un botón para participar en él.	Exitoso	Apareció la página con la información básica del evento, así como la información del lugar. También aparece un botón de “Participar”.
2	Prueba Unitaria	Se hace click en el botón de participar.	El sistema registra la persona como un participante del evento.	Exitoso	El sistema registró a la persona como un participante. Se ha recargado la página indicando al usuario que ya es un participante de dicho evento.

CAPÍTULO VIII

8.1. Viabilidad del Proyecto

Para que es proyecto se logre llevar a cabo es necesario determinar cuáles son los elementos indispensables para la implementación del mismo. Para lo que se ha descrito anteriormente se realiza la siguiente premisa: ¿cuáles son los elementos necesarios para que el sistema funcione correctamente?, ¿cuánto será el costo total (en términos económicos) que tendrá el sistema a implementar?, ¿Se encuentra permitido dentro de los aspectos legales?

A continuación, se realiza una propuesta de implementación de acuerdo a la viabilidad técnica, económica y legal.

8.1.1. Viabilidad Técnica

Desde un punto de vista técnico, la realización de este proyecto requiere de algunos recursos y requisitos técnicos. A continuación, se describe el cuadro de recursos necesarios para la implementación del sistema, el cual utilizará una plataforma Web.

Tabla 22 Viabilidad Técnica

Recurso		Características	Cantidad
Hardware	Laptop	Procesador: Intel Core i5 RAM: 8Gb Disco Duro: 500Gb Video: nVidia 1Gb Pantalla: 15.6” SO: Windows 10	1
Software	Gestor de Base de Datos: Mysql		1
	Servidor de Aplicaciones XAMP		
Dominio	Dominio y Hosting	GoDaddy	1
Muebles e inmuebles y otro medios de comunicación	Luz Internet		1

8.1.2. Viabilidad Económica

Para poder demostrar que el sistema es viable económicamente, a continuación, se detallan los recursos que se necesitan para la construcción e implementación del sistema y sus respectivos costos.

Análisis de Costos

Recursos Humanos

El presente proyecto de investigación ha requerido de una sola persona, la cual ha representado los distintos roles que se mencionan a continuación.

- **Analista de Requerimientos:** Es el encargado de identificar a los stakeholders, modelar el negocio e identificar los requerimientos funcionales y no funcionales que formarán parte del sistema
- **Analista de Sistemas:** Encargado de realizar el análisis del sistema que se está realizando y representar en forma gráfica los distintos componentes que formarán parte del sistema.
- **Arquitecto:** Se encarga de establecer la arquitectura sobre la que va a estar implementada el sistema.
- **Administrador de Base de Datos:** Se encarga de todo lo relacionado a la Base de Datos, desde su concepción hasta su mantenimiento y continua actualización y mejora.
- **Programador:** Realiza la implementación del sistema para que el usuario final pueda acceder a él de una manera fácil.
- **Tester:** Se encarga de las pruebas funcionales sobre el trabajo que ha presentado el Programador.
- **Documentador:** Está encargado de todos los documentos necesarios para un buen entendimiento del sistema.

Recursos de Hardware

- Laptop Intel Core i5 8GB 500Gb

Recursos de Software

El software utilizado en este proyecto es Open Source, por lo que su costo es cero.

- Enterprise Architect 13
- Gestor de Base de Datos Mysql
- Visual Studio Code
- Xampp
- Composer
- Bitbucket

Recursos de Producción

- Hosting

Muebles e inmuebles

- Internet
- Costo de Luz

Tiempo de implementación del proyecto

- El proyecto tiene una duración de 1 año para completar el alcance que se ha definido, que incluye el despliegue de la aplicación para que pueda ser utilizada por la organización.

Costo de los Recursos

- Seguidamente se presenta la tabla de costos donde se detalla todo lo mencionado anteriormente. Esta estimación se encuentra basada en la moneda nacional (Nuevo Sol).

Recursos	Egresos												Totales	
	Mes 1 En soles S/.	Mes 2 En soles S/.	Mes 3 En soles S/.	Mes 4 En soles S/.	Mes 5 En soles S/.	Mes 6 En soles S/.	Mes 7 En soles S/.	Mes 8 En soles S/.	Mes 9 En soles S/.	Mes 10 En soles S/.	Mes 11 En soles S/.	Mes 12 En soles S/.		En soles S/.
Recursos Humanos	Analista de Requerimientos	0.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	2,200.00
	Analista de Sistemas	0.00	0.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	2,000.00
	Arquitecto	0.00	0.00	0.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	1,800.00
	Administrador de Base de Datos	0.00	0.00	0.00	250.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	1,850.00
	Programador	0.00	200.00	200.00	200.00	200.00	250.00	250.00	250.00	250.00	250.00	250.00	250.00	2,550.00
	Tester	0.00	0.00	0.00	0.00	0.00	250.00	250.00	250.00	250.00	250.00	250.00	250.00	1,750.00
	Documentador	0.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	200.00	2,200.00
	Costo Total RRHH	0.00	600.00	800.00	1,250.00	1,200.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	1,500.00	14,350.00
Hardware	Laptop Intel Core i5 8GB 500Gb	1,500.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1,500.00
	Costo Total Hardware	1,500.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1,500.00
Software	Enterprise Architect Portable 13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Gestor de Base de Datos Mysql	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Visual Studio Code	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Composer	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	GitHub	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Costo Software	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Total	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Hosting	Hosting + Dominio Goddady	70.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	70.00
	Costo Total Hosting	70.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	70.00
Muebles e inmuebles y otros medios de comunicación	Costo Luz	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	60.00	720.00
	Internet	120.00	120.00	120.00	120.00	120.00	120.00	120.00	120.00	120.00	120.00	120.00	120.00	1,440.00
	Costo Total Comunicación	180.00	180.00	180.00	180.00	180.00	180.00	180.00	180.00	180.00	180.00	180.00	180.00	2,160.00
Costo Total de la implementación del sistema													16,080.00	

Figura 64 Tabla de costos

Fuente: Elaboración propia

Beneficios

Si bien el objetivo de este proyecto no es que la organización gane más dinero, se ha podido calcular un beneficio económico que puede ser usado para una mejora interna de la misma, es decir, un aumento en la calidad de sus recursos.

Se sabe que cada semana en las reuniones dominicales existe un tiempo de pasar a recoger los diezmos ofrendas que traen los asistentes, esta cantidad es utilizada para los gastos que la iglesia requiere, que son el costo del alquiler del local, la luz, el agua y solucionar algún problema eventual que sucede.

No se tiene un ingreso económico semanal definido e invariable debido a que todo depende de cuánto estén dispuestos a ofrendar los asistentes ya que esto sale de ellos mismos. Hay algunas ocasiones en que se cubre todo lo necesario con los diezmos y ofrendas, otras ocasiones en las que no se cubre, y así fluctúa a través del tiempo.

Las variables necesarias para medir los ingresos aproximados cada mes son tres:

- Cantidad de ofrendas (CO)
- Cantidad de diezmos (CD)
- Otros (OI)

La fórmula que calcularía las ganancias sería la siguiente:

$$CO + CD + OI$$

2019						
	Mes 0	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5
	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.
Asistentes que diezman	10	10	10	10	10	10
Asistentes que ofrendan	20	20	22	24	25	25
Cantidad de Diezmos	1800.00	1850.00	2400.00	2400.00	2500.00	2500.00
Cantidad de Ofrendas	200.00	200.00	300.00	250.00	300.00	300.00
Otros	600.00	430.00	450.00	450.00	450.00	600.00
Total de Ingresos	2600.00	2480.00	3150.00	3100.00	3250.00	3400.00

Figura 65 Cuadro de beneficios económicos

Fuente: Elaboración propia

Para el cálculo del flujo de caja se han tomado como ingresos los descritos anteriormente y como egresos los gastos que se tienen de los servicios que paga la iglesia, el sueldo del pastor y otros gastos menores (limpieza, comida, etc.)

2019																2020																		
	Mes 0	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12	Mes 13	Mes 14	Mes 15	Mes 16	Mes 0	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12	Mes 13	Mes 14	Mes 15	Mes 16
	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.	En soles S/.		
Ingresos																																		
Asistentes que diezman	10	10	10	10	10	10	14	14	14	14	16	16	16	16	28	28	28	10	10	10	10	10	10	14	14	14	14	16	16	16	16	28	28	28
Asistentes que ofrendan	20	20	22	24	25	25	25	25	25	25	25	25	25	25	25	30	30	30	20	20	22	24	25	25	25	25	25	25	25	25	25	30	30	30
Cantidad de Diezmos	1800.00	1850.00	2400.00	2400.00	2500.00	2500.00	2600.00	2600.00	2600.00	2600.00	2750.00	2750.00	2750.00	2750.00	2900.00	2900.00	2900.00	1800.00	1850.00	2400.00	2400.00	2500.00	2500.00	2600.00	2600.00	2600.00	2750.00	2750.00	2750.00	2750.00	2900.00	2900.00	2900.00	
Cantidad de Ofrendas	200.00	200.00	300.00	250.00	300.00	300.00	350.00	350.00	370.00	350.00	380.00	340.00	400.00	360.00	410.00	340.00	420.00	200.00	200.00	300.00	250.00	300.00	300.00	350.00	350.00	370.00	350.00	380.00	340.00	400.00	360.00	410.00	340.00	420.00
Otros	600.00	430.00	450.00	450.00	450.00	600.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	600.00	430.00	450.00	450.00	450.00	600.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00	450.00
Cantidad de días	30	31	30	31	30	31	31	28	31	30	31	30	31	31	30	31	30	30	31	30	31	30	31	31	28	31	30	31	30	31	30	31	30	
Total de Ingresos	2600.00	2480.00	3150.00	3100.00	3250.00	3400.00	3400.00	3400.00	3400.00	3400.00	3580.00	3540.00	3600.00	3560.00	3760.00	3690.00	3770.00	2600.00	2480.00	3150.00	3100.00	3250.00	3400.00	3400.00	3400.00	3400.00	3580.00	3540.00	3600.00	3560.00	3760.00	3690.00	3770.00	
Egresos																																		
Alquiler de local	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00	1300.00
Pastor	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00	1500.00
Servicio de Luz	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Servicio de agua	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00	50.00
Otros	70.00	50.00	30.00	70.00	100.00	300.00	80.00	70.00	35.00	70.00	50.00	30.00	45.00	60.00	70.00	40.00	45.00	70.00	50.00	30.00	70.00	100.00	300.00	80.00	70.00	35.00	70.00	50.00	30.00	45.00	60.00	70.00	40.00	45.00
Total de Egresos	3020.00	3000.00	2980.00	3020.00	3050.00	3250.00	3030.00	3020.00	2985.00	3020.00	3000.00	2980.00	2995.00	3010.00	3020.00	2990.00	2995.00	3020.00	3000.00	2980.00	3020.00	3050.00	3250.00	3030.00	3020.00	2985.00	3020.00	3000.00	2980.00	2995.00	3010.00	3020.00	2990.00	2995.00
Totales	420.00	520.00	170.00	80.00	200.00	150.00	370.00	380.00	435.00	380.00	580.00	560.00	605.00	550.00	740.00	700.00	775.00	420.00	520.00	170.00	80.00	200.00	150.00	370.00	380.00	435.00	380.00	580.00	560.00	605.00	550.00	740.00	700.00	775.00

Figura 66 Tabla de viabilidad económica

Fuente: Elaboración propia

De acuerdo al estudio de costo beneficio se observa que, en el sexto mes desde la implementación del sistema, existe un aumento de los ingresos en la organización.

Cálculo del VAN y TIR

Para calcular la viabilidad del proyecto se hacen necesarios la evaluación de dos parámetros financieros muy utilizados en los proyectos:

Valor Actual Neto (VAN): Es la actualización de los valores futuros (flujos de caja) a la fecha actual. Si a este valor le descontamos la inversión inicial, tenemos el valor neto de nuestro proyecto.

Tasa Interna de Retorno (TIR): Es la tasa de descuento de un proyecto que permite que un proyecto sea rentable.

Factor simple de actualización (FSA): Es la actualización de valores futuros en valores actuales.

Mes	Flujo Caja Neto	VAN
	En soles S/.	En soles S/.
0	-420.00	-420.00
1	-520.00	-892.73
2	170.00	-752.23
3	80.00	-692.13
4	200.00	-555.52
5	150.00	-462.39
6	370.00	-253.53
7	380.00	-58.53
8	435.00	144.40
9	380.00	305.56
10	580.00	529.17
11	560.00	725.45
12	605.00	918.22
13	550.00	1077.54
14	740.00	1272.40
15	700.00	1439.98
16	775.00	1608.64

TIR	25%
-----	-----

Figura 67 Tabla de VAN y TIR

Fuente: Elaboración propia

8.1.3. Viabilidad Legal

- **Capítulo I, Artículo 2º, numeral 5 de la Constitución Política del Perú**

Este artículo refiere el derecho de las personas hacia la información en ello expresa lo siguiente, “Al solicitar sin expresión de causa la información que requiera y a recibirla de cualquier entidad pública, en el plazo legal, con el costo que suponga el pedido”. Es decir que todo peruano tiene derecho a la información, la cual es brindada por las entidades públicas, estableciendo el costo de dicha información.

La tesis toma como principio este derecho debido a que la información mostrada en la solución Web es estrictamente pública en todos sus aspectos.

- **Capítulo I, Artículo 2º, numeral 6 de la Constitución Política del Perú**

La constitución política expresa mediante el artículo 2, numeral 6 lo siguiente “A que los servicios informáticos, computarizados o no, públicos o privados, no suministren informaciones que afecten la intimidad personal y familiar”.

Este capítulo extrae que los servicios informáticos no deben afectar la intimidad personal.

CAPÍTULO IX: GESTIÓN DEL PROYECTO

9.1. Organización del Proyecto

9.1.1. Organigrama del proyecto

El proyecto ha sido desarrollado bajo el siguiente esquema jerárquico:

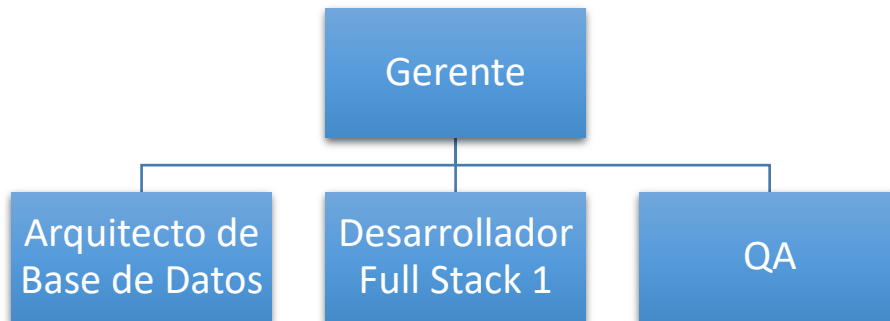


Figura 68 Organigrama del proyecto

Fuente: Elaboración propia

- **Gerente:** Encargado de la administración de recursos físicos, humanos y encargado de los conocimientos de cada una de las tecnologías a utilizar.
- **Arquitecto de Base de Datos:** Encargado de realizar y mantener la Base de Datos, de modo que toda la información esté estructurada de la mejor forma.
- **Desarrollador Full Stack:** Encargados del front end y backend del sistema. También se encargan de realizar la implementación del mismo.
- **QA:** Encargado de las pruebas unitarias, las pruebas de integración y la validación de las características del sistema, de modo que cumplan los requerimientos no funcionales de usabilidad y calidad.

9.1.2. Estructura de Desglose del proyecto (EDT)

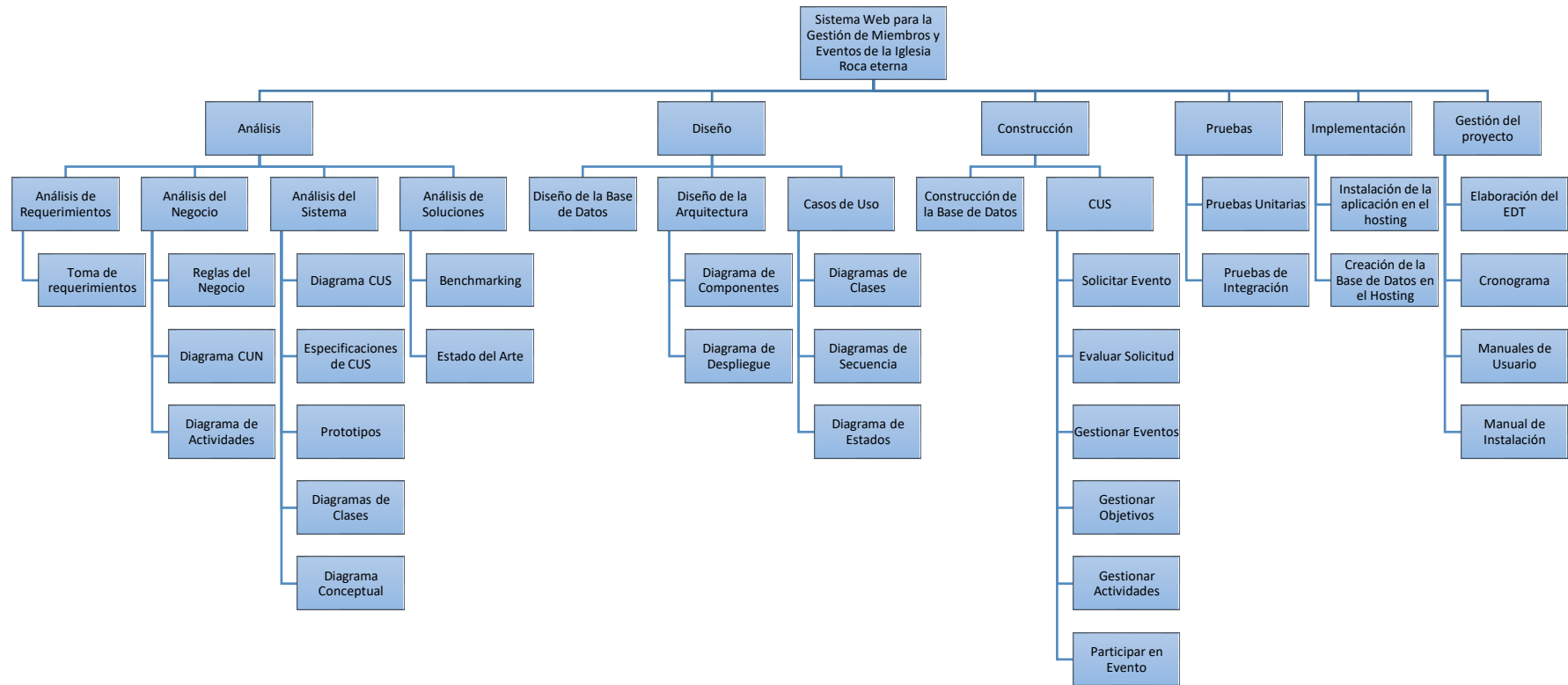


Figura 69 EDT del proyecto

Fuente: Elaboración propia

9.2. Estimación y ejecución del proyecto

9.2.1. Cronograma de ejecución del proyecto

El Diagrama de Gantt generado al utilizar el software MS Project muestra el siguiente resumen. De lo que se observa la fecha de inicio el lunes 07 de mayo del 2018 y terminó el viernes 3 de mayo del 2019.

