

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# DOCUMENTO DE DISEÑO WEB

**Nombre del proyecto:**  
**Organización**  
**Fecha:**  
**Versión:**

**Relax UC**  
**Salud Estudiantil UC**  
**27/10/16**  
**0.1**

## **Historia del Documento**

<b>Versión</b>	<b>Fecha</b>	<b>Autor(es)</b>	<b>Razón del Cambio</b>
<b>0.1</b>	15/10/2016	Esteban Meza	Completar descripción general
<b>0.2</b>	25/10/2016	Diego Sinay	Completar Vista Lógica
<b>1.0</b>	27/10/2016	Esteban Meza	Completar Vista Deployment
<b>1.0</b>	27/10/2016	Faustino Marañón	Completar Vista de Implementación
<b>1.0</b>	27/10/2016	Francisco Garri	Completar Vista de Datos
<b>1.0</b>	27/10/2016	Felipe Baitelman	Corregir Vista Lógica
<b>2.0</b>	21/11/2016	Felipe Baitelman	Actualizar documento y agregar vistas
<b>3.0</b>	02/12/2016	Felipe Baitelman	Corregir detalles y preparar documento para entrega final.

## Equipo de Desarrollo

Nombres y Apellidos	Rol	Contacto
<b>Felipe Baitelman</b>	Administrador del Proyecto	fjbaitelman@uc.cl (+569) 8889 4203
<b>Camilo Álvarez</b>	Desarrollador	cealvarez@uc.cl (+569) 8252 8100
<b>Javier Chicao</b>	Desarrollador	jachicao@uc.cl (+569) 9489 2643
<b>Patricio Díaz</b>	Desarrollador	pndiaz1@uc.cl (+569) 8841 9241
<b>Francisco Garri</b>	Analista	fagarri@uc.cl (+569) 9650 8858
<b>Faustino Marañon</b>	Desarrollador	famarano@uc.cl (+569) 7709 8312
<b>Esteban Meza</b>	Analista	ejmeza@uc.cl (+569) 8260 4162
<b>Rodrigo Saavedra</b>	Desarrollador	rpsaavedra@uc.cl (+569) 5723 9913
<b>Diego Sinay</b>	Arquitecto/ Desarrollador	dsinay@uc.cl (+569) 9078 6585
<b>Lukas Zorich</b>	Arquitecto/ Desarrollador	lukas.zorich@gmail.com (+569) 7165 4542

## Contraparte del Proyecto

Nombres y Apellidos	Rol	Contacto
<b>María Paz Jana</b>	Coordinadora	<a href="mailto:mpjana@uc.cl">mpjana@uc.cl</a> 2 2354 1046
<b>María Trinidad Barriga</b>	Psicóloga	<a href="mailto:mtbarriga@uc.cl">mtbarriga@uc.cl</a> 2 2354 1046
<b>María Soledad Zuzulich</b>	Directora Salud Estudiantil	<a href="mailto:mzuzulich@uc.cl">mzuzulich@uc.cl</a> 2 2354 5402

## Tabla de Contenidos

Historia del Documento .....	i
Equipo de Desarrollo .....	ii
Contraparte del Proyecto .....	ii
1. Descripción General .....	1
1.1 Propósito.....	1
1.2 Alcance .....	1
1.3 Visión General .....	2
1.4 Definiciones, Acrónimos y Abreviaturas .....	2
1.5 Referencias.....	3
2. Vista Lógica .....	4
2.1 Arquitectura del Sistema.....	4
2.2 Arquitectura Lógica .....	5
2.2.1 Escenarios .....	5
2.2.2 Admin.....	7
2.2.3 Media.....	7
2.2.4 Pages.....	8
2.1.2.5 API.....	9
2.3 Arquitectura de los Módulos .....	10
2.1.3.1 Interfaz de Usuario .....	10
2.1.3.1.1 Pages.....	11
2.1.3.1.2 Media.....	13
2.1.3.1.4 Calendar .....	18
2.1.3.1.5 Admin.....	20
3. Vista de Implementación.....	21
3.1 Estructura de la Aplicación.....	21
4. Vista de Datos.....	22
4.1 Modelo de Datos.....	22
4.2 Distribución .....	22
5. Vista de Deployment .....	23
5.1 Arquitectura Técnica.....	23
5.2 Tecnología requerida .....	24
5.3 Deployment.....	24
6. Anexos .....	25

# 1. Descripción General

## 1.1 Propósito

El sistema es una aplicación web y móvil que tiene como objetivo ser una herramienta que permita al usuario acceder a servicios similares a los entregados en el Servicio de Salud Estudiantil sin la necesidad de asistir a un centro físico. La idea es que la aplicación fomente el autocuidado del usuario a lo largo del tiempo llevándolo a un estilo de vida más saludable. La aplicación tiene también como objetivo recolectar datos sobre los niveles de estrés y ansiedad en la comunidad UC para orientar los esfuerzos de la Dirección de Salud Estudiantil. Finalmente, la aplicación debe permitir tomar horas online en el Servicio de Salud Estudiantil, reemplazando al sistema presencial que se usa actualmente.

Los contenidos principales de la aplicación son imaginerías, secuencias de imágenes, ejercicios de respiración, sonidos relajantes.

