



# Documento de Especificación del Proyecto

MAURO AMARANTE ESPARZA	A01191903
MARTHA BENAVIDES MARTÍNEZ	A01280115
DAVID MARTÍNEZ GARZA	A01191485
JESÚS NAVARRO MARÍN	A00813111

EQUIPO 2

## Contenido

I. Introducción.....	2
II. Principales Objetivos.....	2
III. Criterios de Aceptación.....	2
IV. Mapa del Sitio.....	3
IV.I Contenido de la Aplicación Web.....	3
V. Especificaciones Funcionales .....	4
VI. Especificaciones Técnicas.....	4
VII. Plan de Contenido .....	9
VII.I. Control de Versiones .....	9
VII.II. Responsabilidades.....	9
VII.III. WBS .....	10
VII.IV. Estimación de Tiempo .....	11
VII.V. Diagrama de Gantt.....	12
VII.VI. Metas Intermedias y Reuniones.....	13
VIII. Plan de Pruebas .....	14
IX. Apéndices.....	15
IX.I. Notas de las Juntas .....	15
IX.II. Conceptos de Diseño.....	16
IX.III. Prototipo .....	19
IX.IV. Suposiciones .....	19
IX.V. Glosario de Términos.....	19
X. Firmas del Cliente y el Líder del Equipo .....	19
X. Firmas del Cliente y del Líder del Equipo.....	19

## ***I. Introducción***

El presente documento contiene la especificación del proyecto realizado por MDJ, conformado por Mauro Amarante, Martha Benavides, David Martínez y Jesús Navarro, para el Centro Estudiantil. En el mismo, se detallan los acuerdos a los que tanto el equipo, representado por Mauro Amarante y mencionado a partir de este momento en el documento como MDJ, como Centro Estudiantil, representado por Alejandra Garza González y mencionada a partir de este momento en el documento como El Cliente se comprometen de común acuerdo.

Asimismo, este documento es una especificación formal de requerimientos, que de común acuerdo tanto El Cliente como MDJ han generado, con la intención de marcar la pauta en el trabajo que será realizado por MDJ durante el período de Enero – Mayo 2017. Se extiende la confidencialidad del mismo a: El Cliente, MDJ, y al Ing. Alfredo Salazar, mencionado a partir de este momento en el documento como El Profesor.

El propósito del sistema es facilitar la administración de grupos estudiantiles y sus eventos en el Tec de Monterrey Campus Monterrey. En los últimos años se ha utilizado un sistema el cual el Cliente, Alejandra Garza González, desea recrear y agregar más funcionalidades que considera necesarias para administrar a todos los diferentes grupos estudiantiles. Este sistema ayuda al Cliente a organizar todos sus eventos dentro del campus. El Cliente quiere facilitar y simplificar estos procesos para poder ser más productivos y efectivos al momento de administrar los grupos y sus eventos, por lo que MDJ pretende desarrollar un sistema que cumpla con las necesidades de los administradores y alumnos de los grupos estudiantiles.

## ***II. Principales Objetivos***

El principal objetivo de la aplicación web será el proporcionar a los estudiantes pertenecientes a los grupos estudiantiles del Tecnológico de Monterrey la oportunidad de registrar sus eventos, a realizar en las instalaciones de la universidad, de manera rápida y sencilla a través de una interfaz amigable, en la cual puedan agregar todos los datos necesarios de forma que los administradores de los grupos estudiantiles puedan validar en línea los eventos para facilitar el proceso y administrar el progreso de la planeación, dando paso a que se tenga un mayor control sobre los mismos en una misma plataforma, evitando de esta manera re-trabajo innecesario.

## ***III. Criterios de Aceptación***

La aplicación web está diseñada para ser la herramienta que los administradores y alumnos necesitan para administrar los eventos de grupos estudiantiles, por lo que los criterios de aceptación definidos por El Cliente son:

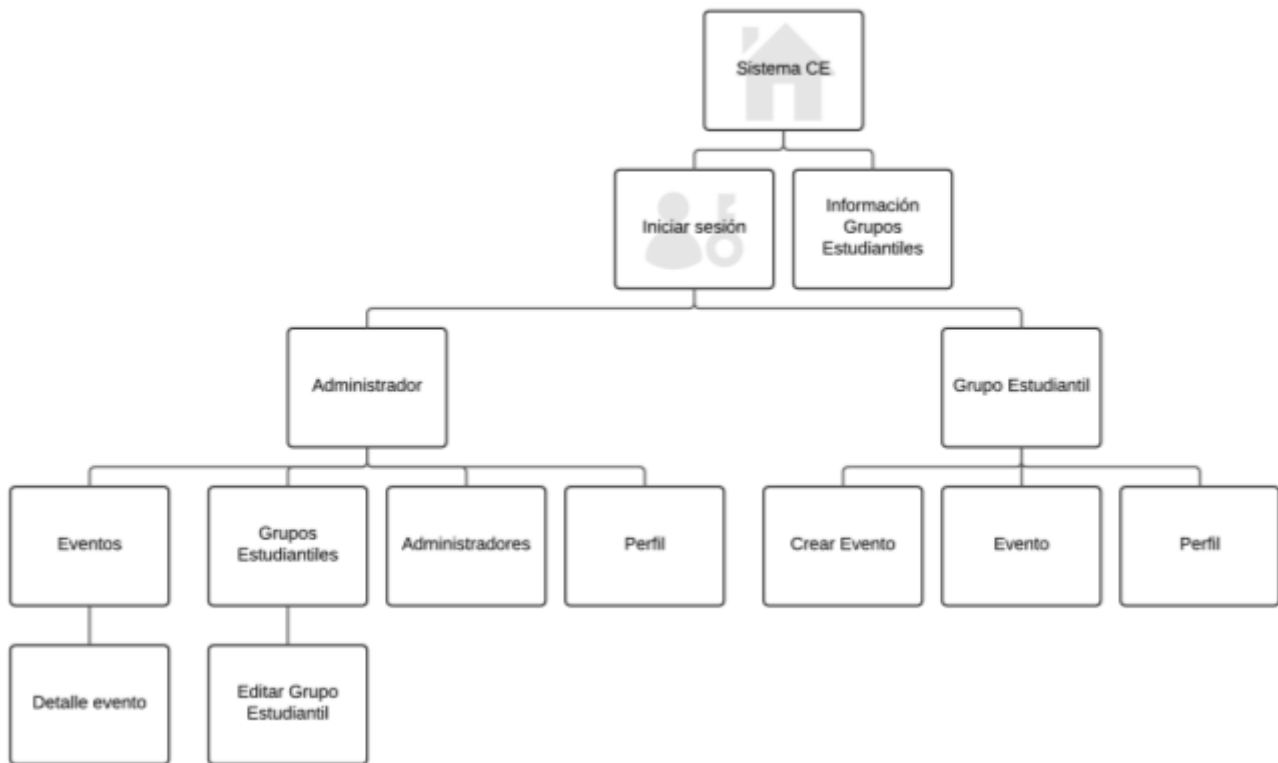
- Facilitar el proceso de administrar y planear eventos de grupos estudiantiles.
- Mejorar la experiencia al administrar y planear eventos de grupos estudiantiles.
- Reducir los conflictos y la mala comunicación entre alumnos y administradores.
- Mejorar el servicio que ofrecen los administradores de Centro Estudiantil al tener toda la información en una misma plataforma.