Información del proyecto 'Cronograma_Rumbo_Ágil'

Fecha de comienzo: jue 30/05/19 Fecha actual: dom 7/02/21

Fecha de fin: vie 3/05/19 Fecha de estado: NOD

Programar a partir de: Fecha de comienzo del proyecto Calendario: Estándar

Todas las tareas comienzan lo antes posible. Prioridad: 500

Campos personalizados de empresa

Departamento:

Nombre de campo personalizado	Valor
-------------------------------	-------

Ayuda Estadísticas... Aceptar Cancelar

Figura 70 Resumen del cronograma del proyecto

Fuente: Elaboración propia

9.2.2. Cronograma detallado

1	▲ Sistema Web para la Gestión de miembros y Eventos de la Iglesia Roca Eterna	260 días	lun 7/05/18	vie 3/05/19
1.1	▲ Análisis	40 días	vie 25/05/18	jue 19/07/18
1.1.1	Análisis del negocio	11 días	vie 25/05/18	vie 8/06/18
1.1.2	Análisis de requerimientos	10 días	lun 11/06/18	vie 22/06/18
1.1.3	Análisis del sistema	15 días	lun 25/06/18	vie 13/07/18
1.1.4	Análisis de soluciones	4 días	lun 16/07/18	jue 19/07/18
1.2	▲ Diseño	26 días	vie 20/07/18	vie 24/08/18
1.2.1	Diseño de la Base de Datos	11 días	vie 20/07/18	vie 3/08/18
1.2.2	Diseño de la arquitectura	5 días	lun 6/08/18	vie 10/08/18
1.2.3	Casos de Uso	10 días	lun 13/08/18	vie 24/08/18
1.3	▲ Construcción	165 días	lun 27/08/18	vie 12/04/19
1.3.1	Base de Datos	2 días	lun 27/08/18	mar 28/08/18
1.3.2	Programación de CUS	163 días	mié 29/08/18	vie 12/04/19
1.4	▲ Pruebas	10 días	lun 15/04/19	vie 26/04/19
1.4.1	Pruebas unitarias	5 días	lun 15/04/19	vie 19/04/19
1.4.2	Pruebas de Integración	5 días	lun 22/04/19	vie 26/04/19
1.5	▲ Implementación	2 días	mar 30/04/19	mié 1/05/19
1.5.1	Instalación del aplicación	1 día	mar 30/04/19	mar 30/04/19
1.5.2	Instalación de la Base de Datos	1 día	mié 1/05/19	mié 1/05/19
1.6	▲ Gestión del proyecto	2 días	jue 2/05/19	vie 3/05/19
1.6.1	Manual de usuario	1 día	jue 2/05/19	jue 2/05/19
1.6.2	Manual de instalación	1 día	vie 3/05/19	vie 3/05/19

Figura 71 Cronograma detallado del proyecto

Fuente: Elaboración propia

CONCLUSIONES

- 1) El sistema ha permitido que los datos de la congregación puedan ser clasificados, ordenados y representados de tal forma que haya un crecimiento en la organización y el conocimiento de la iglesia entre sus miembros.
- 2) El módulo de planificación de eventos ha servido para que no quede información al aire desde el momento de la concepción de la solicitud del mismo, pasando por su desarrollo y terminando con la culminación del evento. Esto ayudó a los líderes a identificar los puntos débiles y fuertes de cada tipo de evento.
- 3) La mejora de la organización de la iglesia, gracias al sistema brindado, ha logrado que los miembros de la congregación se comprometan más, aumentando así el sustento económico y calidad de los procesos que se desarrollan dentro de la misma.
- 4) Realizar funcionalidades asociadas con ministerios, seguimiento de miembros y creación de reportes mucho más específicos, son de utilidad para el crecimiento de la iglesia a un nivel organizacional, y logra que el compromiso de los miembros avance al siguiente nivel.
- 5) Gracias a la implementación de este sistema, se logró aumentar la calidad de los eventos que se planifican dentro de la iglesia un 30%; se optimizó en su totalidad el tiempo y esfuerzo que se invertía para generar los reportes en un 80% y se pudo agilizar el tiempo de registro de los distintos eventos y su organización en un 80%.

RECOMENDACIONES

- 1) Hemos notado que el uso de notificaciones en tiempo real como los mensajes de texto, los correos electrónicos e incluso las mismas notificaciones en tiempo real pueden hacer que el conocimiento entre miembros-miembros y miembros-líderes sea más amigable y mucho más efectiva. Por lo que el desarrollo próximo de una aplicación móvil es indispensable para lograr dicho objetivo.
- 2) Aplicar los conocimientos de UX (User Experience) al momento de construir las interfaces de usuario, son útiles para que el sistema sea mucho más amigable e intuitivo a la hora de realizar todas las acciones que permite el sistema.
- 3) Un estudio del negocio más profundo ayuda al planteamiento de muchas más opciones de solución y aumenta la probabilidad de que estas soluciones sean óptimas y puedan abarcar tanto los problemas actuales como aquellos que los usuarios podrían encontrar en un futuro.

BIBLIOGRAFÍA

- (Colombia), U. d. (5 de Septiembre de 2011). *SlideShare*. Obtenido de <https://es.slideshare.net/mstabare/arquitecturas-de-software-parte-1>
- Alegsa, L. (2017). *alegsa*. Obtenido de http://www.alegsa.com.ar/Dic/aplicacion_web.php
- Almarza, D. K.-J. (1984). *Jornadas Hispano-Alemanas sobre la personalidad y la obra de Martín Lutero en el V centenario de su nacimiento*. Salamanca, España: San steban.
- Álvarez, M. Á. (2001). *Desarrollo Web*. Obtenido de <https://www.desarrolloweb.com/articulos/392.php>
- Anillo Farelo, J. S., & Pájaro Martínez, E. E. (07 de 2014). Obtenido de http://190.242.62.234:8080/jspui/bitstream/11227/737/1/Tesis_Anillo_Pajaro.pdf
- Anzilutti, D. B. (s.f.). *Ineventos*. Obtenido de Ineventos: <https://www.ineventos.com/pe/blog/que-es-un-evento.aspx>
- Aprende a Programar*. (s.f.). Obtenido de Aprende a Programar: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-ipara-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70&Itemid=193
- Aranibar, N. (2011). *monografias*. Obtenido de <http://www.monografias.com/trabajos88/mysql-worckbench/mysql-worckbench.shtml>
- Blogolengua. (s.f.). *Blogolengua*. Obtenido de Blogolengua: <http://www.blogolengua.com/2009/04/parafernalialia.html>
- Burkholder, J. (14 de 04 de 2018). *tgccoalición*. Obtenido de <https://www.coalicionporelevangelio.org/articulo/crece-la-iglesia/>
- Carlos, J. (29 de 09 de 2016). *Merca.20*. Obtenido de Merca.20: <https://www.merca20.com/la-organizacion-eventos-corporativos-evolucion/>

Clavo, O. (02 de 02 de 2013). *Osmer Clavo*. Obtenido de Osmer Clavo:
<http://osmerclavo.blogspot.com/2013/02/beneficios-tangibles-e-intangibles.html>

definición.de. (s.f.). *Definición.De*. Obtenido de Definición.de:
<https://definicion.de/evento/>

Documentación de Vue. (2018). Obtenido de Vue.js: <https://es-vuejs.github.io/vuejs.org/v2/guide/>

Ecu.red. (s.f.). Obtenido de Ecu.red: <https://www.ecured.cu/Eventos>

Escobar, M. (2011). *Matar a Lutero*. Tennessee, Estados Unidos: Graciella Lelli.

García, J. M. (11 de 12 de 2015). *arsys*. Obtenido de
<https://www.arsys.es/blog/programacion/que-es-laravel/>

Hoy, I. (2017). *Iglesia Hoy*. Obtenido de <http://www.iglesiahoy.com>

IBM. (s.f.). Obtenido de IBM:
https://www.ibm.com/support/knowledgecenter/es/SSFPJS_7.5.1/com.ibm.wbpm.wid.bpel.doc/topics/cunder.html

IBM. (s.f.). Obtenido de IBM:
<https://www.ibm.com/support/knowledgecenter/es/SSV2LR/com.ibm.wbpm.wid.bpel.doc/busrules/topics/cundbus.html>

Iglesia, H. d. (2017). *Hablemos de Iglesia*. Obtenido de
<https://app.hablemosdeiglesia.com/>

LucidChart. (s.f.). Obtenido de LucidChart: <https://www.lucidchart.com/pages/es/qué-es-el-lenguaje-unificado-de-modelado-uml>

Marco, V. D. (15 de Noviembre de 2015). *VictorDíaz*. Obtenido de
<https://victordiaz.me/websocket>

Metodoss. (s.f.). Obtenido de Metodoss: <https://metodoss.com/metodologia-rup/>

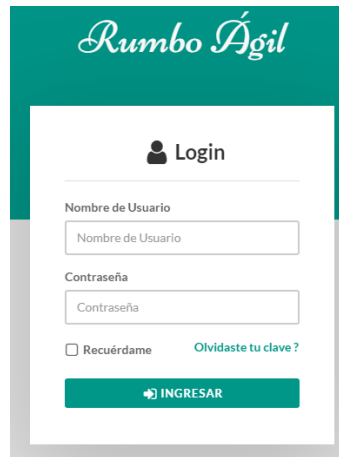
Moya, R. (s.f.). *modelo 4+1 vistas de kruchten*. Obtenido de
<https://jarroba.com/modelo-41-vistas-de-kruchten-para-dummies/>

- Moyano Mejía, G. W. (2015). Obtenido de <http://repositorio.puce.edu.ec/bitstream/handle/22000/8105/Tesis-GWMM-Entrega-vFinal.pdf?sequence=1&isAllowed=y>
- O'Neil, J. (1991). *Martín Lutero*. Madrid, España: Grafiris. S.A.
- paraquesirven.com*. (s.f.). Obtenido de paraquesirven.com: <http://paraquesirven.com/para-que-sirve-google-drive/>
- Rambla informática*. (2014). Obtenido de Rambla Informática: <http://www.ramblainf.com/enterprise-architect.html>
- Reicek. (2015). *Platzi*. Obtenido de Platzi: <https://platzi.com/blog/laravel-framework-php/>
- Softlife. (2017). *Softlife*. Obtenido de <http://www.misoftlife.com/>
- Warren, R. (1995). *Una iglesia con propósito*. Estados Unidos: Zondervan.
- WikiPedia. (s.f.). *Iglesia Presbiteriana ortodoxa*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Iglesia_presbiteriana_ortodoxa
- Yeeply. (26 de 01 de 2018). *Yeeply*. Obtenido de Yeeply: <https://www.yeeply.com/blog/6-tipos-desarrollo-de-aplicaciones-web/>

ANEXO 1: Manual de Usuario - Líder de Ministerio

1. Iniciar sesión

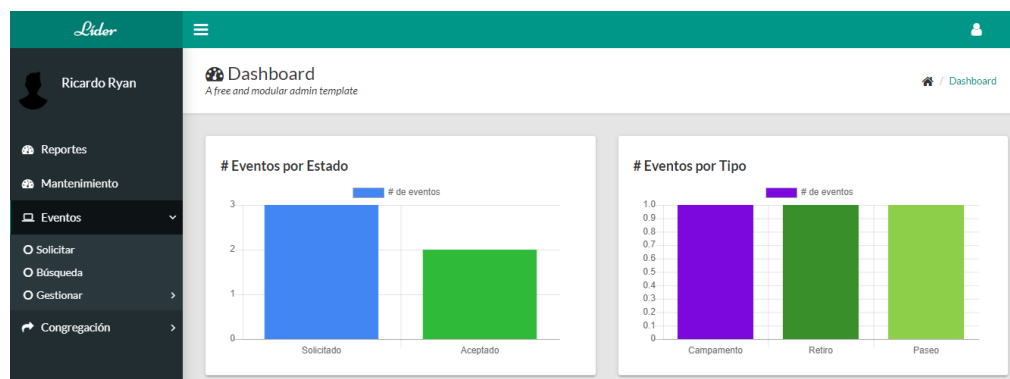
Ingresa el nombre de usuario y la contraseña correspondiente para poder ingresar al sistema, luego hace click en “Ingresar”. Si hay algún error en los datos, se mostrará el error y pedirá ingresar los datos nuevamente.



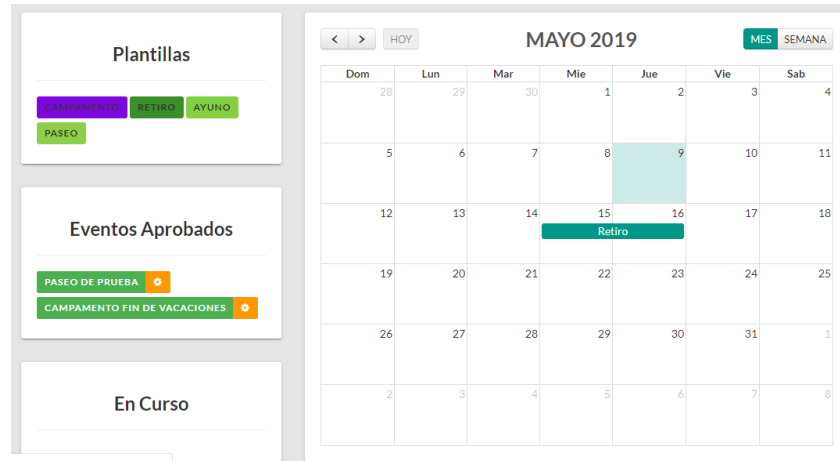
The login form is titled "Rumbo Ágil" and "Login". It contains two input fields: "Nombre de Usuario" and "Contraseña". Below the fields are two links: "Recuérdame" (with a checkbox) and "Olvidaste tu clave?". At the bottom is a green button labeled "INGRESAR".

2. Solicitar evento

Se muestra una pantalla con reportes y las opciones correspondientes para el rol de líder del ministerio. Deberá seleccionar la opción “Solicitar” dentro del menú “Eventos”.



En la página de “Solicitar”, arrastre una de las plantillas hasta la fecha en el calendario en la q solicitarás el evento. Aparecerá la plantilla en el calendario como se muestra en la imagen inferior. Una vez ahí, haga click en dicha plantilla.



Aparecerá este modal en la pantalla. Llene los datos correspondientes al evento. Si hay algún error en los datos ingresados, se le mostrará el mensaje del error o errores hasta que los corrija. En caso de que todo esté correcto, el evento se registrará como “solicitado”

Retiro

Nombre del Evento* Retiro

Propósito* Adquirir nuevos conocimientos

Descripción* Descripción del retiro

Imagen

Dirigido a* General

Lugar* La Huaca

Día de Inicio* 15/05/2019

Día de Fin* 16/05/2019

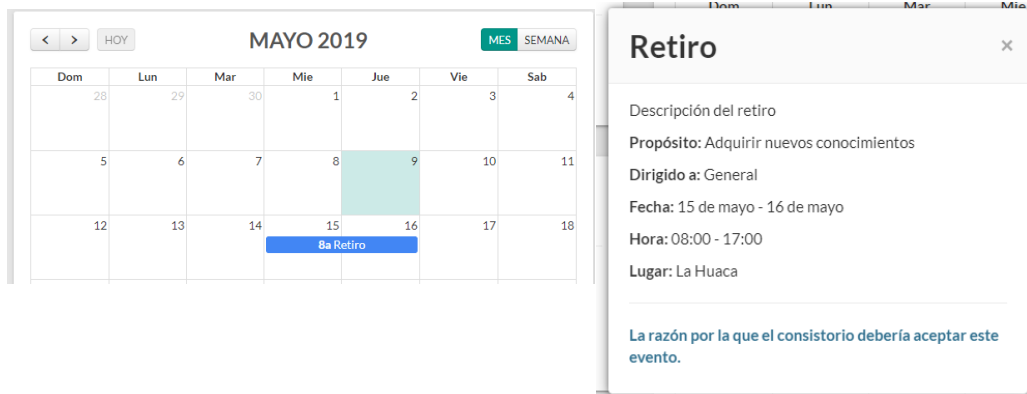
Hora de Inicio 8:00 a.m.