- **Imaginerías:** Son audios con instrucciones o historias relajantes. Pueden ir acompañados de sonidos e imágenes ambientales.
- **Secuencias de imágenes:** Ejercicios de yoga divididos en imágenes explicativas. El usuario debe poder seguir el ejercicio usando solo la aplicación.
- **Ejercicios de respiración:** Ejercicios fáciles de seguir de respiración cuadrada, guiada, diafragmática y alternada.
- **Sonidos:** Audio ambiental relajante. Puede ir acompañado de imágenes.

## 1.2 Alcance

Los entregables se dividen en dos: una aplicación web y una aplicación Android que consisten en lo siguiente:

- Un sistema de reproducción de sonidos e imágenes ambientales de fácil acceso
- Un sistema que muestre imágenes y reproduzca audio prolongado para poder ambientarse
- Ejercicios para guiar la respiración
- Realidad Virtual (video de relajación 360º con audio)
- Manejo dinámico del contenido
- Calendario para manejo de horarios médicos
- Estadísticas sobre la visualización de contenido

A la vez el equipo debe entregar la documentación adecuada a los clientes. Estos consisten en un manual de implementación y uso de las soluciones para que puedan ser utilizadas.

Inicialmente, el cliente quería una aplicación móvil disponible para Android y para iOS, pero en las negociaciones iniciales se determinó que, dadas las características específicas requeridas por el cliente (realidad virtual), será necesario desarrollar una aplicación nativa. Por los recursos

de tiempo disponibles, se decidió solo desarrollar solo en un sistema. Dado el mayor conocimiento del sistema operativo de Google, se concluyó desarrollar la aplicación nativa para Android y posponer para una próxima etapa la presencia en la plataforma iOS.

### 1.3 Visión General

El proyecto consiste en aplicaciones web, móvil y realidad virtual (VR) desarrolladas en conjunto con la Dirección de Salud Estudiantil de la Pontificia Universidad Católica de Chile. El objetivo principal es ayudar a la comunidad UC con los crecientes problemas de ansiedad y estrés. La aplicación web consta de dos partes, usuario y administrador, mientras que móvil es exclusivamente para usuarios. Por último, la aplicación VR está pensada para ser usada en un centro de atención que aún no se construye.

El servicio principal para los usuarios es brindar una plataforma de contenido relajante de fácil acceso y en español. El sistema incluye videos y sonidos de naturaleza, programas de relajación, secuencias de imágenes de yoga, ejercicios de respiración guiada, imaginerías y un sistema de tomas de hora en los centro de salud UC. Además, los usuarios pueden acceder a evaluaciones que miden su nivel de ansiedad, en base a los cuales recibirán recomendaciones para mejorar su situación. En la aplicación VR se podrán visualizar videos 360, incluyendo audios complementarios.

La sección de administrador le permite subir, editar y borrar contenidos de los siguientes tipos: imaginerías, secuencias de imágenes y videos. El administrador también puede acceder a diversas estadísticas sobre la aplicación, como número de usuarios, reproducciones de cada contenido y promedios de los resultados de las evaluaciones psicológicas. Finalmente, el administrador puede editar el calendario de actividades, agregando las fechas y eventos correspondientes. Los usuarios podrán reservar horas según lo subido por el administrador. De esta manera, se facilita el proceso de toma de horas y los administradores pueden analizar las estadísticas de manera rápida para hacer estudios y sacar conclusiones.

### 1.4 Definiciones, Acrónimos y Abreviaturas

- API: Interfaz de programación de aplicaciones.
- Back-end: Capa de acceso a datos de la aplicación.
- Botón SOS: Botón de fácil acceso que lleva directamente a un ejercicio de relajación.
- Calendario de actividades: Calendario que contiene horas que los usuarios pueden reservar.
- Examen de ansiedad: Una serie de preguntas con el objetivo de entregar una evaluación del nivel de estrés del usuario.
- Framework: Estructura de capas que indica que programas pueden o deben utilizarse y como deben interrelacionarse.
- Front-end: Capa de presentación de la aplicación. Involucra interacciones con el Usuario.
- Historia de Usuario: representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario.

- **Imaginería:** Audios de relajación. Pueden ir acompañados de videos o imágenes.
- **Miembro UC:** Persona que pertenece a la comunidad de la Pontificia Universidad Católica de Chile.
- **MVC:** Patrón de arquitectura que separa la aplicación en tres componentes lógicos: Modelo, Vista y Controlador.
- **Naturaleza:** Videos de naturaleza acompañados de sonidos relajantes.
- **Perfil de usuario:** Cuenta con la cual un usuario puede acceder a las funcionalidades de la aplicación.
- **Plan de ejercicios:** Una serie de ejercicios predeterminados para seguir a lo largo de un determinado tiempo.
- **Respiración alternada:** Similar a la respiración diafragmática, pero se alterna la fosa nasal usada en cada inhalación y respiración.
- **Respiración cuadrada:** Ejercicio en el que se inhala, retiene, exhala y retiene en cuatro tiempos cada uno. Para guiar se usa un cuadrado que indica el progreso.
- **Respiración diafragmática:** Ejercicio de respiración en el que se exhala en el doble de tiempo que en el que se inhala. Tiene 3 niveles que dependen del tiempo de inhalación y exhalación.
- **Respiración guiada:** Ejercicio que le indica al usuario cuando inhalar y cuando respirar.
- **RoR:** Ruby on Rails. RoR es un framework para desarrollar aplicaciones web en el lenguaje Ruby. Fue utilizado para el desarrollo de la aplicación web.
- **Ruby:** Lenguaje de programación.
- **Secuencias de imágenes:** Series de imágenes explicativas de ejercicios de yoga.