Asimismo, el proyecto será considerado exitoso y funcional si cumple con los siguientes aspectos:

- Hay dos tipos de usuarios: administrador y grupo estudiantil (alumno).
- El administrador tiene acceso a las siguientes áreas:
  - Eventos
  - Grupos Estudiantiles
  - Administración
  - Perfil
- Los grupos estudiantiles tienen acceso a las siguientes áreas:
  - Eventos
  - Perfil

## IV. Mapa del Sitio

La aplicación web será conformada por diversas secciones, por lo que el mapa del sitio para la misma será como se muestra a continuación:



### IV.I Contenido de la Aplicación Web

El contenido de la aplicación web se presenta en la siguiente lista mencionando los servicios clave de la misma:

- Información Grupos Estudiantiles: Contiene una visión general de lo que se hace en los Grupos Estudiantiles y otra información relevante.
- Iniciar sesión: En esta sección los usuarios podrán ingresar al sistema con sus diferentes cuentas y contraseñas.
- Administrador
  - Eventos: Contiene una lista con todos los eventos creados por los alumnos.
  - Detalle Evento: Contiene información relevante de un evento.
  - Grupos Estudiantiles: Tiene las funcionalidades de crear, editar o eliminar los diferentes grupos estudiantiles.
  - Editar Grupo Estudiantil: Ofrece la posibilidad de editar la información sobre un Grupo Estudiantil.
  - Administradores: Contiene una lista de los administradores registrados en el sistema.
  - Perfil: Permite al usuario ver la información relevante de su cuenta en el sistema.
- Grupo Estudiantil
  - Crear Evento: Permite que el alumno cree un evento en el sistema.
  - Evento: Ofrece la posibilidad al alumno de ver o modificar la información relevante de algún evento.
  - Perfil: Permite al usuario ver la información relevante de su cuenta en el sistema.

## ***V. Especificaciones Funcionales***

La aplicación web a desarrollar debe tener una buena experiencia de usuario que le permita a los grupos estudiantiles crear eventos mediante un proceso más sencillo, por lo que en la siguiente lista se describe lo que se espera experimentar en cada sección del sitio:

- Información Grupos Estudiantiles: Contiene una vista limpia en la cual se utilizan los colores de la oficina de Grupos Estudiantiles, al mismo tiempo que contiene diferentes secciones con la información que El Cliente desee incluir. Al entrar a la página debe aparecer la opción para poder iniciar sesión y si se avanza en la página se debe tener un botón para iniciar sesión en la esquina superior derecha.
- Iniciar sesión: La vista para iniciar sesión debe ser incluida en la página principal, por lo cual no debe cambiar a otra página.
- Administrador: Muestra los eventos registrados para la semana actual, así como la siguiente y los demás botones para acceder a los demás servicios para administradores.
  - Eventos: Contiene una lista de todos los eventos registrados con opciones de búsqueda y filtros sobre la misma.
  - Detalle Evento: Contiene una vista con los diferentes detalles pertinentes a un evento.
  - Grupos Estudiantiles: Muestra una lista de todos los diferentes Grupos Estudiantiles con opciones de búsqueda y filtros sobre la lista.
  - Editar Grupo Estudiantil: Contiene una vista con los diferentes detalles pertinentes al Grupo Estudiantil.
  - Administradores: Muestra una la lista con la información de los diferentes administradores del sistema.
  - Perfil: Contiene vista con la información del perfil.
- Grupo Estudiantil: Muestra los eventos registrados del grupo estudiantil y además incluye los botones para acceder al resto de los servicios para alumnos.
  - Crear Evento: Contiene una vista con una forma solicitando toda la información necesaria para registrar un evento.
  - Evento: Muestra una vista con toda la información del evento para ser revisada y/o editada.
  - Perfil: Contiene una vista con la información del perfil.

## ***VI. Especificaciones Técnicas***

Las especificaciones técnicas se muestran a continuación a través de los requerimientos funcionales y no funcionales, las especificaciones de los servidores y navegadores web, además de la estructura de la base de datos y los lenguajes de programación que serán utilizados para la realización de la aplicación web.

Para las cuestiones visuales y de diseño, mejor conocido como Frontend, el lenguaje de programación a utilizar será Materialize CSS en su Versión 0.98, de modo que se le pueda brindar a la página un ambiente moderno y responsivo.

Por otro lado, para el desarrollo de Backend se utilizará el lenguaje de Ruby on Rails en su Versión 4.2, por lo cual se podrá hacer uso de librerías conocidas como Gems.

### Requerimientos Funcionales

R.F.001. El sistema debe distinguir si es un alumno o un administrador quien accede al sistema.

R.F.002. El sistema debe desplegar un mensaje de error si un usuario intenta acceder al sistema con una contraseña equivocada.

R.F.003. El sistema debe desplegar un mensaje de error si un usuario intenta acceder al sistema con una cuenta que no existe.

R.F.004. El sistema debe permitir al usuario cambiar su contraseña.