Hora de Fin 5:00 p.m.

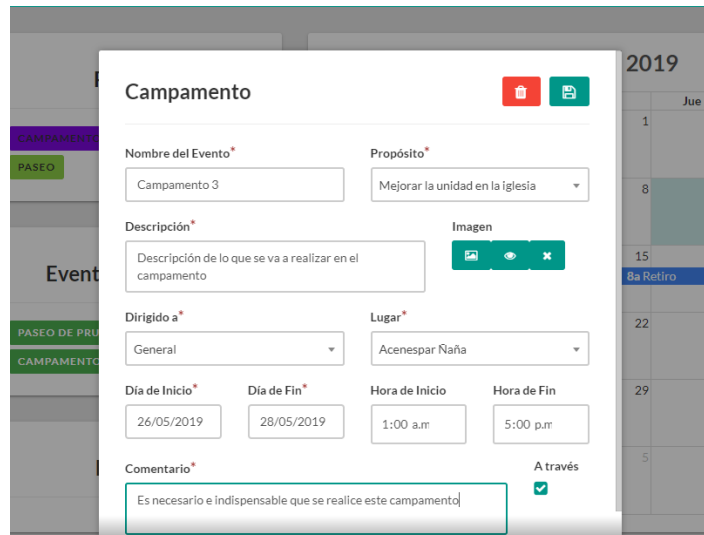
Comentario* La razón por la que el consistorio debería aceptar este evento.

A través

Aparecerá el evento solicitado en el calendario como se muestra en la siguiente imagen. Si hace click en dicho evento, aparece un popup con la información de este.



En caso de que desee registrar el evento sin pasar por la evaluación, al momento de registrar deberá seleccionar la casilla “a través”. Como sigue



El evento aparecerá en el calendario de la siguiente manera.

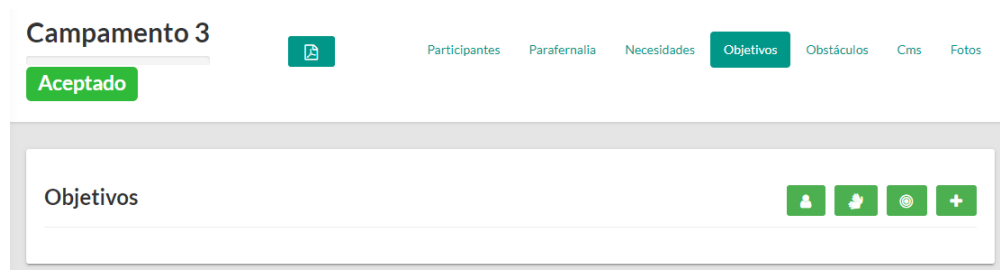
5	6	7	8	9	10	11
12	13	14	8a Retiro		17	18
19	20	21	22	23	24	25
11a Campamento 3			29	30	31	1

3. Añadir objetivos y actividades

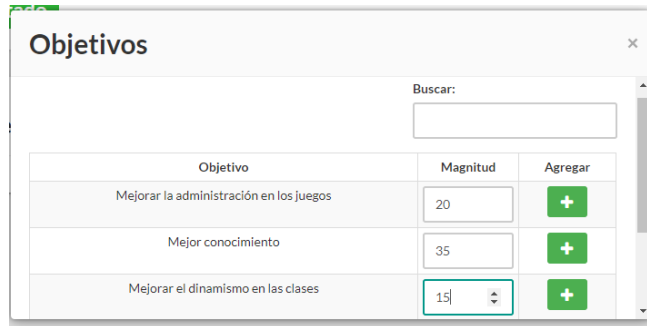
En la página de “Solicitar”, en la tarjeta “Eventos Aprobados” a la izquierda del calendario, haga click en el botón con el ícono del engranaje para ir directamente a la página de planificación. O en su defecto haga click en el evento dentro del calendario y luego en el popup de información, haga click en el botón con el engranaje.



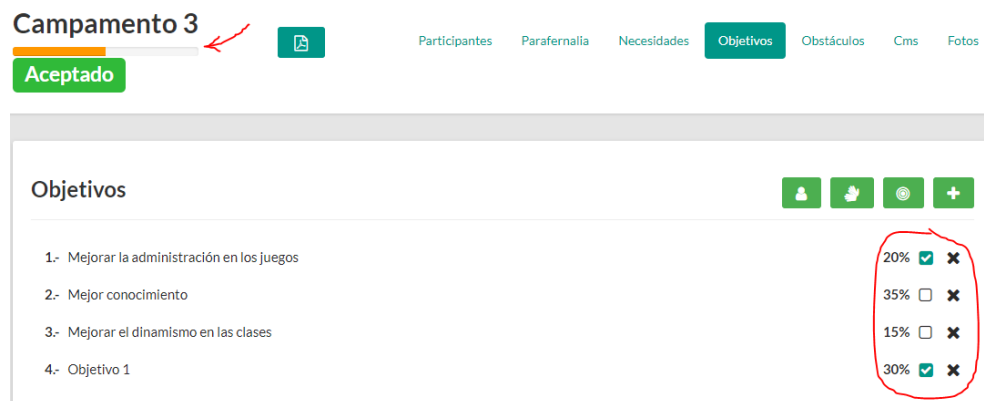
Aparecerá la página de planificación del evento. En la parte superior hay una serie de etiquetas que activan tarjetas. Seleccione la etiqueta “Objetivos” y aparecerá la tarjeta de objetivos.



Haga click en el botón con el ícono de una diana (tiro al blanco). Aparecerá un popup donde deberá elegir el objetivo que desea añadir, colocarle una magnitud y luego hacer click en el botón con el ícono de un “+”. Puede filtrar por nombre los objetivos si utiliza el campo de búsqueda. Podrá agregar cuantos objetivos desee, siempre y cuando la suma de sus magnitudes no pase las 100 unidades.




Puede eliminar un objetivo añadido haciendo click en la “X” q se muestra en la parte derecha. Puede “realizar” un objetivo haciendo click en la casilla al lado de la magnitud del mismo. El porcentaje del evento irá aumentando o disminuyendo de acuerdo a cuantos objetivos hayan sido alcanzados o no.



Para añadir actividades a un objetivo, haga click en el botón con el ícono de un par de manos. En el popup que aparecerá elija la actividad que añadirá al objetivo, la fecha en que inicia y culmina, y luego haga click en el botón “+”



Puede eliminar una actividad añadida haciendo click en la “X” que se muestra en la parte derecha. Puede “realizar” una actividad haciendo click en la casilla al lado de la fecha de la misma. El porcentaje del evento irá aumentando o disminuyendo de acuerdo a cuántas actividades y objetivos hayan sido alcanzados o no.

Campamento 3  [Participantes](#) [Parafernalia](#) [Necesidades](#) **Objetivos** [Obstáculos](#) [Cms](#) [Fotos](#)


Aceptado

Objetivos

1.- Mejorar la administración en los juegos	Domingo 26 - Lunes 27	20%	X
» 10.- Actividad 1	Domingo 26	<input checked="" type="checkbox"/>	X
» 12.- Actividad 3	Domingo 26 - Lunes 27	<input type="checkbox"/>	X
» 24.- Actividad 4	Lunes 27	<input checked="" type="checkbox"/>	X
2.- Mejor conocimiento	Domingo 26 - Martes 28	35%	X
» 11.- Actividad 2	Domingo 26	<input type="checkbox"/>	X
» 26.- Actividad 6	Domingo 26 - Martes 28	<input checked="" type="checkbox"/>	X
» 25.- Actividad 5	Martes 28	<input type="checkbox"/>	X
3.- Mejorar el dinamismo en las clases	Domingo 26	15%	X
» 28.- Actividad 8	Domingo 26	<input checked="" type="checkbox"/>	X
4.- Objetivo 1		30%	<input checked="" type="checkbox"/> X

4. Añadir obstáculos

Si selecciona la etiqueta “Obstáculos” en la parte superior de la página, podrá ver la tarjeta de obstáculos. Está tarjeta se actualiza automáticamente al realizar acciones con los objetivos y actividades. Solo se pueden añadir obstáculos a las actividades, no a los objetivos en sí.

Campamento 3  [Participantes](#) [Parafernalia](#) [Necesidades](#) [Objetivos](#) **Obstáculos** [Cms](#) [Fotos](#)

Aceptado

Obstáculos

1.- Mejorar la administración en los juegos	No hubo obstáculo
» 10.- Actividad 1	No hubo obstáculo
» 12.- Actividad 3	No hubo obstáculo
» 24.- Actividad 4	No hubo obstáculo
2.- Mejor conocimiento	No hubo obstáculo
» 11.- Actividad 2	No hubo obstáculo
» 26.- Actividad 6	No hubo obstáculo
» 25.- Actividad 5	No hubo obstáculo
3.- Mejorar el dinamismo en las clases	No hubo obstáculo
» 28.- Actividad 8	No hubo obstáculo
4.- Objetivo 1	

Para añadir un obstáculo a alguna actividad, haga click en el botón con un rostro triste y en el popup que aparecerá llene los campos. Selecciona el obstáculo que quiera añadir a alguna actividad, la fecha de inicio y fin del obstáculo y finalmente haga click en el botón “+”

Actividad	Obstáculo	Inicio	Demora	Agregar
<u>Actividad 1</u>	Falta de organización	26/05/2019	27/05/2019	<input checked="" type="checkbox"/>
<u>Actividad 3</u>	Falta de tiempo	26/05/2019	28/05/2019	<input checked="" type="checkbox"/>
<u>Actividad 4</u>	Falta de organización	27/05/2019	27/05/2019	<input checked="" type="checkbox"/>
Actividad 2	Falta de organización	26/05/2019	26/05/2019	<input checked="" type="checkbox"/>
<u>Actividad 6</u>	Falta de organización	28/05/2019	28/05/2019	<input checked="" type="checkbox"/>

En la lista de obstáculos se distinguirán 4 tipos de registros con 4 colores: color verde, no hubo obstáculo; color azul, existe un obstáculo durante la actividad; color mostaza, hubo un retraso en la actividad; rojo, la actividad no pudo llevarse a cabo. Se pueden eliminar obstáculos de las actividades si hace click en la “x” y puede indicar que la actividad no pudo realizarse si selecciona la casilla “total” al lado de la fecha del obstáculo. Si selecciona la casilla de una actividad que ha sido “realizada”, esta casilla se desactivará automáticamente y el porcentaje disminuirá.

Objetivos	Progreso	Acciones
1.- Mejorar la administración en los juegos Domingo 26 - Lunes 27	20%	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
» 10- Actividad 1	Domingo 26	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
» 12- Actividad 3	Domingo 26 - Lunes 27	<input type="checkbox"/> <input checked="" type="checkbox"/>
» 24- Actividad 4	Lunes 27	<input type="checkbox"/> <input checked="" type="checkbox"/>
2.- Mejor conocimiento Domingo 26 - Martes 28	35%	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
» 11- Actividad 2	Domingo 26	<input type="checkbox"/> <input checked="" type="checkbox"/>
» 26- Actividad 6	Domingo 26 - Martes 28	<input type="checkbox"/> <input checked="" type="checkbox"/>
» 25- Actividad 5	Martes 28	<input type="checkbox"/> <input checked="" type="checkbox"/>
3.- Mejorar el dinamismo en las clases Domingo 26	15%	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
» 28- Actividad 8	Domingo 26	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
4.- Objetivo 1	30%	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

Obstáculos ⊕

<p>1.- Mejorar la administración en los juegos</p> <p>» 10.- <i>Actividad 1</i></p> <p>» 12.- <i>Actividad 3</i></p> <p>» 24.- <i>Actividad 4</i></p>	<p>Falta de organización Domingo 26 - Lunes 27 <input type="checkbox"/> ✕</p> <p>Falta de tiempo Domingo 26 - Martes 28 <input type="checkbox"/> ✕</p> <p>Falta de organización Lunes 27 <input type="checkbox"/> ✕</p>
<p>2.- Mejor conocimiento</p> <p>» 11.- <i>Actividad 2</i></p> <p>» 26.- <i>Actividad 6</i></p> <p>» 25.- <i>Actividad 5</i></p>	<p>No hubo obstáculo</p> <p>Falta de organización Martes 28 <input checked="" type="checkbox"/> ✕</p> <p>No hubo obstáculo</p>
<p>3.- Mejorar el dinamismo en las clases</p> <p>» 28.- <i>Actividad 8</i></p>	<p>Falta de tiempo Domingo 26 - Martes 28 <input type="checkbox"/> ✕</p>
<p>4.- Objetivo 1</p>	

5. Cerrar sesión

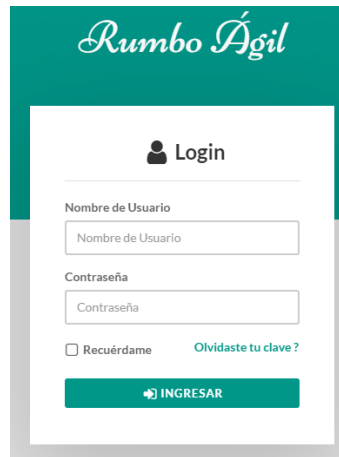
Al finalizar las operaciones, en la esquina superior derecha de la pantalla, encontrará el ícono que se muestra en la siguiente figura. Haga click en él y luego haga click en cerrar sesión.



ANEXO 2: Manual de Usuario - Consistorio

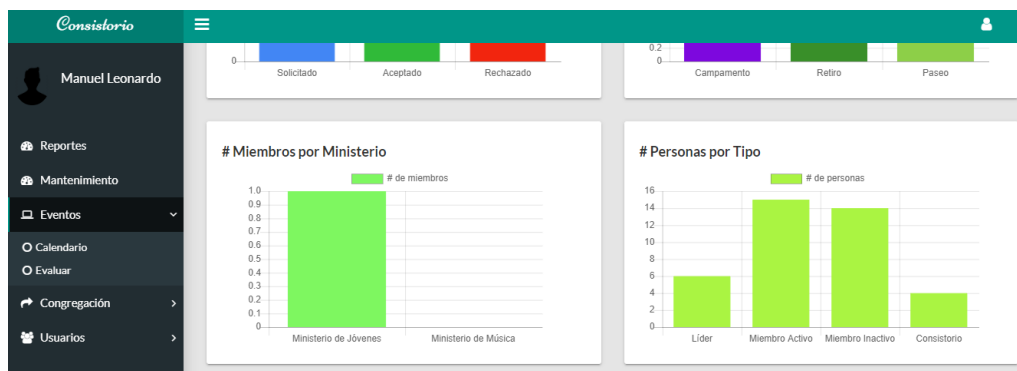
1. Iniciar sesión

Ingresa el nombre de usuario y la contraseña correspondiente para poder ingresar al sistema, luego hace click en “Ingresar”. Si hay algún error en los datos, se mostrará el error y pedirá ingresar los datos nuevamente.

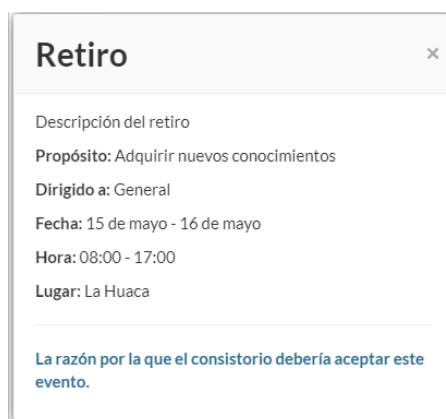
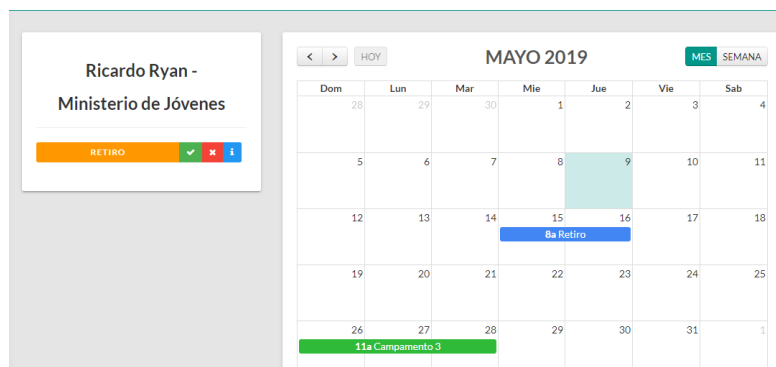


2. Evaluar evento

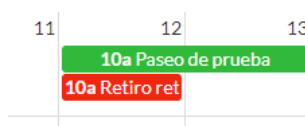
Se muestra una pantalla con los reportes de la iglesia. Debe hacer click en la opción “Evaluar”, dentro del menú “Eventos”



En la página aparecerá un calendario con los eventos aprobados, rechazados y los cancelados hasta el momento. Al lado izquierdo se muestra una tarjeta de acceso rápido con el nombre del líder que solicitó el evento y el ministerio que lidera como título, seguido de la lista de eventos por evaluar. Si hace click en el botón con una “i” aparecerá un popup con información del evento, así como el mensaje del líder que lo solicitó.