## 1.5 Referencias

1. Plantilla Diseño IIC2154 v. 1.4

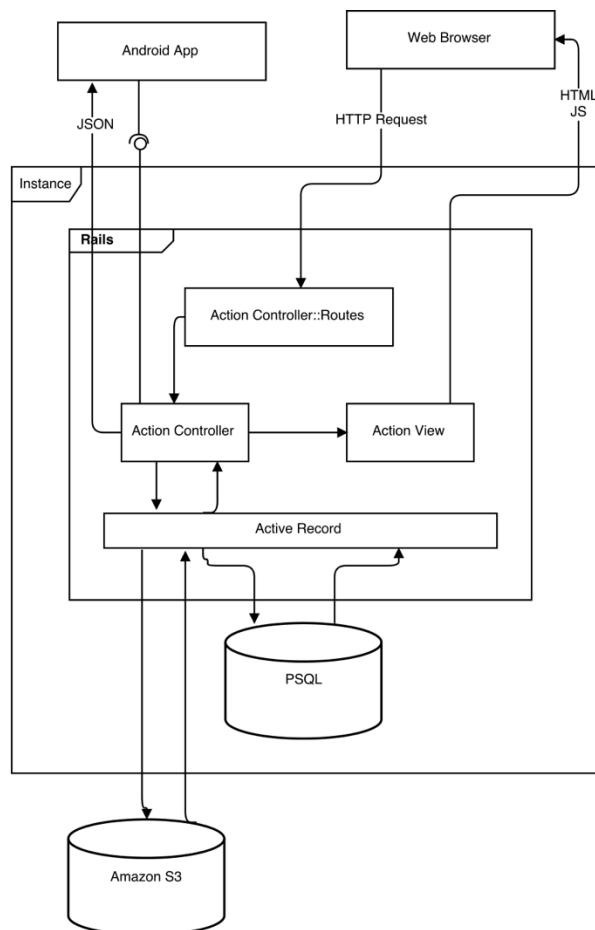
## 2. Vista Lógica

Esta vista presenta 3 niveles de arquitectura. El primer nivel corresponde al patrón arquitectónico, el cual sigue un modelo MVC. El segundo nivel corresponde al nivel lógico, donde se especifican los módulos más generales de la aplicación. Esta capa se puede profundizar, incluyendo en ella la API, las principales funcionalidades y los distintos modelos del sistema. En este nivel se utilizan diagramas de paquetes. Por último, existe una tercera capa más atómica que se centra en las historias de usuario y el modelo de navegación del sistema. Se utilizan diagramas de flujo.

### 2.1 Arquitectura del Sistema

En el primer nivel el enfoque se centra en el patrón arquitectónico de la aplicación. Esto incluye el modelo MVC de RoR, la base de datos que se utiliza y servidor donde se monta la aplicación. Se utiliza un modelo abstracto de la arquitectura general del sistema.

Diagrama 2.1: Arquitectura general del sistema

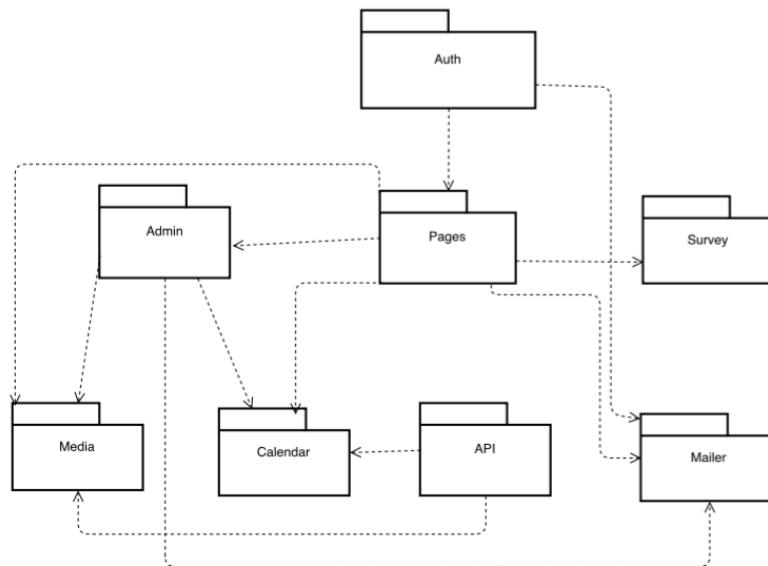


El patrón de arquitectura elegido está alineado con la forma de trabajar en RoR. En ese sentido, se centra en la relación modelo-vista-controlador. Cada una de estas capas representa una abstracción de comportamiento lógico de la aplicación. La capa de *Action Controller* es el conjunto de módulos encargados de coordinar el funcionamiento del servidor. *Active Record* es una clase de la cual todos los modelos heredan. Esto permite abstraer la interacción con la DB de manera de ofrecer un interfaz simple. Finalmente, como se observa en el diagrama, *Action View* agrupa todos los recursos visuales que se despliegan en un navegador.

## 2.2 Arquitectura Lógica

La arquitectura Lógica presenta un refinamiento de la Arquitectura del Sistema. Los módulos más relevantes para esta sección son Media, Admin, Pages y API, las cuales van a ser explicadas en detalle. Aquí cada modulo es una agrupación de controladores, vistas, y modelos. Esto permite visualizar la interacción de los componentes del sistema.