- R.F.005. El sistema debe permitir al usuario cambiar el nombre de su perfil de usuario.
- R.F.006. El sistema debe desplegar a los administradores los eventos de la semana y la siguiente semana.
- R.F.007. El sistema debe ofrecer a los administradores la opción de ver la lista de eventos.
- R.F.008. El sistema debe ofrecer a los administradores la opción de ver la lista de Grupos Estudiantiles.
- R.F.009. El sistema debe ofrecer a los administradores la opción de ver la lista de administradores registrados.
- R.F.010. El sistema debe tener por cada evento la autorización pendiente de los departamentos de logística, finanzas, mercadotecnia y consejo.
- R.F.011. El sistema debe permitir a los administradores dar autorización por parte de un departamento a un evento.
- R.F.012. El sistema debe registrar el evento en el calendario solamente si tiene la autorización de todos los departamentos.
- R.F.013. El sistema debe notificar a los administradores cuando un evento sea aprobado por todos los departamentos.
- R.F.014. El sistema debe organizar todos los eventos en una lista.
- R.F.015. El sistema debe tener un mecanismo de filtros para localizar de manera rápida los eventos.
- R.F.016. El sistema debe tener un filtro que defina cuáles eventos están pendientes de autorizar por los diferentes departamentos.
- R.F.017. El sistema debe tener un mecanismo de búsqueda para localizar de manera rápida los eventos.
- R.F.018. El sistema debe utilizar los siguientes colores para distinguir entre los diferentes consejos y sus eventos en la lista de eventos:
- FEITESM - Azul
  - CCE - Rojo
  - CSA - Morado
  - CARE - Naranja
  - Consejo/Institucional - Gris
- R.F.019. El sistema debe permitir a los administradores ver los detalles de un evento desde la lista de eventos.
- R.F.020. El sistema debe registrar la fecha y usuario cuando un evento cambie de estatus o sea aprobado por algún departamento.
- R.F.021. El sistema debe tener a un evento en estatus de 'En Proceso' cuando se crea el evento.
- R.F.022. El sistema debe tener a un evento en estatus de 'Autorizado' cuando un evento es aprobado por todos los departamentos.
- R.F.023. El sistema debe tener a un evento en estatus de 'Cancelado' cuando un Grupo Estudiantil cancele un evento.
- R.F.024. El sistema debe tener un inventario actualizado del mobiliario disponible para un evento.
- R.F.025. El sistema debe de mantener actualizada la lista de inventario para un evento.

R.F.026. El sistema debe mostrar un mapa con las ubicaciones disponibles para tener un evento al momento de hacer el registro del mismo.

R.F.027. El sistema debe de mantener actualizada las ubicaciones disponibles para un evento.

R.F.028. El sistema debe tener una sección dentro de un evento en la cual se puedan incluir detalles de comida y estacionamiento que les sirva de referencia a esos departamentos externos.

R.F.029. El sistema debe permitir a los administradores crear grupos estudiantiles.

R.F.030. El sistema debe permitir a los administradores eliminar grupos estudiantiles.

R.F.031. El sistema debe permitir a los administradores deshabilitar temporalmente algún grupo estudiantil.

R.F.032. El sistema debe permitir a los administradores editar la información de los grupos estudiantiles.

R.F.033. El sistema debe permitir a los administradores crear otras cuentas de administradores.

R.F.034. El sistema debe permitir a los administradores eliminar otras cuentas de administradores.

R.F.035. El sistema debe permitir a los administradores restablecer la contraseña de otras cuentas de administradores.

R.F.036. El sistema debe crear un folio único para cada evento que sea registrado.

R.F.037. El sistema debe ofrecer al alumno la opción de crear un evento.

R.F.038. El sistema debe permitir al alumno subir archivos en las secciones que lo requieran.

R.F.039. El sistema no debe permitir la creación de nuevos eventos a alumnos sin 3 días hábiles antes de la fecha deseada.

R.F.040. El sistema debe actualizar la lista de eventos con cada cambio realizado a un evento.

R.F.041. El sistema debe permitir a los alumnos registrar a los coordinadores y colaboradores de sus eventos.

R.F.042. El sistema debe ofrecer al alumno la opción de modificar un evento ya existente.

#### Requerimientos Funcionales a NO desarrollar en este Proyecto – A implementar después del semestre

R.F.043. El sistema debe enviar correos notificando a alumnos y administradores de los siguientes eventos:

- 30 días faltantes para un evento
- 14 días faltantes para un evento
- 7 días faltantes para un evento
- Cambio en algún estatus
- Autorización del evento
- Cambio de mesa directiva

R.F.044. El sistema debe tener una vista estilo calendario para los eventos.

R.F.045. El sistema debe permitir a los administradores mandar una alerta a los organizadores de un evento si tienen algún pendiente por revisar.

R.F.046. El sistema debe permitir a los administradores dar autorización a un presidente de un grupo estudiantil para que éste pueda registrar o editar a su mesa directiva.

R.F.047. El sistema debe permitir a los administradores revocar autorización a un presidente de un grupo estudiantil para que éste pueda registrar o editar a su mesa directiva.

R.F.048. El sistema solamente debe dar acceso a los alumnos que tengan los puestos de Presidente, Vicepresidente y Coordinador General en sus mesas directivas.

R.F.049. El sistema debe permitir a los alumnos autorizados el editar la información de su mesa directiva.

R.F.050. El sistema debe realizar una encuesta al momento de registrar algún cambio en una mesa directiva para obtener información sobre el porqué del cambio

R.F.051. El sistema debe permitir a los administradores el autorizar un cambio en una mesa directiva

R.F.052. El sistema debe generar un directorio de los alumnos inscritos en el periodo actual.

#### Requerimientos No Funcionales:

R.N.F.001. El sistema debe tener una interfaz moderna que cumpla con los estándares de diseño web y buena experiencia del usuario.

R.N.F.002. El sistema será capaz de trabajar correctamente en los navegadores Google Chrome V.55 o superior, Internet Explorer V.11 o superior, Firefox V.50 o superior y Safari V.5.1 o superior.

R.N.F.003. El sistema debe tener almacenado sus contenidos en el servidor Heroku Cloud Application.

R.N.F.004. El sistema debe mantener constante comunicación con la base de datos de Ruby on Rails.

R.N.F.005. El sistema debe incluir el logotipo del Tec de Monterrey en sus vistas.