Si desea aceptar o rechazar rápidamente un evento. Solo haga click en el botón con un “check” para aprobar o en el que tiene una “x” para rechazar. Esto provocará que el evento cambie de estado automáticamente y se muestre en el calendario.



Si desea enviar un mensaje de la razón de la aprobación o rechazo del evento solicitado, haga click en evento que se encuentra dentro del calendario. En el modal, llene el campo de comentario y haga click en el “check” para aprobar o en la “x” para rechazar. Luego el evento se mostrará en el calendario.

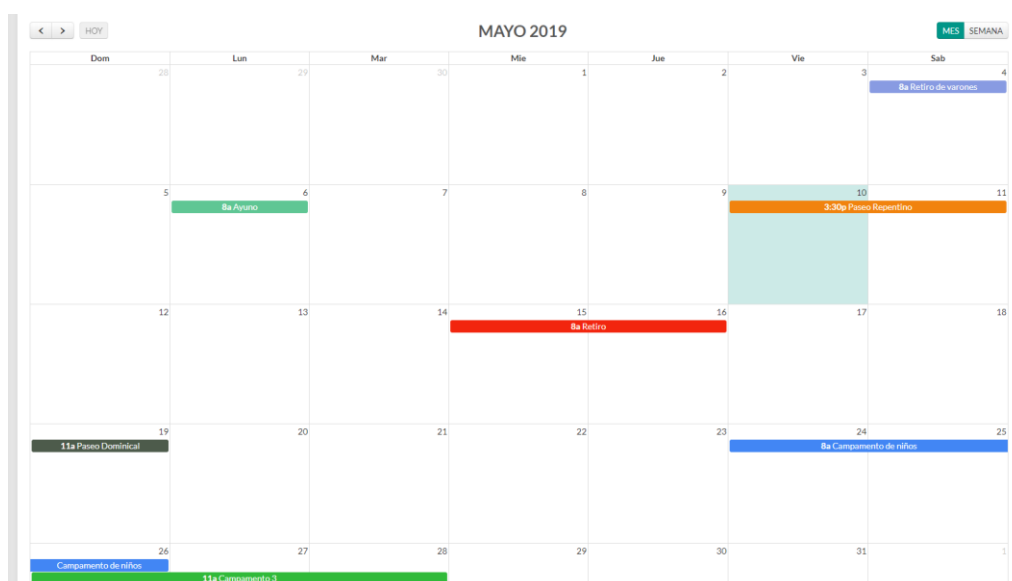
Retiro ✖ ✔

Comentario*

La razón por la que el consistorio debería aceptar este evento.

3. Gestionar Eventos

Debe hacer click en la opción “Calendario” dentro del menú “Eventos”. En la página aparecerá un calendario con todos los eventos hasta el momento (solicitados - azul, aceptados - verde, rechazados - rojo, cancelados - plomo, en curso - amarillo, finalizados – verde agua, cerrados - lila).



Existen distintas opciones que puede realizar con los distintos eventos, todo depende del estado de dichos eventos. Si el estado es “Aceptado” al hacer click en el, se abrirá un modal en el cual podrá modificar los datos necesarios y guardar los nuevos datos.

Campamento 3

Nombre del Evento*

Propósito*

Descripción*

Imagen

Dirigido a*

Lugar*

Día de Inicio*

Día de Fin*

Hora de Inicio

Hora de Fin

Si el estado es “Solicitado” o “Rechazado” al hacer click se muestra un popup con la información básica del evento.

Retiro

Descripción del retiro

Propósito: Adquirir nuevos conocimientos

Dirigido a: General

Fecha: 15 de mayo - 16 de mayo

Hora: 08:00 - 17:00

Lugar: La Huaca

Rechazado

Si el estado es “En curso”, “Cancelado”, “Finalizado” o “Cerrado”, al hacer click se abrirá en una nueva pestaña, la página de planificación de ese evento.

Paseo Repentino

En Curso

Participantes Parafernalia Necesidades Objetivos Obstáculos Cms Fotos

Participantes

Nombre Completo	Edad	Tipo de Persona	Rol	Opciones
Ricardo Ryan	24	Lider	Encargado de los juegos	
Manuel Leonardo	25	Lider	Encargado de Música	
Julissa Fabiola	24	Miembro Inactivo	Sin Asignar	
Beatriz	61	Lider	Encargada de Niños	
Graciela Rene	94	Miembro Inactivo	Sin Asignar	

4. Cerrar sesión

Al finalizar las operaciones, en la esquina superior derecha de la pantalla, encontrará el ícono que se muestra en la siguiente figura. Haga click en él y luego haga click en cerrar sesión.



ANEXO 3: Manual de Usuario - Miembro

1. Iniciar sesión

Ingrese a la página web de la iglesia. En la parte inferior hay una lista de opciones, haga click en la que dice “Iniciar Sesión”. Ingrese sus datos y haga click en ingresar.



The image shows a login form titled "Ingrese Sus Datos" overlaid on a website background. The form has two input fields: "Usuario" and "Contraseña:". Below the fields is a blue button labeled "INGRESAR". At the bottom of the form, there is a link that says "Olvidaste tu contraseña?". The background of the website shows a hand reaching out, and the navigation menu at the top includes "HOME", "ABOUT", "MESSAGES", "EVENTOS", and "INICIAR SESIÓN".

Su nombre aparecerá en el lugar de “Iniciar Sesión” lo que indica que todo está correcto.



2. Participar en un evento

Haga click en “Eventos” en la lista de opciones. Podrá ver una página con los eventos que se han desarrollado anteriormente y los eventos que aún están por llegar. Haga click en la opción de Próximos y luego en nombre del evento al que desea participar.

Últimos Próximos

		
Campamento 3 MAY. 26, 2019 - MINISTERIO DE JÓVENES	Ayuno del mes MAY. 25, 2019 - MINISTERIO DE JÓVENES	Retiro de Líderes MAY. 18, 2019 - MINISTERIO DE JÓVENES
Descripción de lo que se va a realizar en el campamento	Será el ayuno del mes.	Este retiro si está programado para que sea el retiro

El detalle del evento se mostrará en la página. Para ver detalles del lugar, solo haga click en el nombre del mismo. A continuación, haga click en el botón verde que se encuentra debajo.



Campamento 3

Descripción de lo que se va a realizar en el campamento

Dirigo a: General
Inicia: 26 de mayo - 11:00
Termina: 28 de mayo - 17:00

Lugar: Acenespar Ñaña
Club campestre
Dirección: Ñaña
Servicios: Piscina
Teléfonos: 980617583

[ÚNETE A LA AVENTURA](#)

Se recargará la página indicando que usted ya es un participante.

Campamento 3

Descripción de lo que se va a realizar en el campamento

Dirigo a: General

Inicia: 26 de mayo - 11:00

Termina: 28 de mayo - 17:00

Lugar: Acenespar Ñaña

YA ERES UN PARTICIPANTE!

3. Cerrar Sesión

Haga click en la opción que lleva su nombre:

HOME ABOUT MESSAGES ▾ EVENTOS RICARDO RYAN 

ANEXO 4: Manual de Instalación

1. Introducción

La documentación del proceso de instalación del sistema es un punto importante y no puede dejarse de lado. El sistema utiliza un conjunto de tecnologías entre las cuales hay servidores, lenguajes, frameworks, etc. El sistema cuenta con dos partes, una página web abierta al público en general y un sistema web, donde solo puede ingresar aquellos que cuenten con un usuario y password. Por lo mismo se puede elegir entre dos tipos de instalación, la primera es la que incluye la página web, en ese caso se tendría que instalar el sistema en un hosting, para lo cual no hace falta instalar nada. El segundo tipo de instalación se realizaría si el cliente solo decide usar el sistema en un establecimiento determinado, en ese caso si se tendría que instalar algunos programas y servidores.

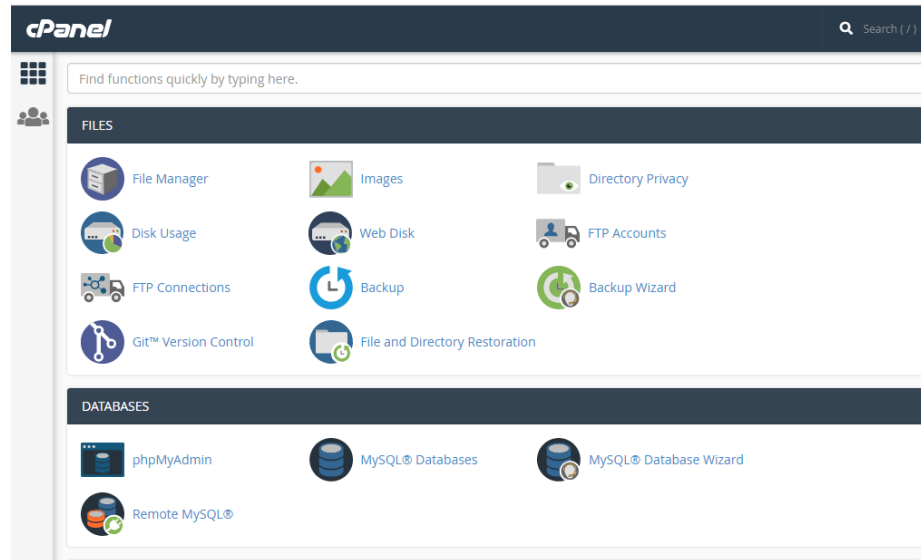
Todas las tecnologías que se usan en el presente proyecto pertenecen a la categoría de software libre, es decir son accesibles para el público de forma gratuita:



2. Instalación en Hosting

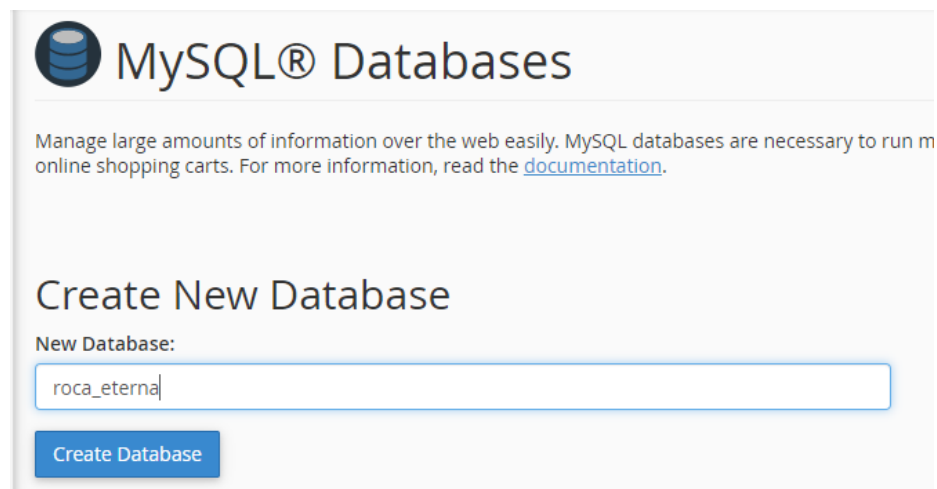
2.1. Acceder al cPanel

Necesita comprar un hosting que tenga cPanel. Una vez comprado, accedemos al cPanel



2.2. Crear la base de datos

Haga click en la opción “Mysql Databases” y luego ingrese el nombre de la base de datos.



Seguidamente baje hacia la sección “Mysql Users”. Ingrese el nombre del nuevo usuario que se creará



MySQL Users

Add New User

Username

roca_eterna

Password

Password (Again)

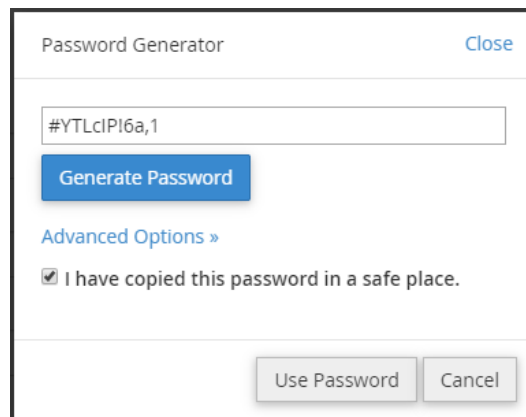
Strength ⓘ

Very Weak (0/100)

Password Generator

Create User

Haga click en “Password Generator”. Active la casilla “I have copied...” y luego haga click en “Use Password”



Password Generator Close

#YTLcIPi6a,1

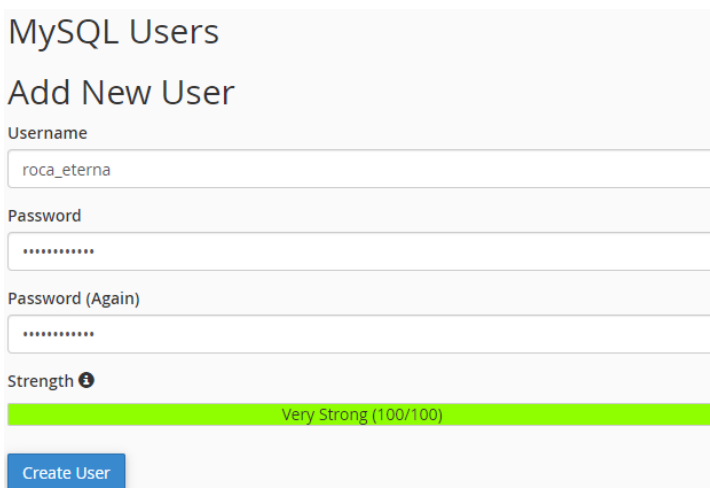
Generate Password

[Advanced Options >](#)

I have copied this password in a safe place.

Use Password Cancel

Finalmente haga click en Create User



MySQL Users

Add New User

Username

roca_eterna

Password

.....

Password (Again)

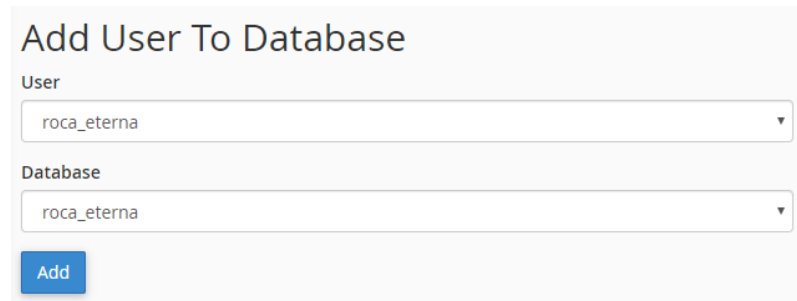
.....

Strength ⓘ

Very Strong (100/100)

Create User

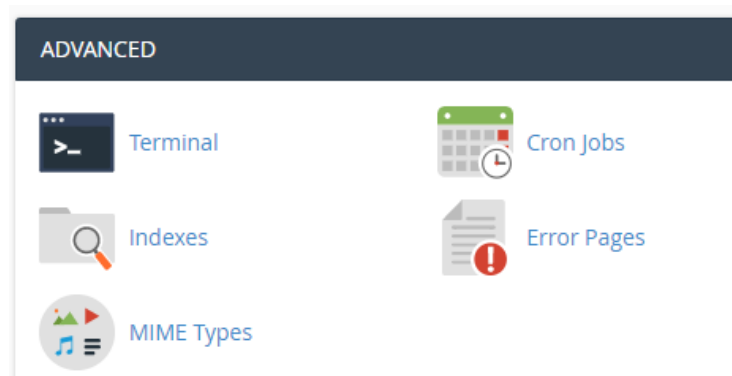
Continué bajando hasta la sección que se muestra en la siguiente imagen. Asegúrese que los campos estén que estén seleccionados sean la Base de Datos y Usuario que acaba de crear y luego haga click en “Add”.



The image shows a web form titled "Add User To Database". It contains two dropdown menus. The first is labeled "User" and has "roca_eterna" selected. The second is labeled "Database" and also has "roca_eterna" selected. Below the dropdowns is a blue button labeled "Add".