Diagrama 2.2: Arquitectura Lógica



### 2.2.1 Escenarios

La Vista de Casos de Uso muestra el *front-end* del sistema. Este es generado dinámicamente utilizando tecnología de contenido web dinámico (imágenes, videos, secuencias, calendario) y estático (ejercicios de respiración y evaluaciones). Desde el punto de vista del *back-end* se tienen páginas dinámicas y estáticas.

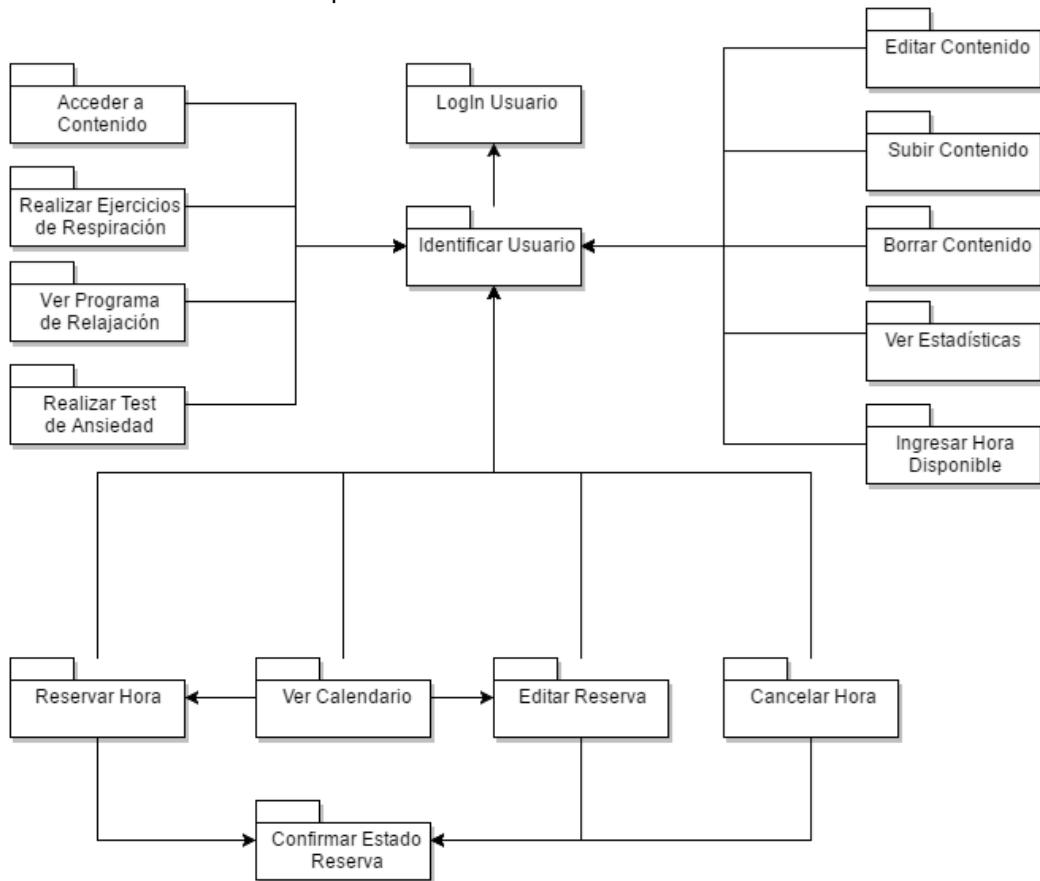
Dado que en para el proyecto se están usando Historias de Uso, para este diagrama se agruparon

historias similares en requerimientos, con un módulo por cada uno. La excepción son los requerimientos de contenidos (imágenes, videos y secuencias), los cuales se agruparon en “contenido”. Cada requerimiento requiere múltiples interacciones con el usuario.

Los flujos de la aplicación son lo más simples posibles para facilitar su uso, lo que es parte del objetivo de esta. Debido a esto las conexiones en el diagrama son de baja complejidad. La interacción más complicada está relacionada al calendario y la toma de horas en el sistema de salud. Además, se puede ver que la aplicación incluye un gran número de módulos y, por lo tanto, una gran cantidad de contenido.

Los módulos identificados y sus interdependencias se presentan en el siguiente diagrama:

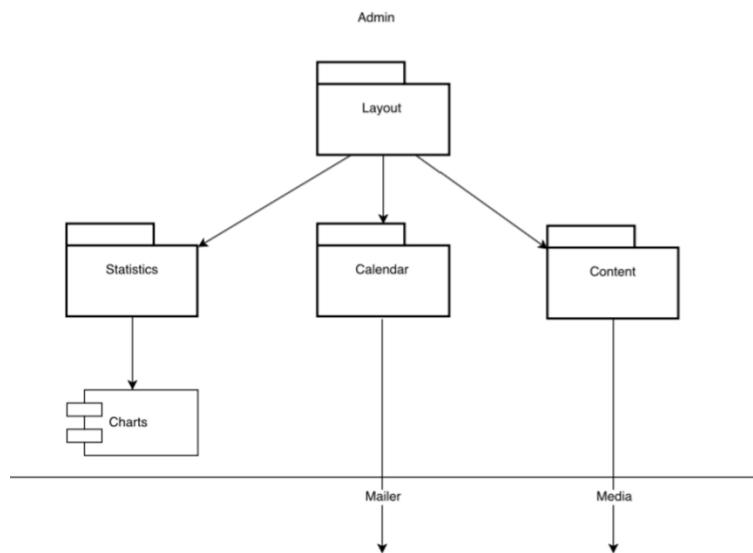
Diagrama 2.3: Módulos e interdependencias



## 2.2.2 Admin

El modulo Admin se relacionan con las funcionalidades que debe poder realizar el Administrador de la aplicación. Las estadísticas se obtienen en base a la información guardada en Logs. Para la representación grafica de este se ocupa el módulo Charts. Calendar se encarga de manejar el sistema de actividades de los administradores. Al mismo tiempo el calendario ocupa el modulo Mailer para notificar a los usuarios. Por último, Content se encarga de la forma en que el Admin maneja el contenido multimedia.