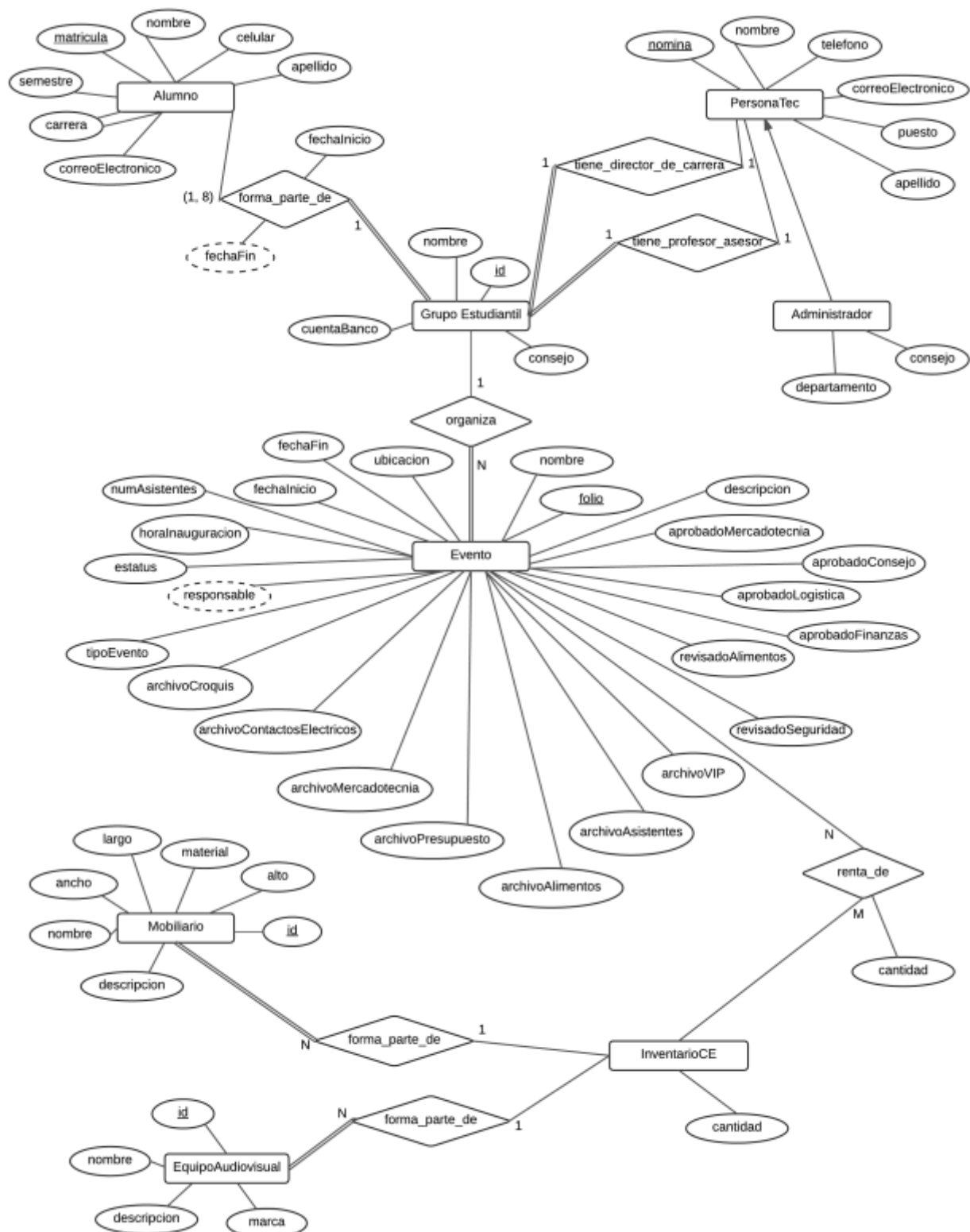
R.N.F.005. El sistema debe ser desarrollada en español.

R.N.F.004. El sistema debe ser capaz de crear varios eventos sin afectar la integridad del sistema.



## Diagrama Base de Datos

A continuación, se muestra la distribución de los elementos de la base de datos, por lo que el siguiente diagrama de Entidad-Relación se tomará como fundamento para la formación de las tablas de la ya mencionada base de datos.



## VII. Plan de Contenido

En el desarrollo del proyecto la participación del Cliente será importante, ya que además de proveer retroalimentación constante, será el encargado de validar que el diseño de las vistas sea el apropiado al contener los colores usados por el Tec de Monterrey, además de que proporcionará la información a aparecer en cada sección de la aplicación.

### VII.I. Control de Versiones

MDJ está conformado por cuatro personas, por lo que para tener un control de versiones se utilizará la herramienta de GitHub, en la cual cada miembro subirá el código que realice de modo que todos los integrantes puedan trabajar en el proyecto de manera sincronizada.