2.3. Clonar el proyecto

Vuelva a la página Home y baje hasta la sección “Advanced”, luego haga click en la opción “Terminal”



Ingresan los comandos que se ven a continuación

```
cd public_html  
git init  
git remote add origin https://bitbucket.org/rricardo/sisticket.git  
git pull origin master
```

Si pide alguna clave, ingresan la clave del repositorio

```

Terminal

This interface provides command line access to your account on the server.

awz1ly6k2hl4@ep3plcpnl0923 [~]$ cd public_html/
awz1ly6k2hl4@ep3plcpnl0923 [~/public_html]$ git init
Initialized empty Git repository in /home/awz1ly6k2hl4/public_html/.git/
awz1ly6k2hl4@ep3plcpnl0923 [~/public_html]$ git remote add origin https://bitbucket.org/rricardo/sisticket.git
awz1ly6k2hl4@ep3plcpnl0923 [~/public_html]$ git pull origin master
remote: Counting objects: 137, done.
remote: Total 137 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (137/137), 252.92 KiB | 1.05 MiB/s, done.
Resolving deltas: 100% (22/22), done.
From https://bitbucket.org/rricardo/sisticket
 * branch            master       -> FETCH_HEAD
 * [new branch]      master       -> origin/master
awz1ly6k2hl4@ep3plcpnl0923 [~/public_html]$

```

Para instalar las librerías ejecutan el siguiente comando: composer install.

```

awz1ly6k2hl4@ep3plcpnl0923 [~/public_html]$ composer install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Warning: The lock file is not up to date with the latest changes in composer.json.
update them.
Package operations: 71 installs, 0 updates, 0 removals
 - Installing symfony/thanks (v1.0.5): Downloading (100%)
 - Installing doctrine/inflector (v1.3.0): Loading from cache
 - Installing doctrine/lexer (v1.0.1): Loading from cache
 - Installing dragonmantank/cron-expression (v2.0.0): Downloading (100%)
 - Installing erusev/parsedown (1.6.4): Downloading (100%)
 - Installing vlucas/phpdotenv (v2.4.0): Downloading (100%)
 - Installing symfony/css-selector (v4.0.4): Downloading (100%)
 - Installing tijsverkoyen/css-to-inline-styles (2.2.1): Loading from cache
 - Installing symfony/polyfill-php72 (v1.7.0): Downloading (100%)
 - Installing symfony/polyfill-mbstring (v1.7.0): Downloading (100%)
 - Installing symfony/var-dumper (v4.0.4): Downloading (100%)
 - Installing symfony/routing (v4.0.4): Downloading (100%)
 - Installing symfony/process (v4.0.4): Downloading (100%)
 - Installing symfony/http-foundation (v4.0.4): Downloading (100%)

```

2.4. Configurar el proyecto

Primero crea el archivo .env, para eso escribe esto en la terminal nano .env.example y modificas según se muestra en las imágenes:

```

LOG_CHANNEL=stack
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=roca_eterna
DB_USERNAME=roca_eterna
DB_PASSWORD=txP$${nKNY2]N
BROADCAST_DRIVER=log
REDIS_PORT=6379
MAIL_DRIVER=smtp
MAIL_HOST=smtp.gmail.com
MAIL_PORT=587
MAIL_USERNAME=roca.eterna.ib@gmail.com
MAIL_PASSWORD=oiyahhucdwormvxb
MAIL_ENCRYPTION=tls
MAIL_FROM_NAME='Rumbo Agil'
PUSHER_APP_ID=

```

Finalmente presionas **CTRL+X**, luego escribes **Y**, le das enter. Cambias el nombre a: **.env** le das enter y luego escribes **Y**

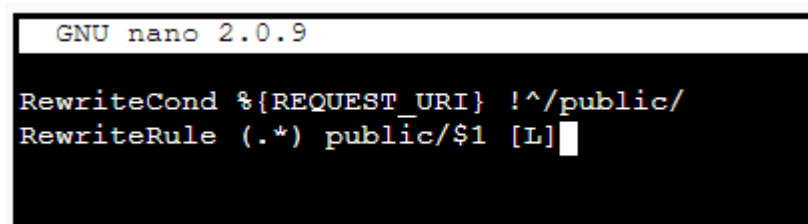
Finalmente escribes **php artisan key:gen** en la terminal y luego **ls**, para verificar que el archivo **.env** esté creado

```
awz1ly6k2h14@p3plcpnl0923 [~/public_html]$ nano .env.example
awz1ly6k2h14@p3plcpnl0923 [~/public_html]$ php artisan key:gen
Application key [base64:gh6AAfMPF5pkDiCmX8ckBldv7dbD/+t0Jt7Zj1D+H
awz1ly6k2h14@p3plcpnl0923 [~/public_html]$ ls
./      .env.example  .gitignore  artisan      composer.json  data
../     .git/         README.md   bootstrap/   composer.lock  pack
.env    .gitattributes app/         cgi-bin/     config/        php
awz1ly6k2h14@p3plcpnl0923 [~/public_html]$
```

Ahora restauramos la base de datos a partir del archivo **.sql** que nos pasaron. Deberás subir ese archivo a la carpeta **public_html**. Luego ejecutarás el siguiente comando **mysql -u roca_eterna -p roca_eterna < roca_eterna_db.sql** , ingresas la contraseña de la Base de datos (2.2) y presionas enter.

Seguidamente crearás el archivo **.htaccess**. Escribes en la terminal **nano .htaccess** e ingresas lo siguiente

```
RewriteCond %{REQUEST_URI} !^/public/
RewriteRule (.*?) public/$1 [L]
```



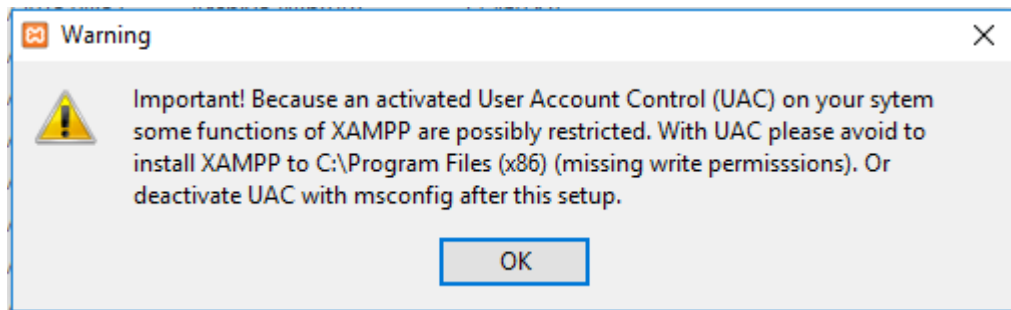
```
GNU nano 2.0.9
RewriteCond %{REQUEST_URI} !^/public/
RewriteRule (.*?) public/$1 [L]
```

Finalmente presionas **CTRL+X**, luego **Y** y finalmente **Enter**

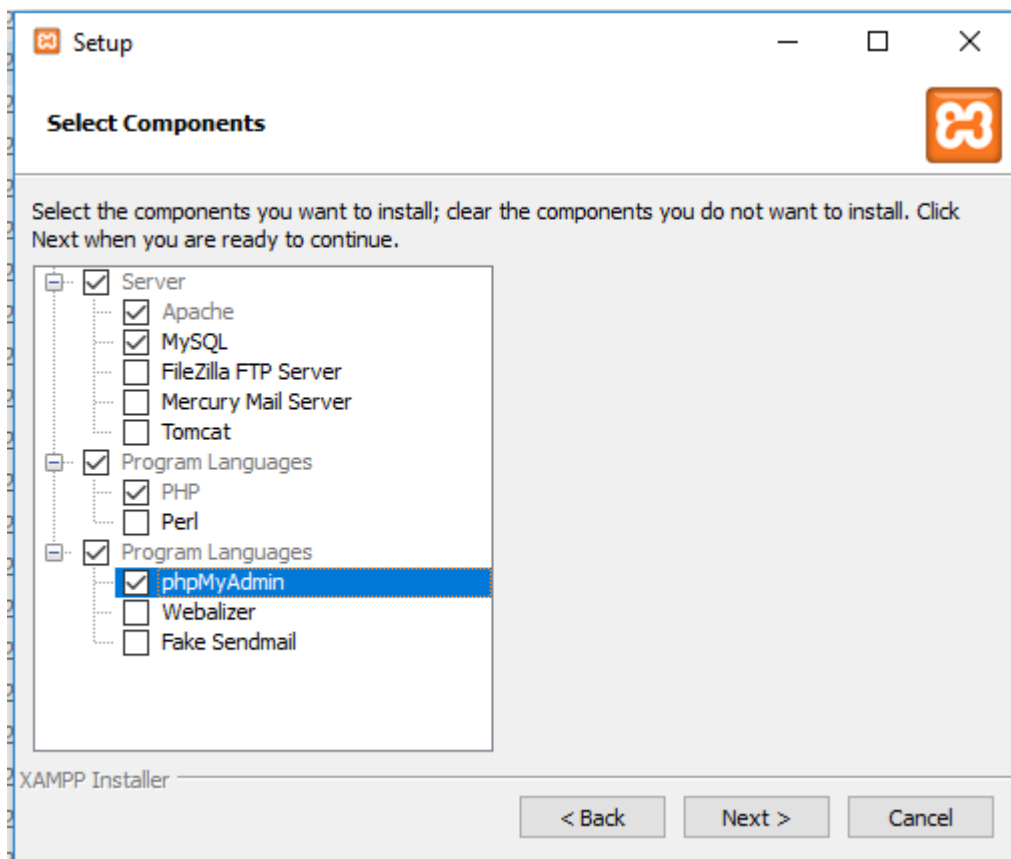
Para el segundo tipo de instalación deberemos instalar apache, mysql, PHP, composer y git.

3. Instalación de XAMPP (Apache, Mysql, PHP)

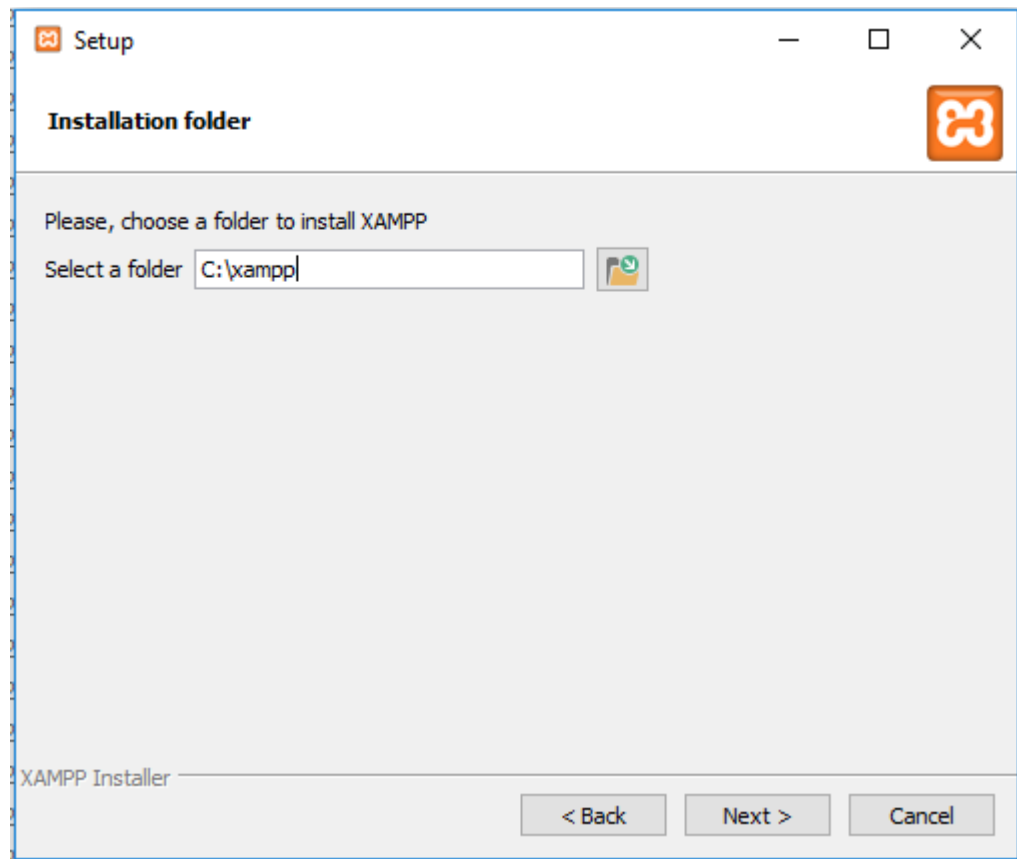
La herramienta de XAMPP es un paquete en el que vienen distintas tecnologías, dentro de las cuales están las que necesitamos. Hacemos click en el instalador de XAMPP y presionamos OK:



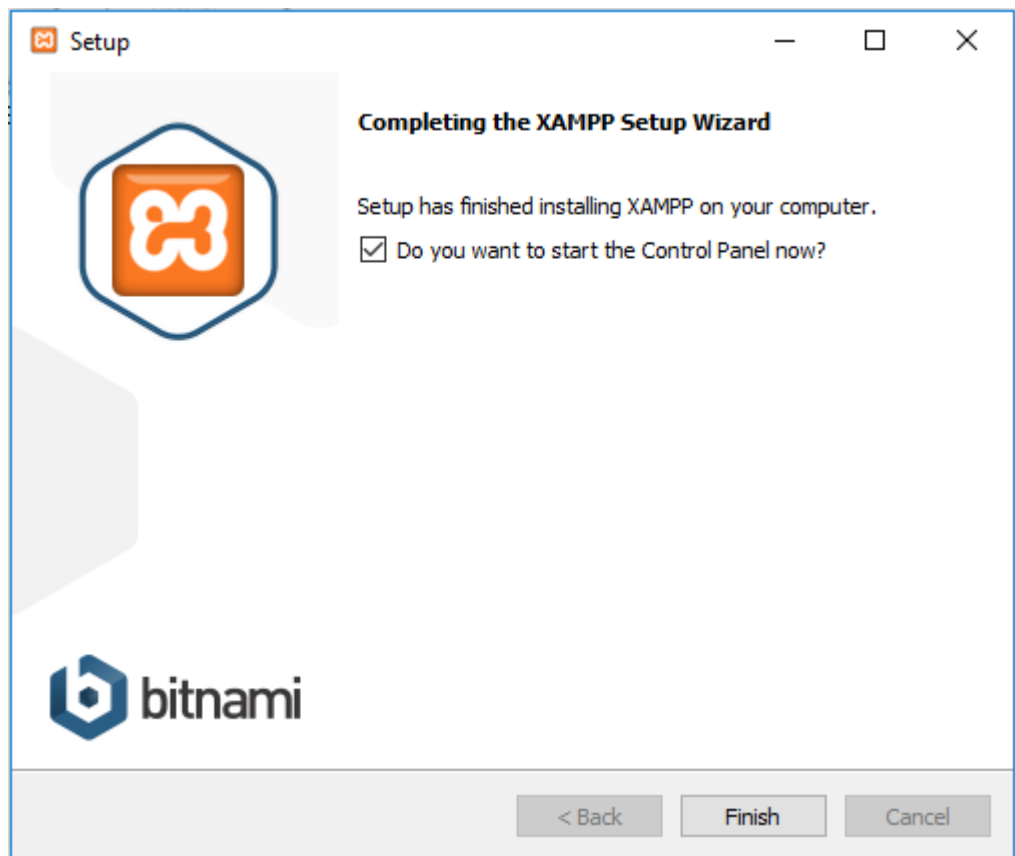
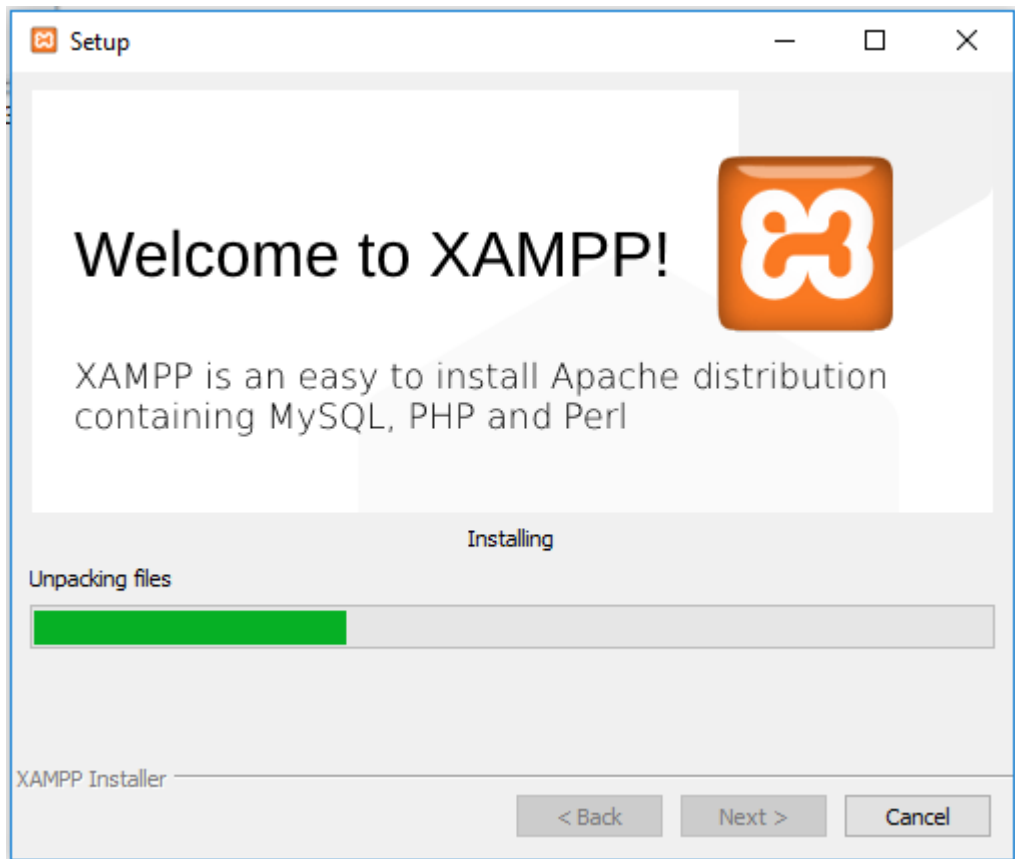
Como vemos, PHP y Apache ya están seleccionados, además de esos dos, seleccionamos la casilla Mysql y phpMyAdmin. Las demás las desactivamos.