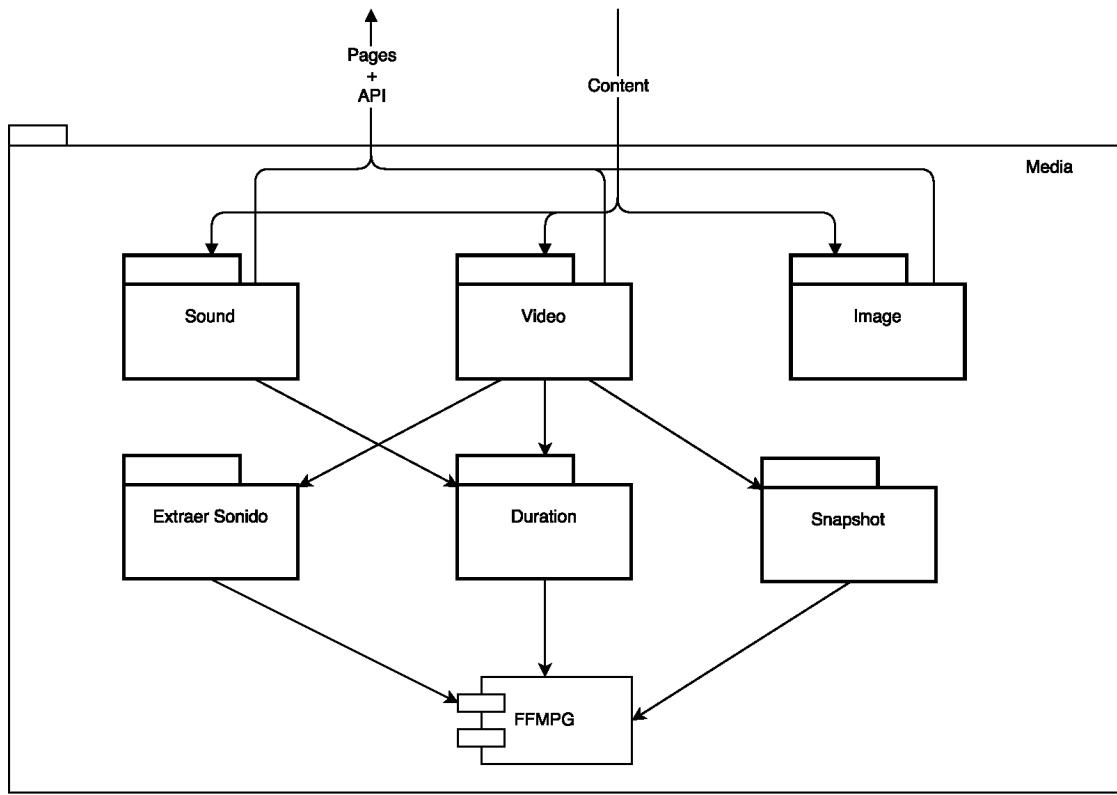
Diagrama 2.4: Modelo módulo Admin



## 2.2.3 Media

Los componentes de multimedia se agrupan en el modulo Media. En este hay tres elementos primitivos. Primero se encuentra Sound, luego Video y por ultimo Image. Estos representan todos los tipos de contenido multimedia que maneja la aplicación. Estos elementos interactúan con modulo FFMPG que permite procesar estos elementos primitivos. FFMPG es un software gratuito que produce librerías y programas para manejar contenido multimedia.