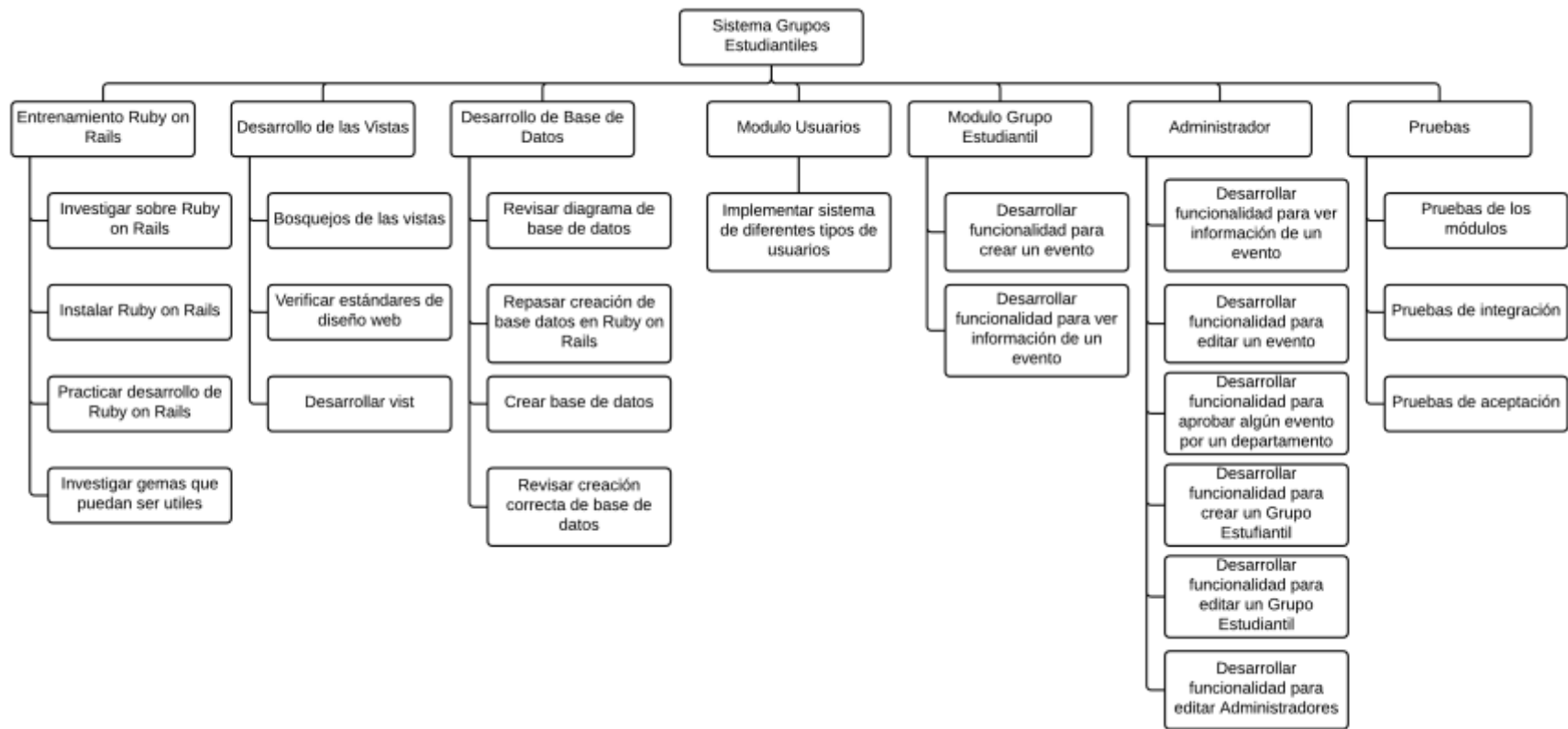
Al mismo tiempo, el usar una herramienta como GitHub dará paso a que se trabaje de manera conjunta de forma paralela, además de que se tendrá la oportunidad de recuperar versiones anteriores en caso de que se presente un problema importante.

La liga del repositorio de GitHub donde estará almacenado el código desarrollado es la siguiente: <https://github.com/mauro94/SistemaGruposEstudiantiles>

### VII.II. Responsabilidades

Los miembros de MDJ trabajarán de manera conjunta con El Cliente para obtener un producto final satisfactorio, por lo que enseguida se enlistan las responsabilidades que cada integrante tendrá en el Proyecto:

Integrante del Equipo	Responsabilidad
Martha Benavides	Desarrolladora Web / Front End
Mauro Amarante	Desarrollador Web / Front End / Líder del Equipo
David Martínez	Desarrollador Web
Jesús Navarro	Desarrollador Web
El Cliente (Alejandra Garza González)	Proveer información a incluir en la página. Participar en las pruebas. Revisar avances. Dar opinión del diseño. Validar requerimientos.