Dejar por defecto la carpeta donde se instalará XAMPP

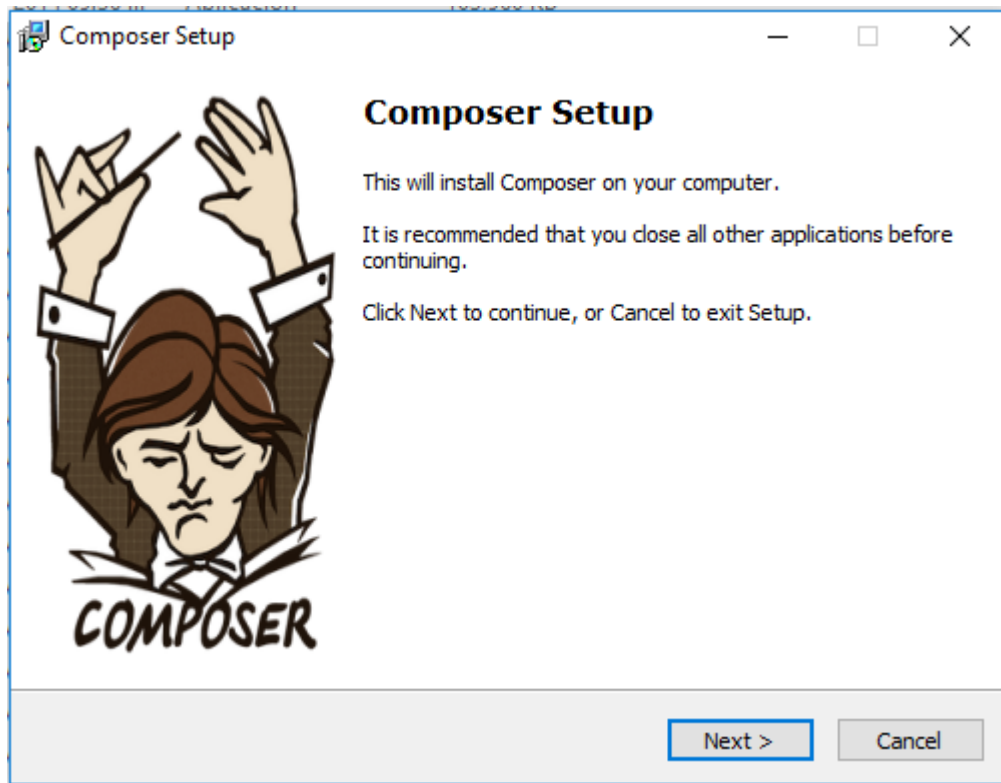


Esperar a que termine la instalación y presionar en Finish:

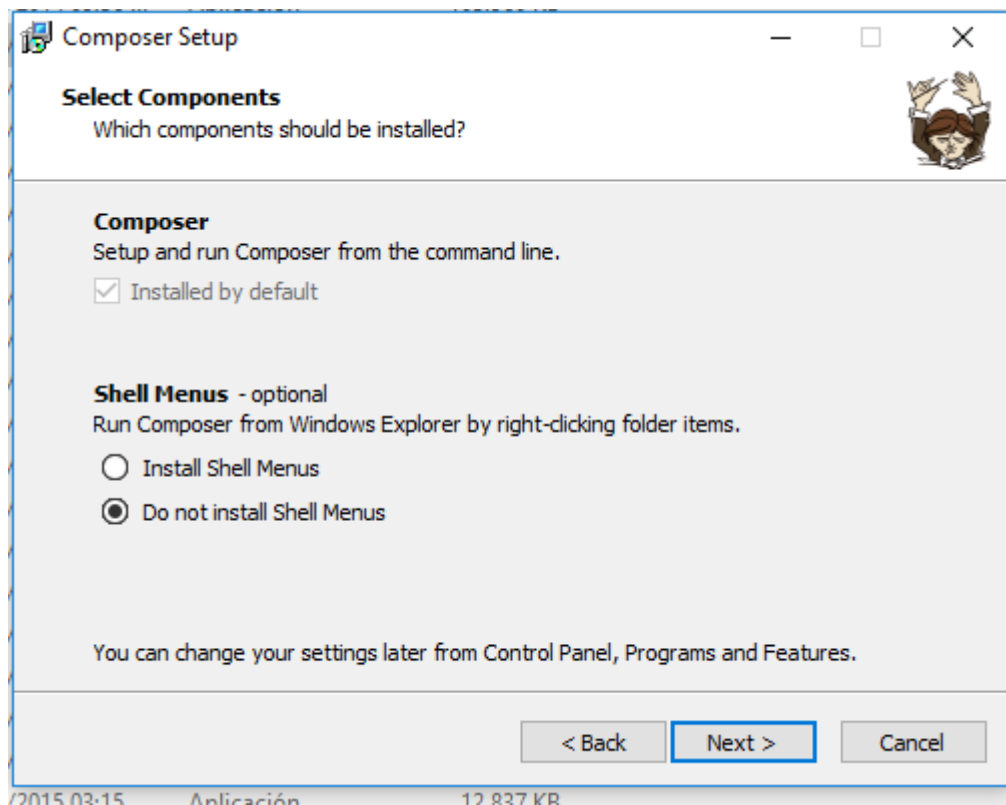


4. Instalar Composer

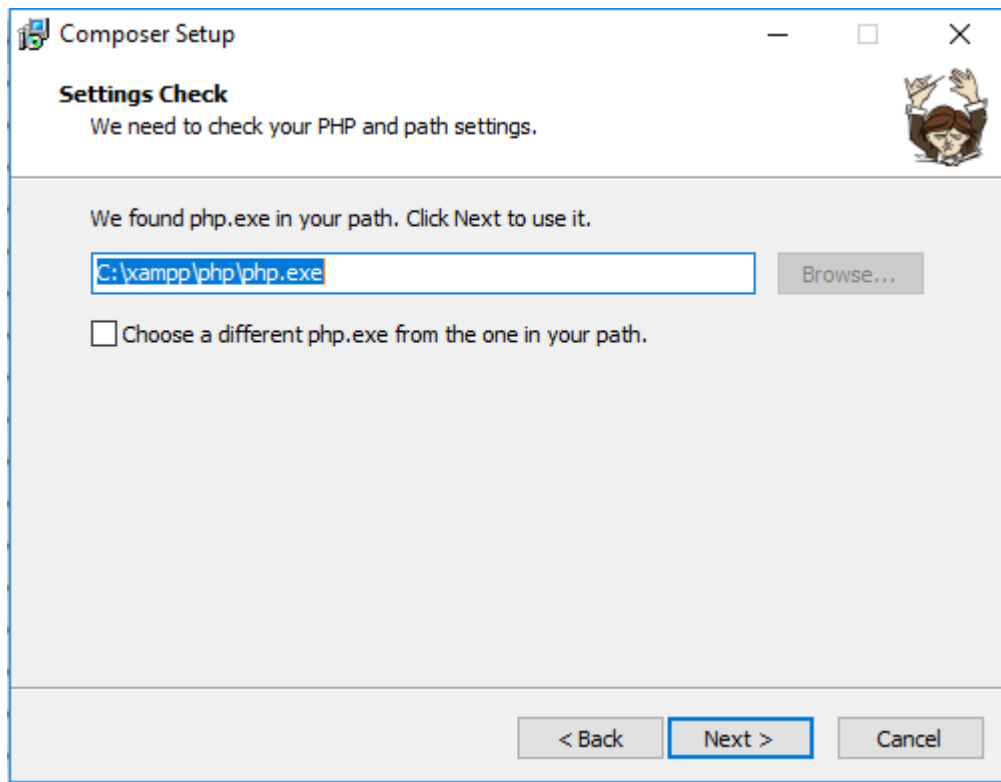
Hacer doble click en el instalador de composer y presionar Next



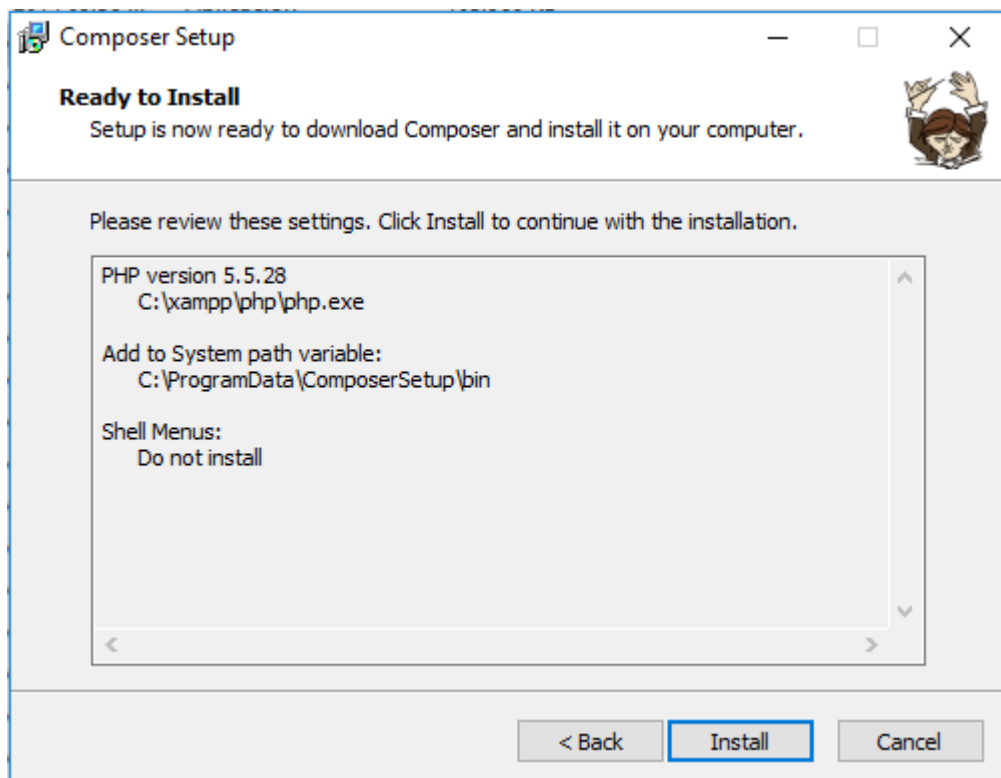
Dejamos todo como se ve a continuación y presionamos Next



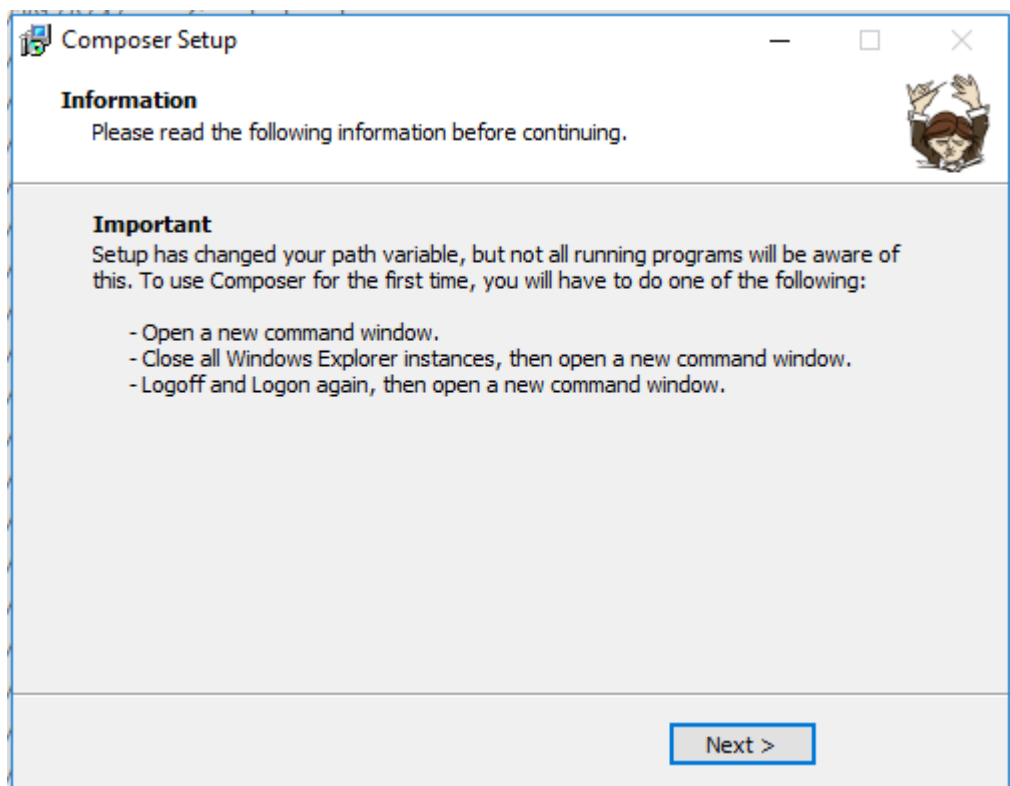
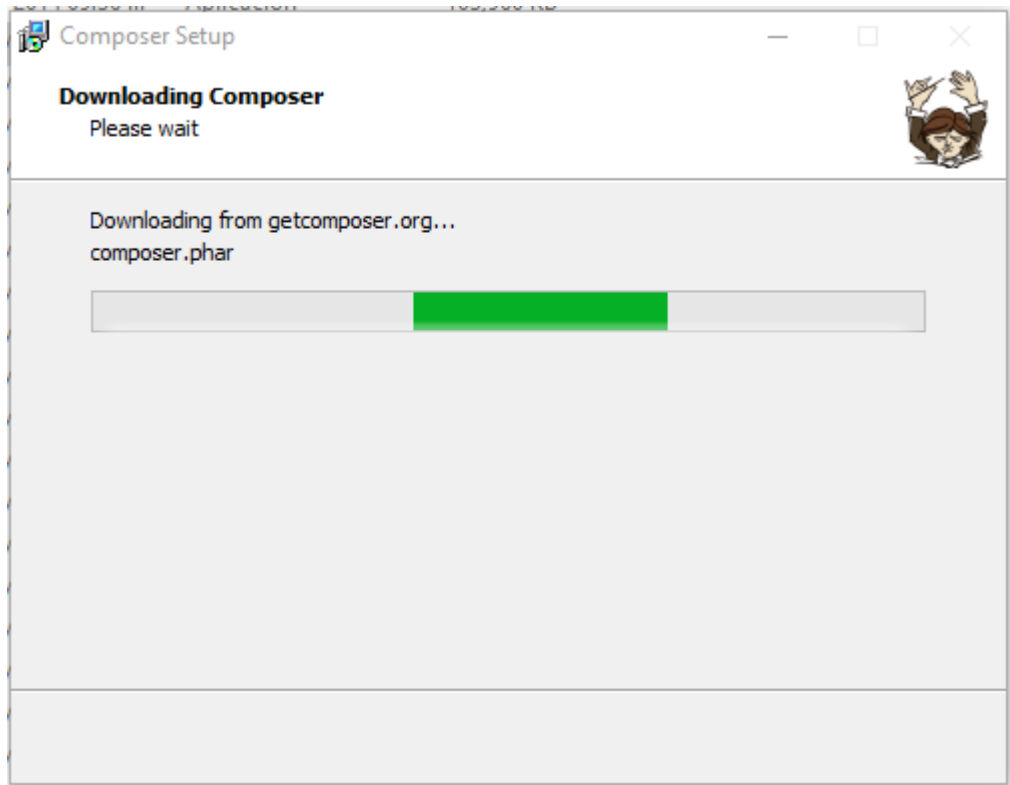
Detectará la carpeta de PHP que ha instalado el XAMPP. Presionamos Next



Presione Next en las otras ventanas que aparecerán hasta que veas la siguiente, haga click en Install