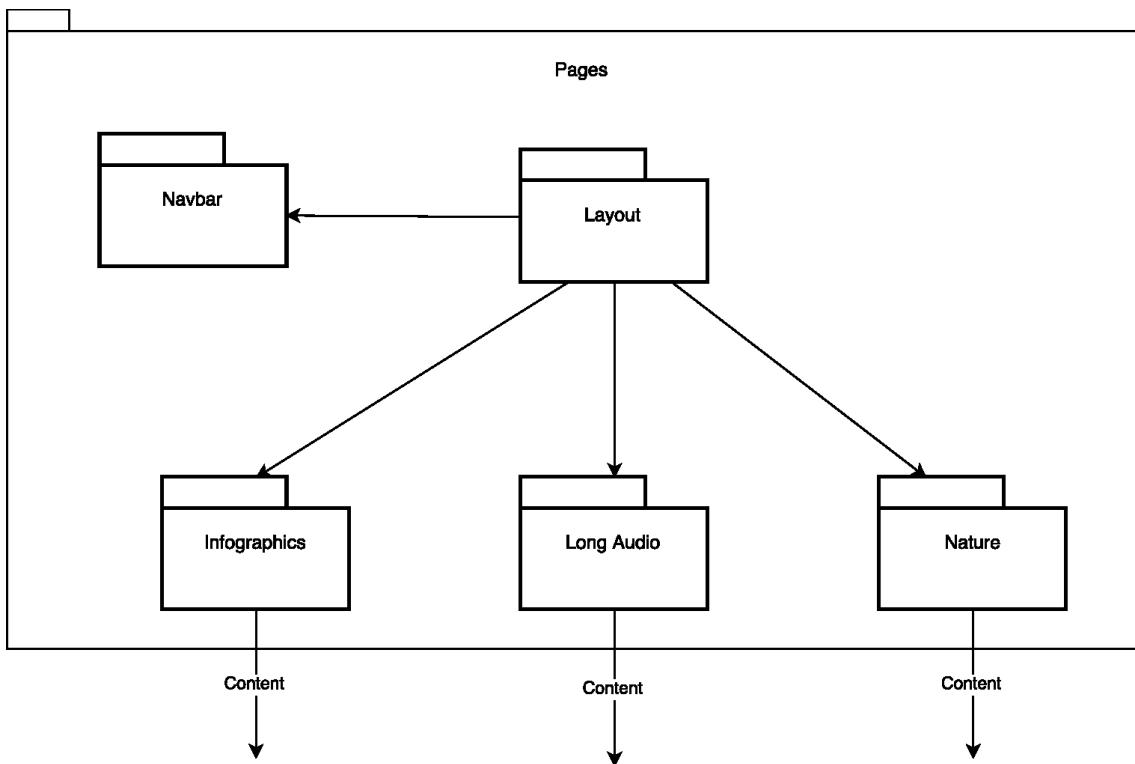
Diagrama 2.5: Modelo módulo Media



#### 2.2.4 Pages

Los componentes de Pages incluye a todos los requerimientos que se exhiben en la aplicación web. Infographics representa las secuencias de imágenes, Long Audio incluye las imaginerías y Nature los videos en conjunto con sus audios. Otros componentes de Pages son los ejercicios de respiración y la visualización del Calendario de toma de horas.

Diagrama 2.6: Modelo módulo Pages



#### 2.1.2.5 API

Los componentes de API son básicamente dos. El primero, Api Contenido, incluye toda la capa de comunicación con Android para que se pueda facilitar los datos correspondientes al contenido, los logs y las actividades del calendario. Por otro lado, Api Auth incluye todo el manejo de la administración de sesión que tiene que tener la aplicación. A continuación se presentan dos diagramas que reflejan el funcionamiento de estas APIs.