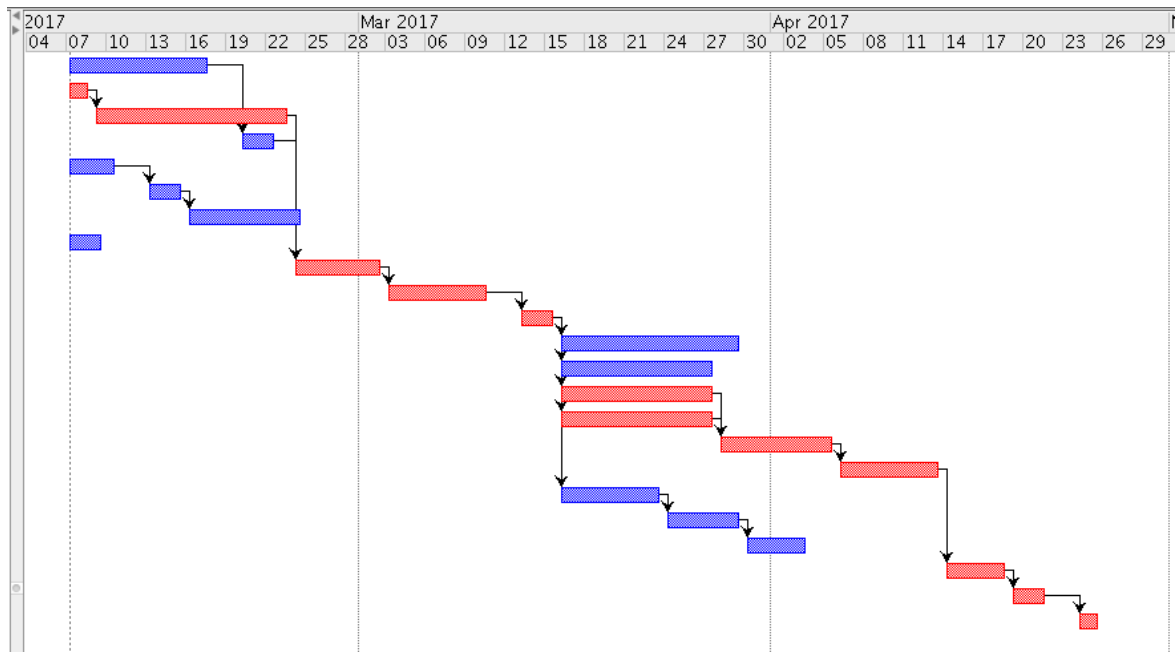


## VII.IV. Estimación de Tiempo

Código	Actividad	Días			
		Duración Optimista	Duración Esperada	Duración Pesimista	Duración PERT
1.1	Investigar sobre Ruby on Rails	5	8	14	9
1.2	Instalar Ruby on Rails	1	2	4	2
1.3	Practicar desarrollo de Ruby on Rails	5	10	20	11
1.4	Investigar gemas que puedan ser útiles	2	3	5	3
2.1	Bosquejos de las vistas	2	4	7	4
2.2	Verificar estándares de diseño web	1	3	5	3
2.3	Desarrollar vistas	5	7	10	7
3.1	Revisar diagrama de base de datos	1	3	5	3
3.2	Repasar creación de base datos en Ruby on Rails	2	5	8	5
3.3	Crear base de datos	3	5	14	6
3.4	Revisar creación correcta de base de datos	1	3	7	3
4.1	Implementar sistema de diferentes tipos de usuarios	5	10	15	10
5.1	Desarrollar funcionalidad para crear un evento	5	7	14	8
5.2	Desarrollar funcionalidad para ver información de un evento	5	8	10	8
6.1	Desarrollar funcionalidad para ver información de un evento	5	8	10	8
6.2	Desarrollar funcionalidad para editar un evento	5	7	10	7
6.3	Desarrollar funcionalidad para aprobar algún evento por un departamento	3	5	10	6
6.4	Desarrollar funcionalidad para crear un Grupo Estudiantil	3	5	10	6
6.5	Desarrollar funcionalidad para editar un Grupo Estudiantil	3	4	7	4
6.6	Desarrollar funcionalidad para editar Administradores	2	3	5	3
7.1	Pruebas de los módulos	2	3	5	3
7.2	Pruebas de integración	2	3	5	3
7.3	Pruebas de aceptación	1	2	3	2
	<b>TOTAL</b>	69	118	203	124

## VII.V. Diagrama de Gantt

	Name	Duration	Start	Finish	Predecessors
1	Investigar sobre Ruby on Rails	9 days	2/7/17 8:00 AM	2/17/17 5:00 PM	
2	Instalar Ruby on Rails	2 days	2/7/17 8:00 AM	2/8/17 5:00 PM	
3	Practicar desarrollo de Ruby on Rails	11 days	2/9/17 8:00 AM	2/23/17 5:00 PM	2
4	Investigar gemas que puedan ser útiles	3 days	2/20/17 8:00 AM	2/22/17 5:00 PM	1
5	Bosquejos de las vistas	4 days	2/7/17 8:00 AM	2/10/17 5:00 PM	
6	Verificar estándares de diseño web	3 days	2/13/17 8:00 AM	2/15/17 5:00 PM	5
7	Desarrollar vistas	7 days	2/16/17 8:00 AM	2/24/17 5:00 PM	6
8	Revisar diagrama de base de datos	3 days	2/7/17 8:00 AM	2/9/17 5:00 PM	
9	Repasar creación de base datos en Ruby on Rails	5 days	2/24/17 8:00 AM	3/2/17 5:00 PM	3;4
10	Crear base de datos	6 days	3/3/17 8:00 AM	3/10/17 5:00 PM	9
11	Revisar creación correcta de base de datos	3 days	3/13/17 8:00 AM	3/15/17 5:00 PM	10
12	Implementar sistema de diferentes tipos de usuarios	10 days	3/16/17 8:00 AM	3/29/17 5:00 PM	11
13	Desarrollar funcionalidad para crear un evento	8 days	3/16/17 8:00 AM	3/27/17 5:00 PM	11
14	Desarrollar funcionalidad para ver información de un evento	8 days	3/16/17 8:00 AM	3/27/17 5:00 PM	11
15	Desarrollar funcionalidad para ver información de un evento	8 days	3/16/17 8:00 AM	3/27/17 5:00 PM	11
16	Desarrollar funcionalidad para editar un evento	7 days	3/28/17 8:00 AM	4/5/17 5:00 PM	14;15
17	Desarrollar funcionalidad para aprobar algún evento por un c	6 days	4/6/17 8:00 AM	4/13/17 5:00 PM	16
18	Desarrollar funcionalidad para crear un Grupo Estudiantil	6 days	3/16/17 8:00 AM	3/23/17 5:00 PM	11
19	Desarrollar funcionalidad para editar un Grupo Estudiantil	4 days	3/24/17 8:00 AM	3/29/17 5:00 PM	18
20	Desarrollar funcionalidad para editar Administradores	3 days	3/30/17 8:00 AM	4/3/17 5:00 PM	19
21	Pruebas de los módulos	3 days	4/14/17 8:00 AM	4/18/17 5:00 PM	17
22	Pruebas de integración	3 days	4/19/17 8:00 AM	4/21/17 5:00 PM	21
23	Pruebas de aceptación	2 days	4/24/17 8:00 AM	4/25/17 5:00 PM	22



## VII.VI. Metas Intermedias y Reuniones

Los integrantes de MDJ tendrán reuniones semanales de manera presencial o remota vía Google Hangouts para compartir avances y resolver los problemas que se presenten.

De igual manera, con El Cliente se realizarán reuniones cada semana siempre y cuando haya por lo menos un miembro del equipo disponible para ir a mostrarle los avances.

Dado lo anterior, el líder de MDJ será el encargado de monitorear que el trabajo del proyecto esté avanzando de acuerdo a las fechas de los Sprints declarados a continuación:

Prioridad	<b>Sprint 1: Feb 1 - 6</b>
1	Instalar Ruby on Rails
1	Investigar sobre Ruby on Rails
1	Practicar desarrollo de Ruby on Rails
	<b>Sprint 2: Feb 7 - 13</b>
2	Investigar sobre Ruby on Rails
2	Practicar desarrollo de Ruby on Rails
1	Bosquejos de las vistas
1	Revisar diagrama de base de datos
	<b>Sprint 3: Feb 14 - 20</b>
2	Verificar estándares de diseño web
1	Desarrollar vistas
3	Practicar desarrollo de Ruby on Rails
	<b>Sprint 4: Feb 21 - 27</b>
2	Investigar gemas que puedan ser útiles
1	Desarrollar vistas
1	Repasar creación de base datos en Ruby on Rails
	<b>Sprint 5: Feb 28 - Mar 6</b>
1	Crear base de datos
	<b>Sprint 6: Mar 7 - 13</b>
1	Crear base de datos
2	Revisar creación correcta de base de datos
	<b>Sprint 7: Mar 14 - 20</b>
1	Implementar sistema de diferentes tipos de usuarios
1	Desarrollar funcionalidad para crear un evento
2	Desarrollar funcionalidad para ver información de un evento (vista administrador)
2	Desarrollar funcionalidad para ver información de un evento (vista alumno)
3	Desarrollar funcionalidad para crear un Grupo Estudiantil
	<b>Sprint 8: Mar 21 - 27</b>
2	Implementar sistema de diferentes tipos de usuarios
2	Desarrollar funcionalidad para crear un evento
1	Desarrollar funcionalidad para ver información de un evento (vista administrador)
1	Desarrollar funcionalidad para ver información de un evento (vista alumno)
1	Desarrollar funcionalidad para crear un Grupo Estudiantil
	<b>Sprint 9: Mar 28 - Abr 3</b>
1	Desarrollar funcionalidad para editar un evento
1	Desarrollar funcionalidad para editar un Grupo Estudiantil
1	Desarrollar funcionalidad para editar Administradores
	<b>Sprint 10: Abr 4 - 10</b>
1	Desarrollar funcionalidad para aprobar algún evento por un departamento
	<b>Sprint 11: Abr 11 - 17</b>
1	Pruebas de los módulos
	<b>Sprint 12: Abr 18 - 24</b>
1	Pruebas de integración
	<b>Sprint 13: Abr 25 - May 2</b>
1	Pruebas de aceptación