Espere hasta que la instalación haya finalizado



Finalmente desactive la casilla y haga click en Finish

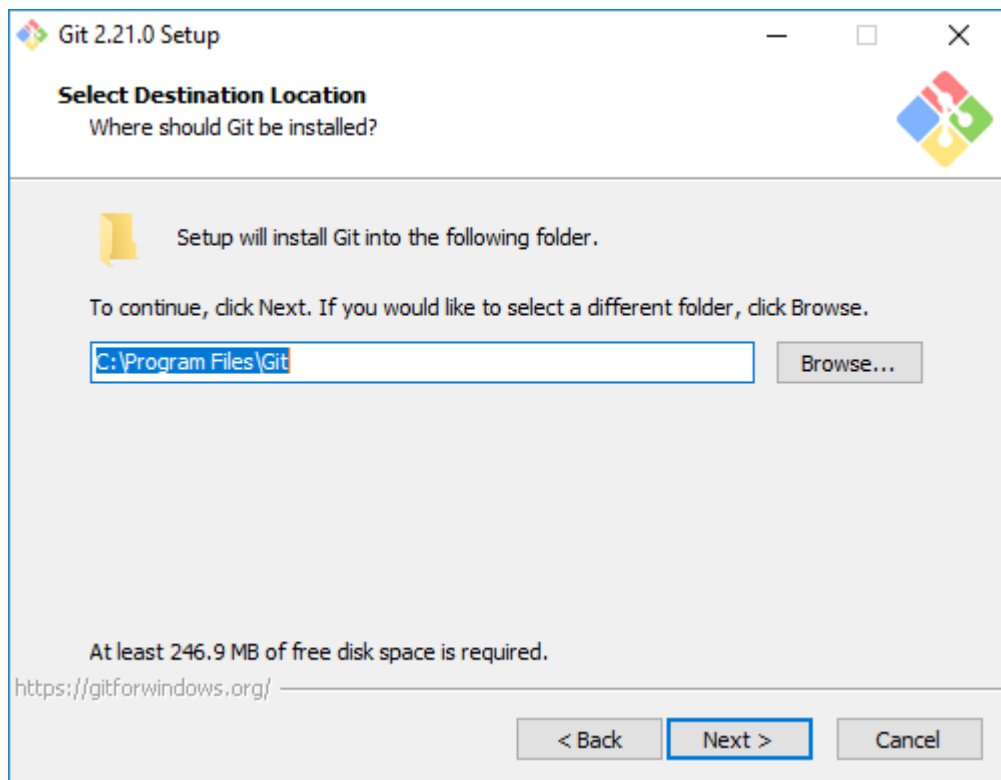


5. Instalar Git

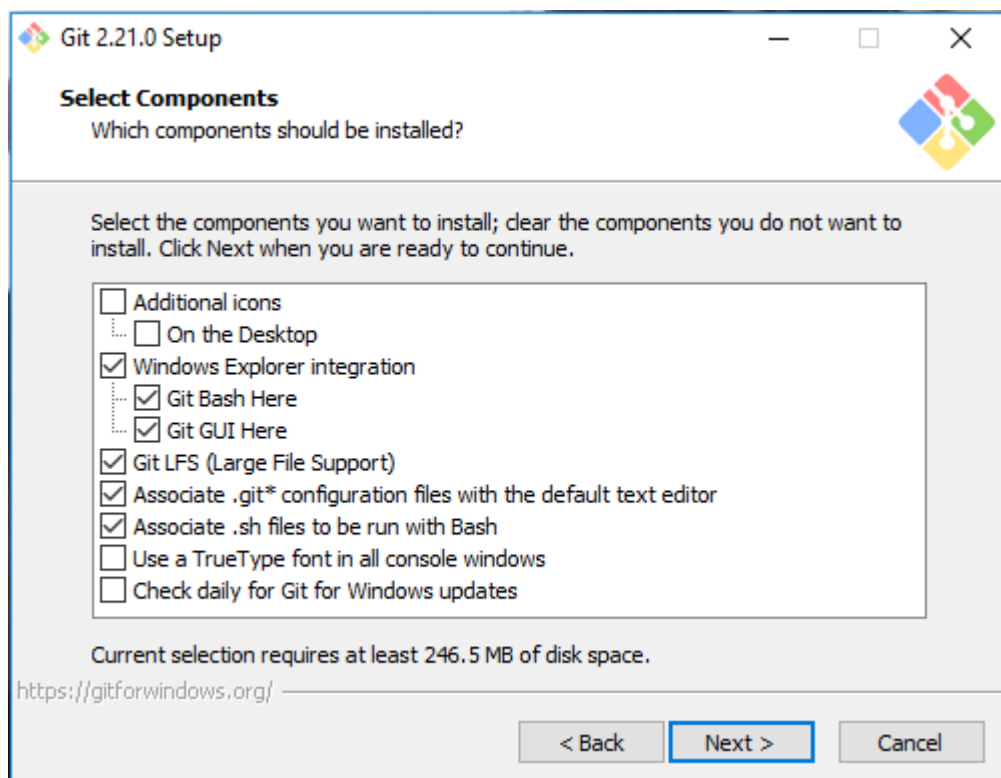
Haga click en el instalador de git acepte los permisos y en la ventana que le aparecerá luego haga click en Next

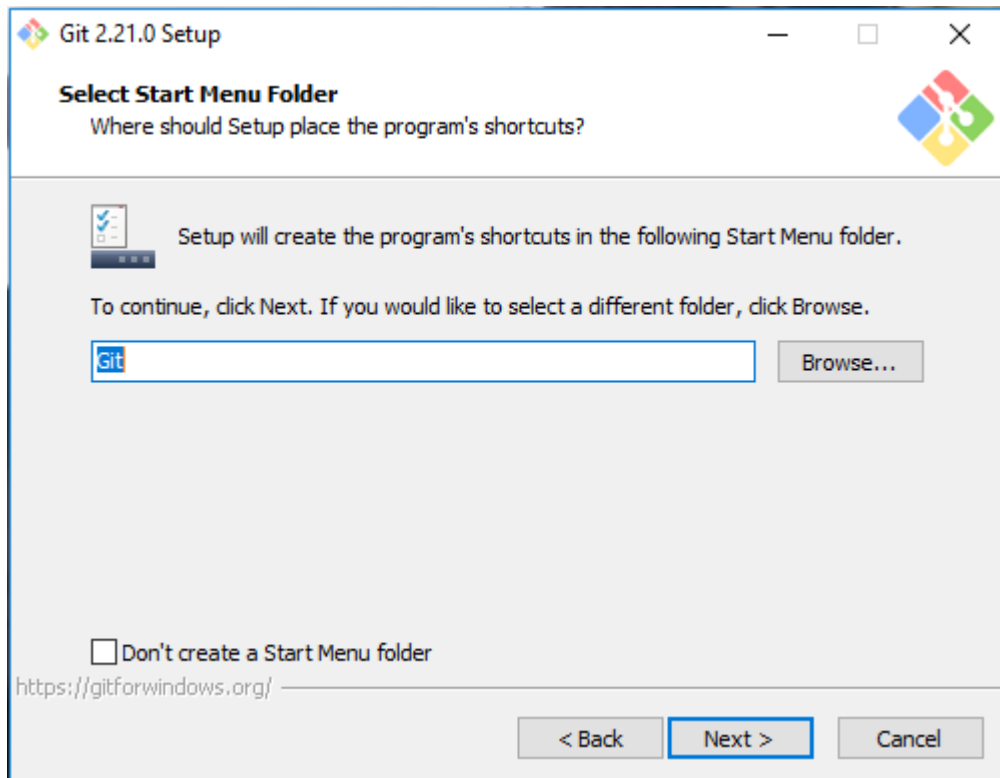


Por defecto deje la ruta especificada

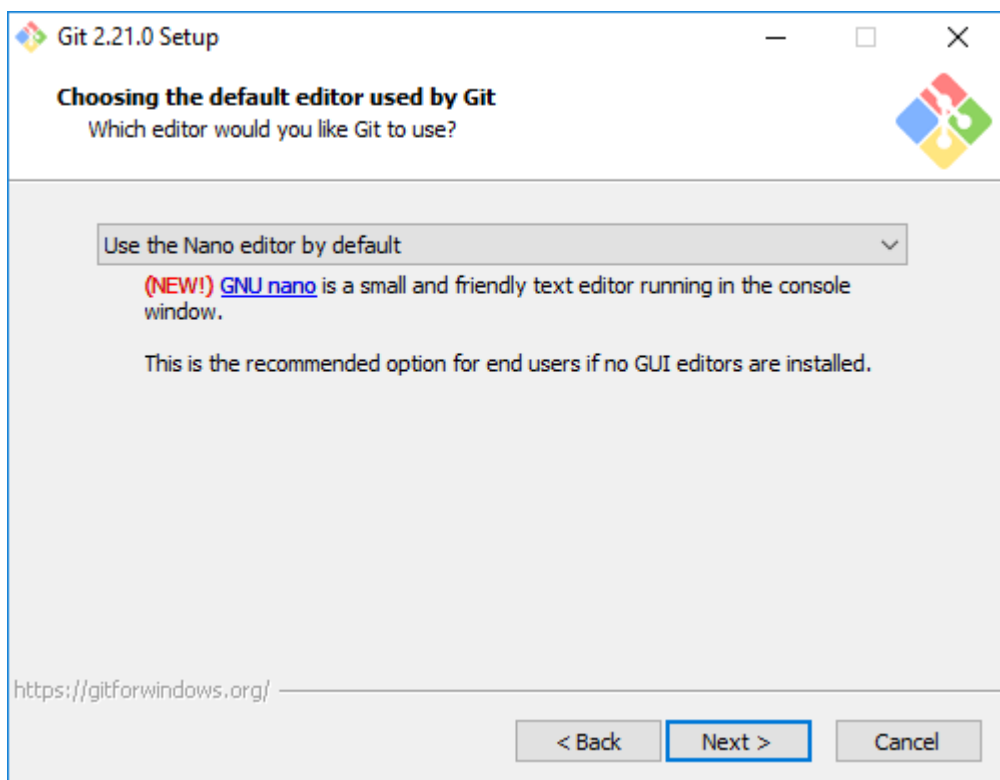


Haga click en Next en las ventanas siguientes

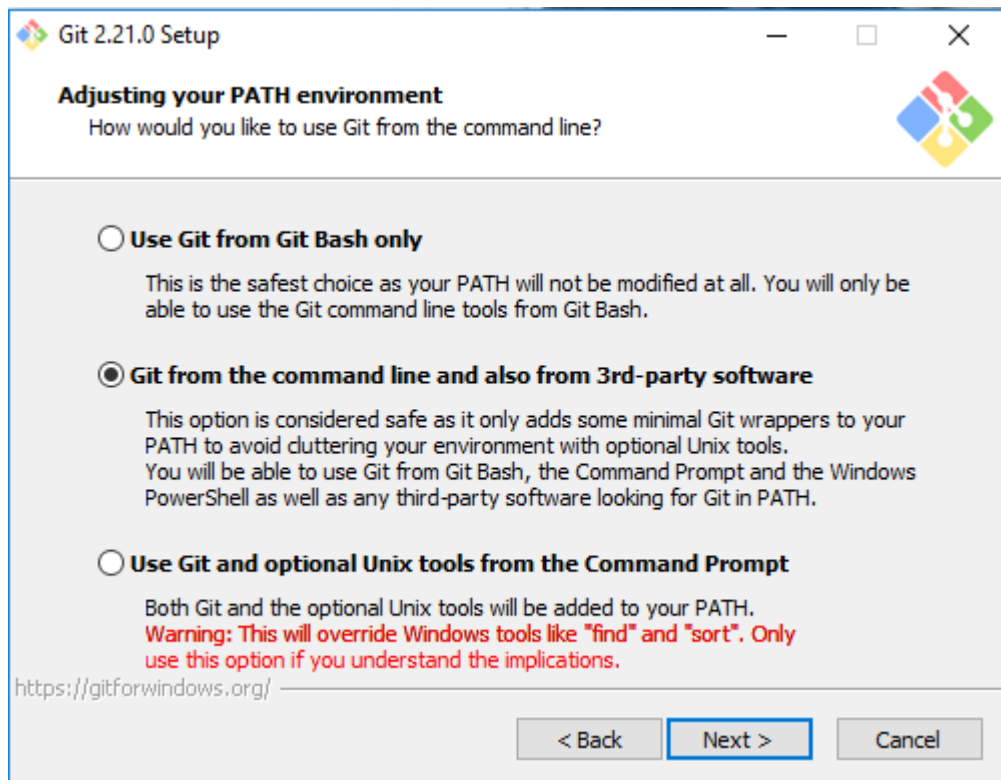




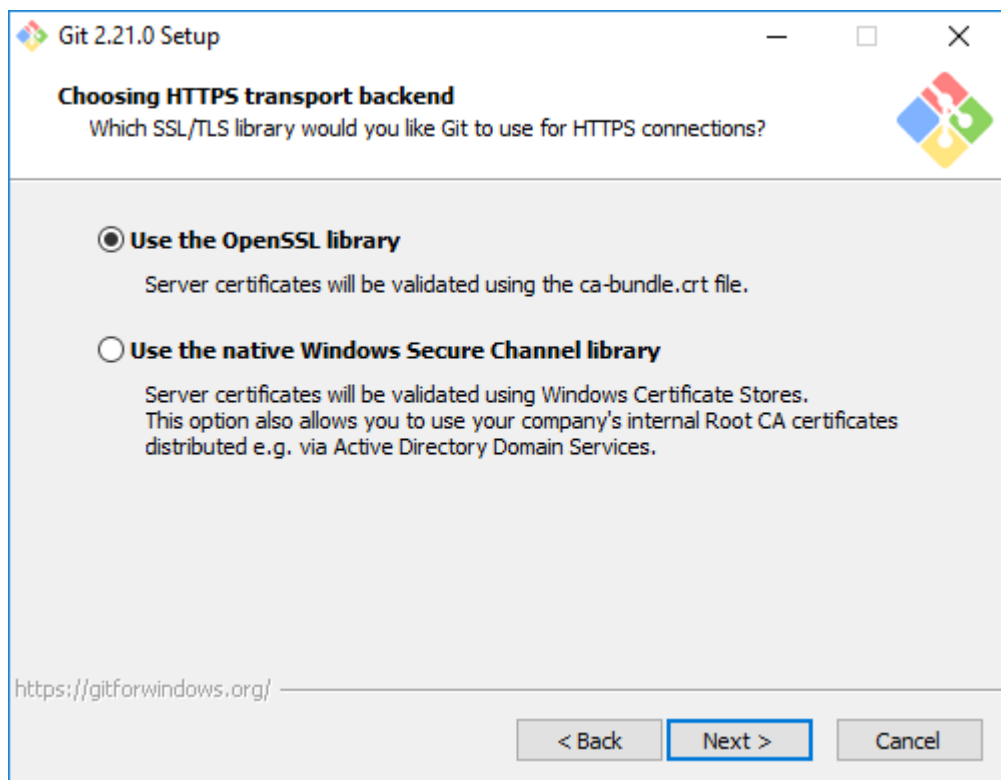
Aquí elija la opción que aparece a continuación



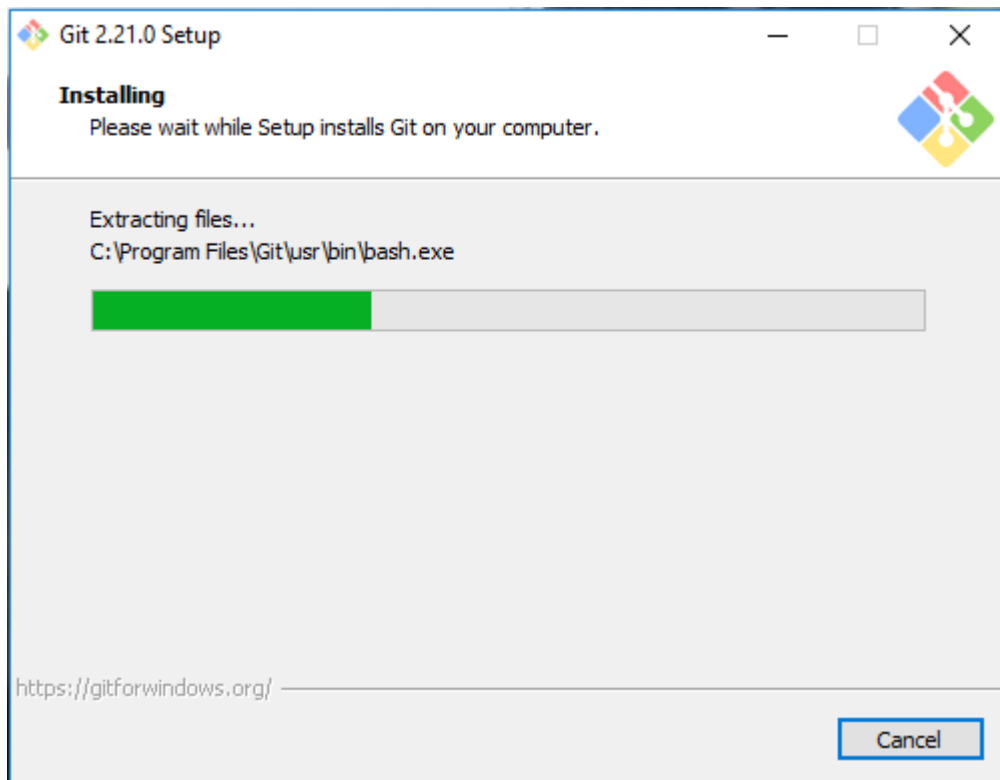
Elija la segunda opción



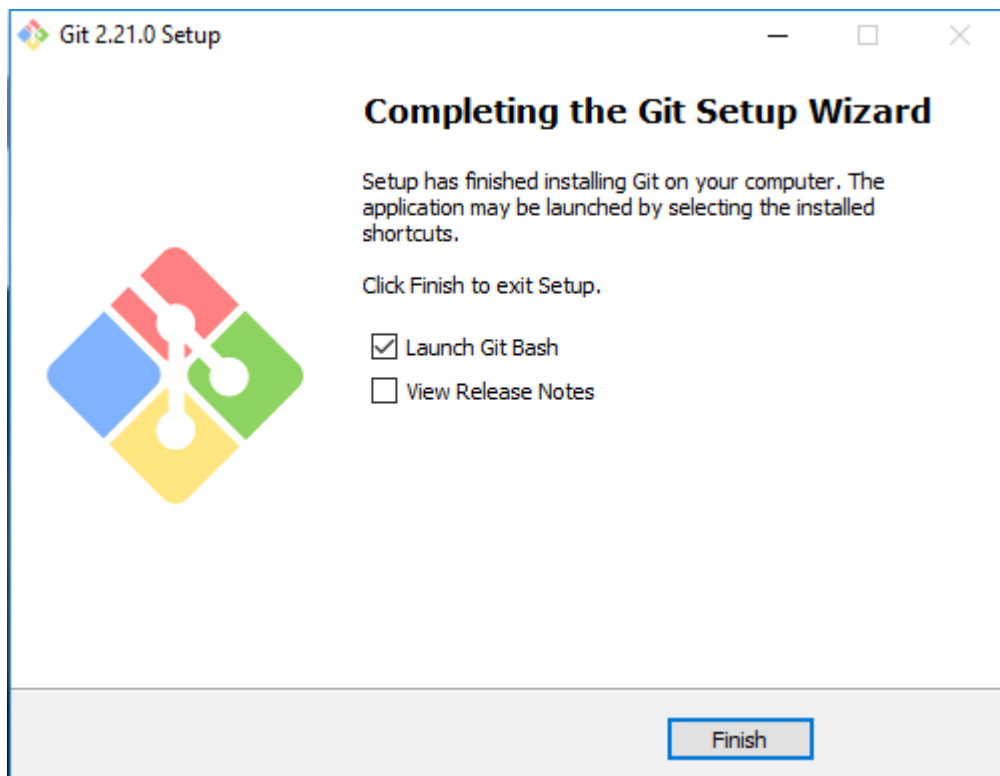
Haga click en next en las siguientes ventanas que aparezcan