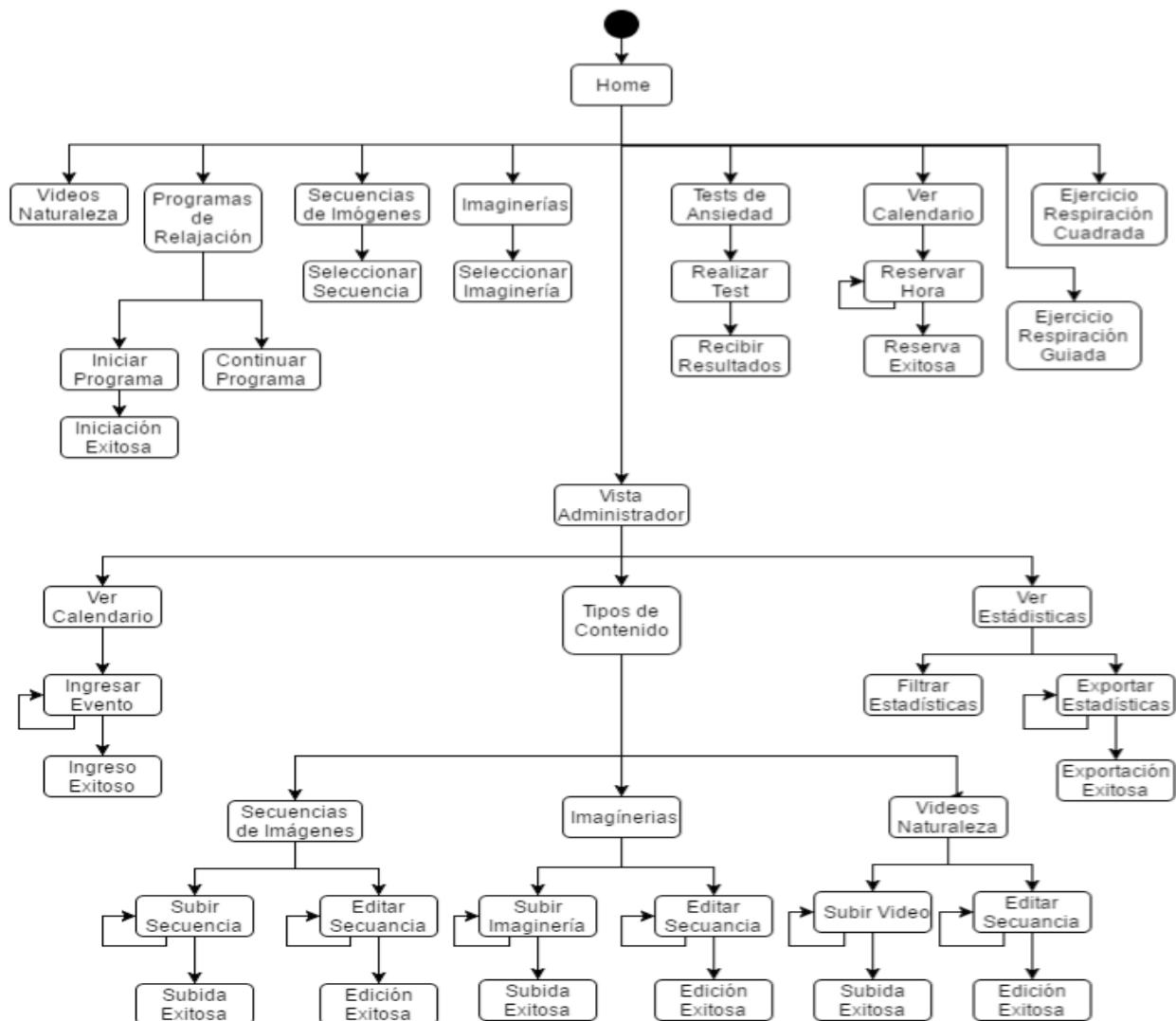
## 2.3 Arquitectura de los Módulos

La Arquitectura de los Módulos presenta un refinamiento de la Arquitectura Lógica. Esta incluye, para cada módulo pertinente, una vista que mejor define su diseño.

### 2.1.3.1 Interfaz de Usuario

#### Modelo de Navegación del Sistema

Diagrama 2.8: Modelo de navegación



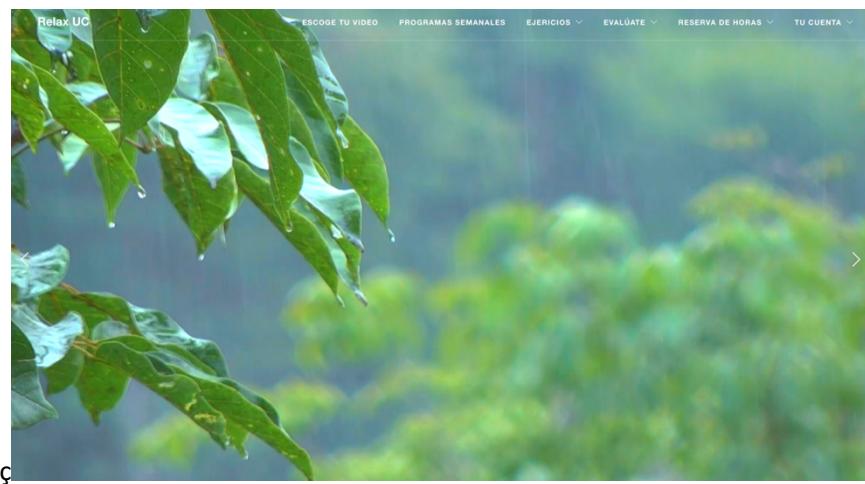
#### **2.1.3.1.1 Pages**

A continuación, las vistas respectivas y sus explicaciones:

##### **Interfaz de Usuario para visualizar Videos de Naturaleza**

Para visualizar los videos, el usuario debe ingresar la sección “Escoge tu Video” en el menú de navegación principal. Luego, el usuario puede utilizar las flechas para encontrar el video que desea ver:

Diagrama 2.9: Vista Naturaleza



##### **Interfaz de Usuario para escuchar Imaginerías**

Para escuchar imaginerías, el usuario debe ingresar la sección “Ejercicios” y luego a “Relaja tu Mente” en el menú de navegación principal. Luego, el usuario puede utilizar las flechas para encontrar la imaginería que desea:

Diagrama 2.10: Vista Imaginerías



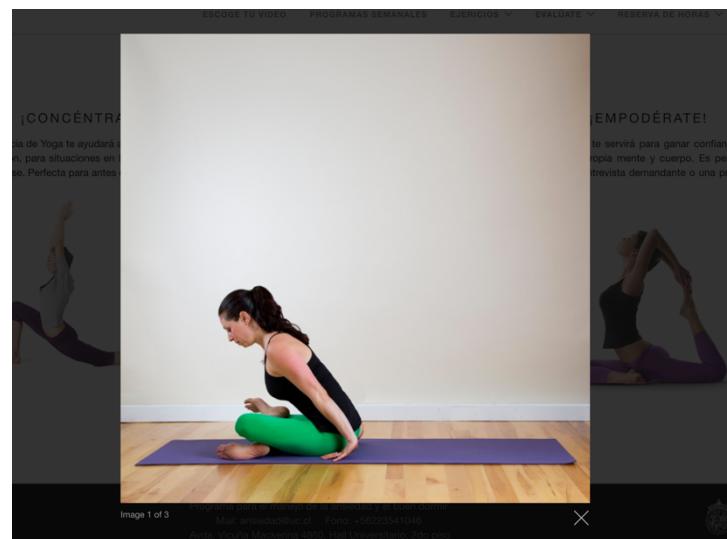
## Interfaz de Usuario para ver Secuencias

Para ver secuencias, el usuario debe ingresar la sección “Ejercicios” y luego a “Relaja tu Cuerpo” en el menú de navegación principal. Luego, el usuario puede seleccionar el tipo de secuencia que desea realizar. Una vez dentro, es necesario moverse con las flechas para ir visualizado los ejercicios respectivos.

Diagrama 2.11: Vista Secuencias General



Diagrama 2.12: Vista Secuencias Específico



## Interfaz de Usuario para ver Programas

Para seguir programas de ejercicios, el usuario debe ingresar la sección “Programas de Relajación” en el menú de navegación principal. Luego, el usuario debe apretar “Comenzar” y podrá comenzar a realizar los distintos programas ofrecidos en la aplicación.