## ***VIII. Plan de Pruebas***

En la cuestión de pruebas, para la parte del diseño y la vista de la aplicación web, se tomará la opinión del Cliente, de los cuatro miembros de MDJ y de otras dos personas externas, de modo que se obtenga una retroalimentación de diversos puntos de vista para poner hacer mejoras.

Entre las pruebas de funcionalidad se encuentran:

- Se debe asegurar que la sección de LogIn valide que no se pueda acceder con datos incorrectos o con elementos vacíos.
- Se debe asegurar que la sección de LogIn pueda distinguir entre si el usuario ingresado pertenece a un administrador o a un alumno.
- Se debe revisar que todos los botones tengan funcionalidad, de manera que no haya botones innecesarios. Al mismo tiempo, cada botón debe realizar la función correcta.
- Se debe asegurar que tanto la vista del administrador, así como la del alumno, cumplan con los requerimientos del Cliente, siendo diferentes ambas.
- El Cliente debe poder navegar por la página y realizar un proceso normal (al que está acostumbrado) de manera que pueda identificar posibles errores o áreas de oportunidad.
- Se debe verificar que la información extraída de la base de datos sea presentada en el sitio manera adecuada y que, además, sea la correcta.

En caso de que lleguen a surgir múltiples detalles, se buscará darle una prioridad a cada detalle dependiendo del orden de los requerimientos ya establecidos en el calendario. El listado de pruebas ira incrementando conforme el proyecto vaya siendo realizado.

Asimismo, cabe destacar que se harán pruebas de módulos, de integración y de aceptación.

## ***IX. Apéndices***

En esta sección se mostrará información relevante sobre el proyecto, tal como lo son las notas hechas en las juntas, así como el prototipo y el glosario de términos.

### **IX.I. Notas de las Juntas**

*Fecha: Enero 20, 2017*

#### **Asistentes**

- Martha (Equipo)
- Mauro (Equipo)
- David (Equipo)
- Alejandra (Coordinadora CCE)
- Marvin (Coordinador Logística)
- Paulina (Coordinadora CARE)
- Marvin (Coordinador CSA)

#### **Notas:**

- El sistema actual está muy desactualizado, tedioso y hacen trabajo doble.
- Se necesita página con vista de administrador y vista de alumno.
  - Los administradores pueden validar eventos.
  - Agregar matrículas para autorizar alumnos (se quiere que se cambie esto a agarrarlos desde una base de datos).
- Ocupan que cada semestre se borre toda la información en lugar que tener que desactivar uno por uno.
- Es muy importante que cada reservación tenga un folio (código) para poder hacer un seguimiento.
- Los alumnos pueden cancelar las peticiones que hayan hecho.
- Se pueden hacer las solicitudes con tres días hábiles.
- Importante: Requieren que la información se actualice de forma automática (para evitar estar copiando la misma información en el drive).
- Pestaña de Recursos
  - Mandar notificaciones (correos) a alumnos y encargados (mediadores) del estatus del evento.
  - 30 días
  - Tener un inventario del mobiliario actualizado
  - Tener mapas y croquis
  - Tener una actualización de los costos por mobiliario
- Permitir el registro en línea de mesas directivas, de modo que los estudiantes lo soliciten y los encargados lo validen para que se le dé el acceso automático a la página.
- Importante: Que los alumnos puedan ver los espacios disponibles.
- Si no tiene TODAS las autorizaciones de todas las áreas, no se debe agendar el evento.
- Si tiene TODAS las autorizaciones, el evento deberá agendarse de forma automática en el calendario.
- Si logística valida el evento, hay que apartarlo (aunque después se cancele por no haber sido autorizado por otras áreas).
- El permiso del área de finanzas es el más importante.
- Se quiere una barra de búsqueda general.
- Se requieren de filtros para que los encargados de cada área vean cuales eventos necesitan (faltan) dé su visto bueno.
- Hay que hacer algo porque al cambiar cada año la información del año pasado se pierde y tiene que ser buscada mediante los filtros (eventos que fueron reservados el año pasado y su folio empieza con 2016##).
- Si les compete, otras áreas (alimentos, seguridad) pueden meterse a poner retroalimentación.