Espere hasta que termine de instalarse





Active la casilla de Launch Git Bash y haga click en Finish



6. Crear la Base de Datos

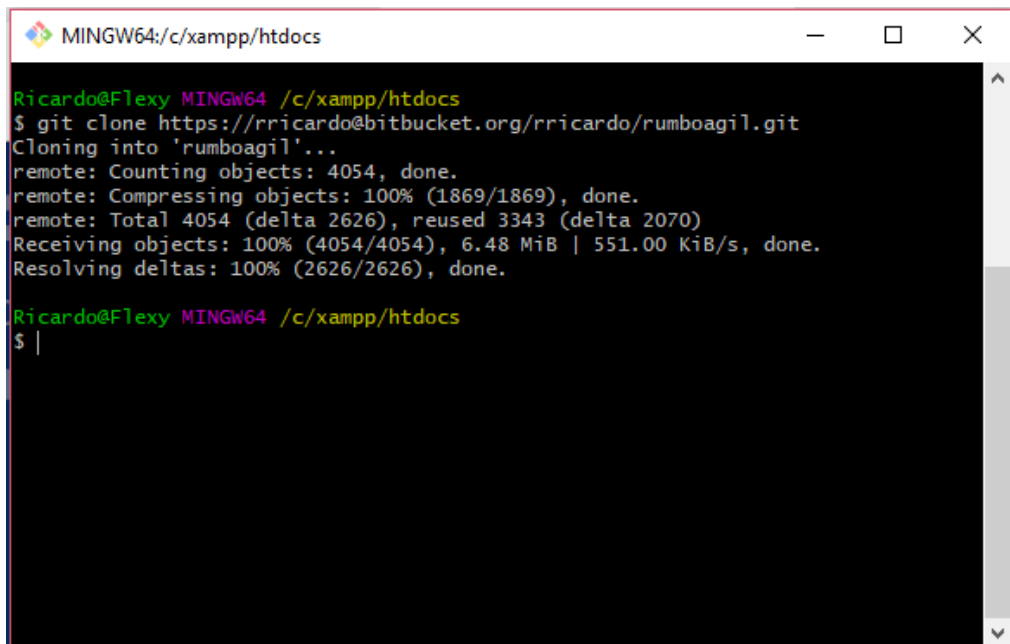
Ingrese en la url la siguiente dirección: <http://localhost/phpmyadmin/> luego ingrese el nombre de la Base de Datos y haga click en Crear:

Bases de datos

 Crear base de datos 

7. Clonar el proyecto

Abra git bash y ubíquese en la siguiente ruta C:/xampp/htdocs. Luego escriba: git clone https://bitbucket.org/rricardo/sisticket.git. Presione enter y espere a que el proyecto termine de clonarse.



```
MINGW64:/c:/xampp/htdocs
Ricardo@Flexy MINGW64 /c:/xampp/htdocs
$ git clone https://rricardo@bitbucket.org/rricardo/rumboagil.git
Cloning into 'rumboagil'...
remote: Counting objects: 4054, done.
remote: Compressing objects: 100% (1869/1869), done.
remote: Total 4054 (delta 2626), reused 3343 (delta 2070)
Receiving objects: 100% (4054/4054), 6.48 MiB | 551.00 KiB/s, done.
Resolving deltas: 100% (2626/2626), done.
Ricardo@Flexy MINGW64 /c:/xampp/htdocs
$ |
```

Ingrese los comandos que aparecen a continuación y espere a que termine de cargar las librerías

```
MINGW64:/c:/xampp/htdocs/rumboagil
Resolving deltas: 100% (2626/2626), done.

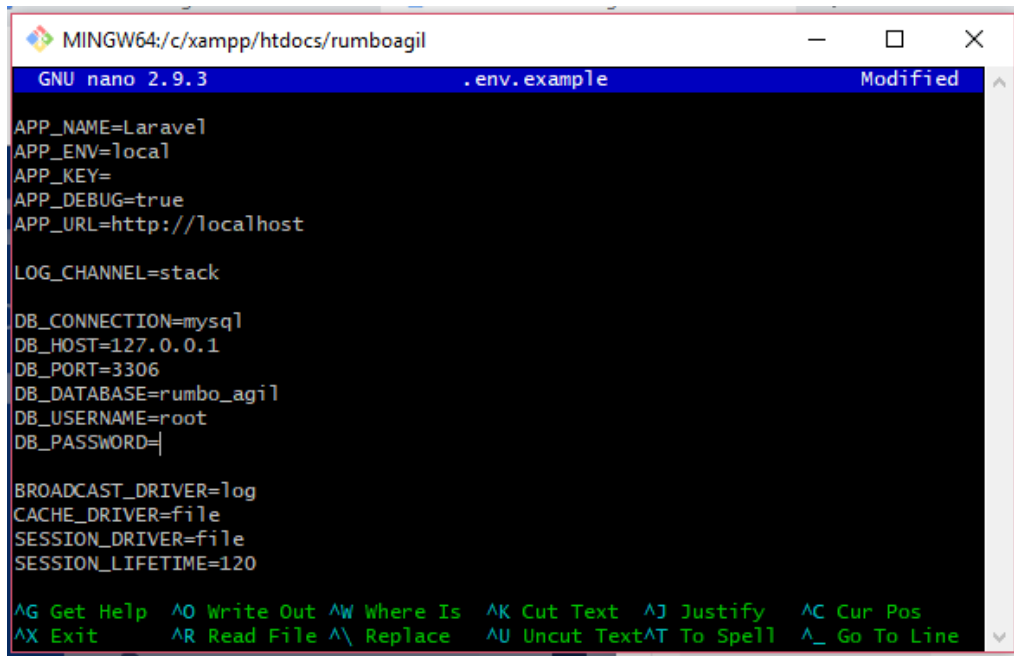
Ricardo@Flexy MINGW64 /c:/xampp/htdocs
$ cd rumboagil/

Ricardo@Flexy MINGW64 /c:/xampp/htdocs/rumboagil (master)
$ composer install
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 112 installs, 0 updates, 0 removals
 - Installing symfony/polyfill-ctype (v1.10.0): Downloading (100%)
 - Installing vlucas/phpdotenv (v2.6.1): Loading from cache
 - Installing symfony/css-selector (v4.2.3): Loading from cache
 - Installing tijsverkoyen/css-to-inline-styles (2.2.1): Loading from cache
 - Installing symfony/polyfill-php72 (v1.10.0): Downloading (100%)
 - Installing symfony/polyfill-mbstring (v1.10.0): Downloading (100%)
 - Installing symfony/var-dumper (v4.2.3): Downloading (100%)
 - Installing symfony/routing (v4.2.3): Downloading (100%)
 - Installing symfony/process (v4.2.3): Downloading (100%)
 - Installing symfony/http-foundation (v4.2.3): Downloading (100%)
 - Installing symfony/contracts (v1.0.2): Loading from cache
 - Installing symfony/event-dispatcher (v4.2.3): Downloading (100%)
 - Installing psr/log (1.1.0): Loading from cache
 - Installing symfony/debug (v4.2.3): Downloading (connecting...)|
```

```
MINGW64:/c:/xampp/htdocs/rumboagil
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.6.0)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0)
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover
Discovered Package: barryvdh/laravel-dompdf
Discovered Package: cartalyst/sentinel
Discovered Package: fideloper/proxy
Discovered Package: intervention/image
Discovered Package: laracasts/utilities
Discovered Package: laravel/tinker
Discovered Package: laravelcollective/html
Discovered Package: laraveles/spanish
Discovered Package: nexmo/laravel
Discovered Package: nunomaduro/collision
Discovered Package: unisharp/laravel-filemanager
Package manifest generated successfully.

Ricardo@Flexy MINGW64 /c:/xampp/htdocs/rumboagil (master)
$ |
```

Luego, escribir el siguiente comando. **nano .env.example** y modificar como se muestra a continuación



```
MINGW64:/c:/xampp/htdocs/rumboagil
GNU nano 2.9.3 .env.example Modified
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

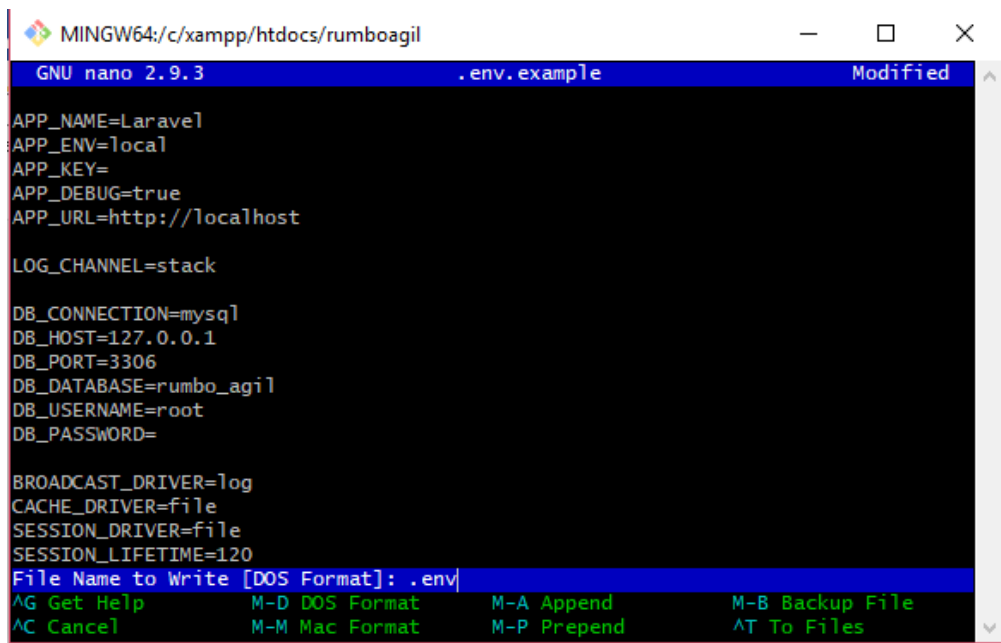
LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=rumbo_agil
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Presionar Ctrl+X, luego Y, y finalmente escribimos `.env`, como se muestra a continuación. Acepte la creación de un nuevo archivo.



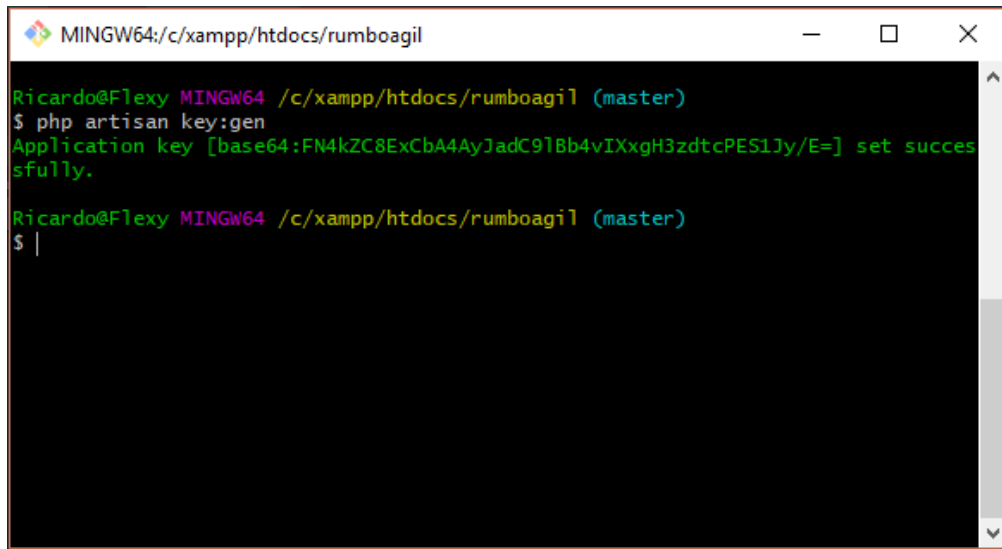
```
MINGW64:/c:/xampp/htdocs/rumboagil
GNU nano 2.9.3 .env.example Modified
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=rumbo_agil
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
File Name to Write [DOS Format]: .env
^G Get Help M-D DOS Format M-A Append M-B Backup File
^C Cancel M-M Mac Format M-P Prepend ^T To Files
```

Ingrese el siguiente comando: `php artisan key:gen`

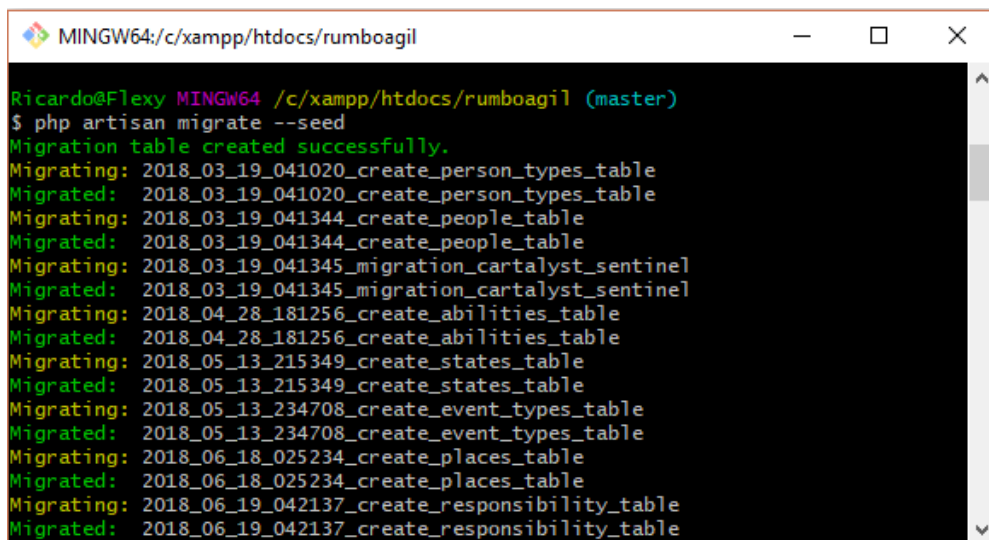


```
MINGW64:/c/xampp/htdocs/rumboagil
Ricardo@Flexy MINGW64 /c/xampp/htdocs/rumboagil (master)
$ php artisan key:gen
Application key [base64:FN4kZC8ExCbA4AyJadC9lBb4vIXgH3zdtcPES1Jy/E=] set successfully.

Ricardo@Flexy MINGW64 /c/xampp/htdocs/rumboagil (master)
$ |
```

8. Correr las migraciones

En la misma consola de git, escribimos el siguiente comando: **php artisan migrate --seed**. Este comando creará las tablas necesarias para el sistema y cargará algunos datos por defecto.



```
MINGW64:/c/xampp/htdocs/rumboagil
Ricardo@Flexy MINGW64 /c/xampp/htdocs/rumboagil (master)
$ php artisan migrate --seed
Migration table created successfully.
Migrating: 2018_03_19_041020_create_person_types_table
Migrated: 2018_03_19_041020_create_person_types_table
Migrating: 2018_03_19_041344_create_people_table
Migrated: 2018_03_19_041344_create_people_table
Migrating: 2018_03_19_041345_migration_cartalyst_sentinel
Migrated: 2018_03_19_041345_migration_cartalyst_sentinel
Migrating: 2018_04_28_181256_create_abilities_table
Migrated: 2018_04_28_181256_create_abilities_table
Migrating: 2018_05_13_215349_create_states_table
Migrated: 2018_05_13_215349_create_states_table
Migrating: 2018_05_13_234708_create_event_types_table
Migrated: 2018_05_13_234708_create_event_types_table
Migrating: 2018_06_18_025234_create_places_table
Migrated: 2018_06_18_025234_create_places_table
Migrating: 2018_06_19_042137_create_responsibility_table
Migrated: 2018_06_19_042137_create_responsibility_table
```

```
MINGW64:/c/xampp/htdocs/rumboagil
Migrating: 2019_03_11_151743_update_events_table_add_price
Migrated: 2019_03_11_151743_update_events_table_add_price
Migrating: 2019_03_15_153738_update_templates_people_table
Migrated: 2019_03_15_153738_update_templates_people_table
Migrating: 2019_04_01_001209_create_departments_table
Migrated: 2019_04_01_001209_create_departments_table
Migrating: 2019_04_01_001210_create_districts_table
Migrated: 2019_04_01_001210_create_districts_table
Migrating: 2019_04_02_233752_update_people_table_add_cols
Migrated: 2019_04_02_233752_update_people_table_add_cols
Seeding: EventTypeTableSeeder
Seeding: PersonTypeTableSeeder
Seeding: StatesTableSeeder
Seeding: RolTableSeeder
Seeding: ConstantTableSeeder
Seeding: PersonTableSeeder
Seeding: UserTableSeeder

Ricardo@Flexy MINGW64 /c/xampp/htdocs/rumboagil (master)
$
```

9. Correr el programa

Ejecutar el siguiente comando: `php artisan serve`

```
MINGW64:/c/xampp/htdocs/rumboagil
Ricardo@Flexy MINGW64 /c/xampp/htdocs/rumboagil (master)
$ php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```