## I.X.II. Conceptos de Diseño

En esta sección se muestran los diseños que se tienen para la distribución gráfica de cada vista de la aplicación web:

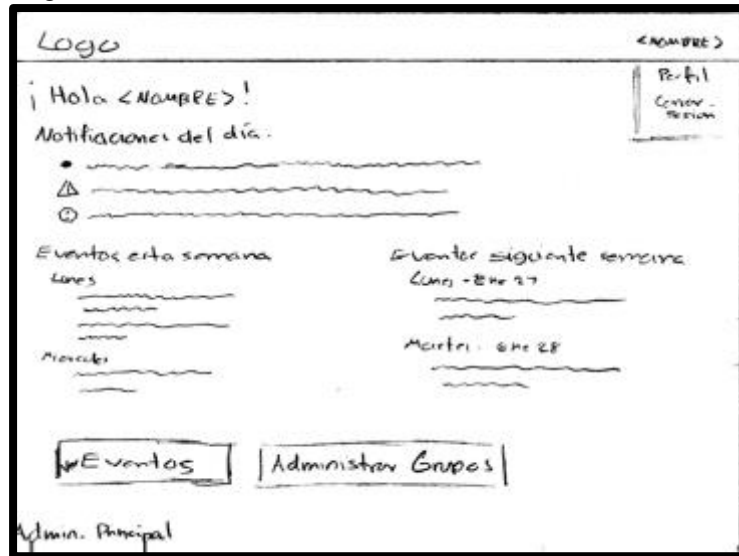
- Pantalla de LogIn



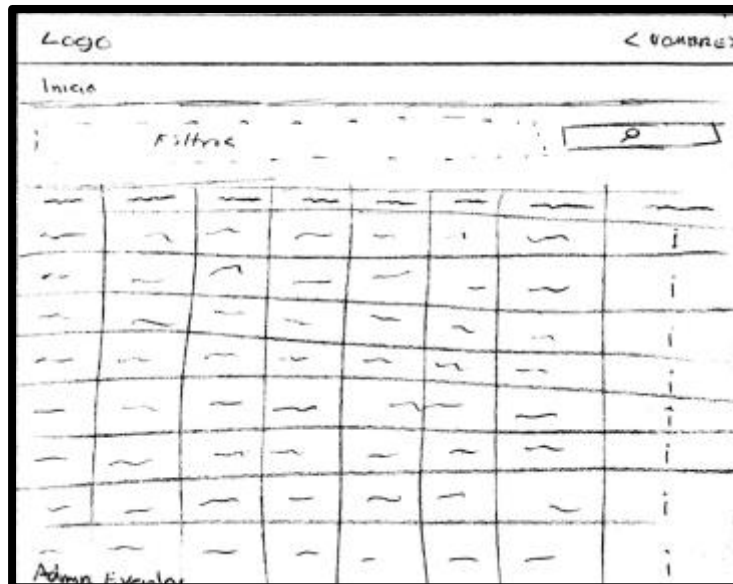
- Vista de Información sobre los Grupos Estudiantiles



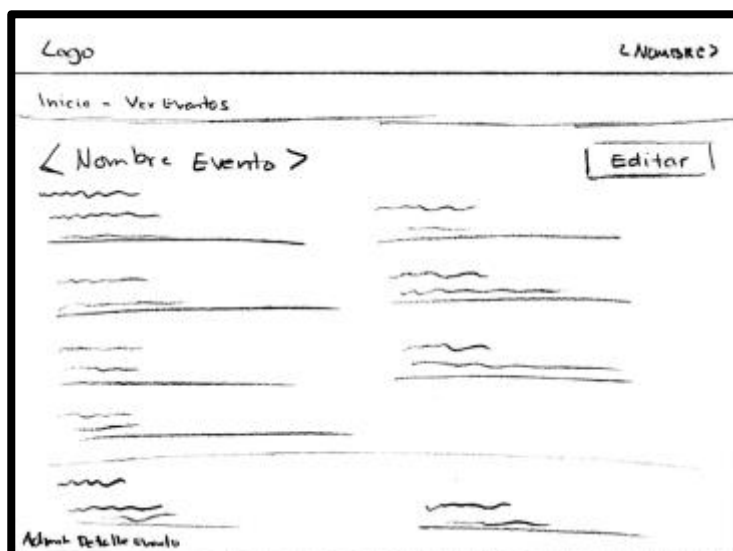
- Página Inicial en vista de Administrador



- Vista que le muestra al administrador la lista de eventos registrados



- Vista que muestra los detalles del evento



- Vista que permite la opción de administrar Grupos Estudiantiles

Logo

Inicio - Administrar Grupos

< Nombre Grupo >

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Guardar Cambios | Eliminar

Admin Editar Grupo

- Vista Administrador que permite agregar nuevos administradores

Logo

Inicio

Crear Administrador

< Nombre >

_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

Admin Administrador

- Vista Alumno que permite crear un nuevo evento.

Logo

i Hola < NOMBRE > !

Anuncios:

\_\_\_\_\_

Crear Evento

Eventos:

Fecha	Nombre	Responsable	Editar
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Perfil

Conectar

Salir

### I.X.III. Prototipo

Para mostrarle un funcionamiento general al Cliente de la aplicación web, se tomarán los dibujos realizados en la sección de diseño y se colocarán en la herramienta Proto.IO (<https://proto.io/>), en la cual El Cliente podrá navegar a través de los dibujos como si fueran las pantallas del sistema, lo que dará paso a que El Cliente tenga una visión general de cómo funcionará la aplicación y se vaya familiarizando con ella.

De igual manera, al Cliente se le brindará la liga de GitHub, con la cual podrá monitorear el avance de la página Web.

### I.X.IV. Suposiciones

Esta sección no aplica debido al hecho de que no hay suposiciones.

### I.X.V. Glosario de Términos

Hay algunas palabras técnicas que pudieran causarle confusión al Cliente, por lo que a continuación se describen éstas:

- *Base de Datos*: Sistema que almacena información pertinente del proyecto para su futura manipulación.
- *Navegador Web*: Programa que permite el acceso a Internet.
- *Frontend*: Parte del sistema perteneciente a la interfaz, por lo cual es la parte que entra en contacto con el usuario.
- *Backend*: Parte de sistema que procesa la información ingresada por el usuario e implementa la funcionalidad.

## X. Firmas del Cliente y el Líder del Equipo

Con la firma de este documento, MDJ se compromete a mantener estricta confidencialidad respecto a la información recibida por parte de El Cliente y utilizar ésta solamente para los fines aquí establecidos. Asimismo, al finalizar el proyecto, MDJ se compromete a devolver al Cliente cualquier documento, clave o cuenta de acceso que le sea proporcionada durante el desarrollo del mismo.

Por su parte, El Cliente se compromete a proveer a MDJ la información acordada, necesaria para la realización exitosa de este proyecto, así como a darle seguimiento al trabajo desarrollado y estar disponible para cualquier duda o aclaración que surja en el transcurso del proyecto.

  
\_\_\_\_\_  
Alejandra Garza González / Cliente  
\_\_\_\_\_  
Mauro Amarante Esparza / Líder del Equipo