Diagrama 2.13: Vista Programas Comienzo

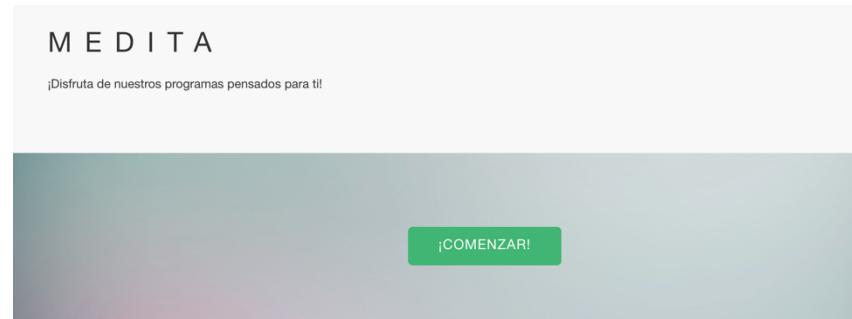


Diagrama 2.14: Vista Programas Interno



### 2.1.3.1.2 Media

El diagrama siguiente se centra en el contenido multimedia de la aplicación. Este contenido es central a nivel de diseño, ya que es el servicio central de la aplicación. Los módulos son los distintos tipos de contenido multimedia a los que tiene acceso el usuario. Al visitar un tipo de contenido que hereda de media, se puede registrar el acceso en Logs de manera genérica, lo que permite tener un registro de que están viendo los usuarios de la aplicación.

La siguiente vista presenta las interfaces de la aplicación. La interfaz Auth permite autenticar usuarios, y las otras tres se utilizan para el manejo de contenido multimedia.

Diagrama 2.15: Diagrama de Contenido

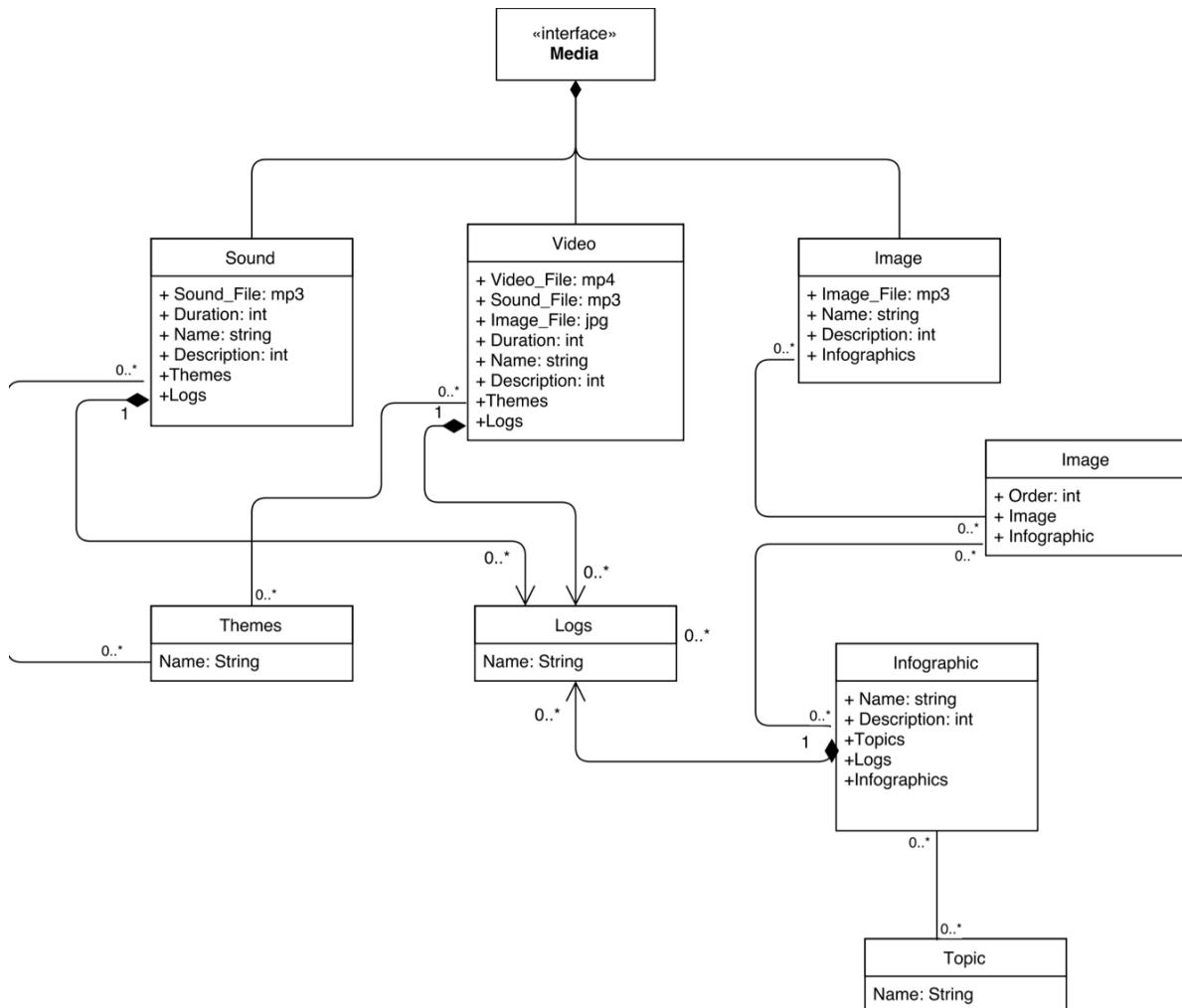
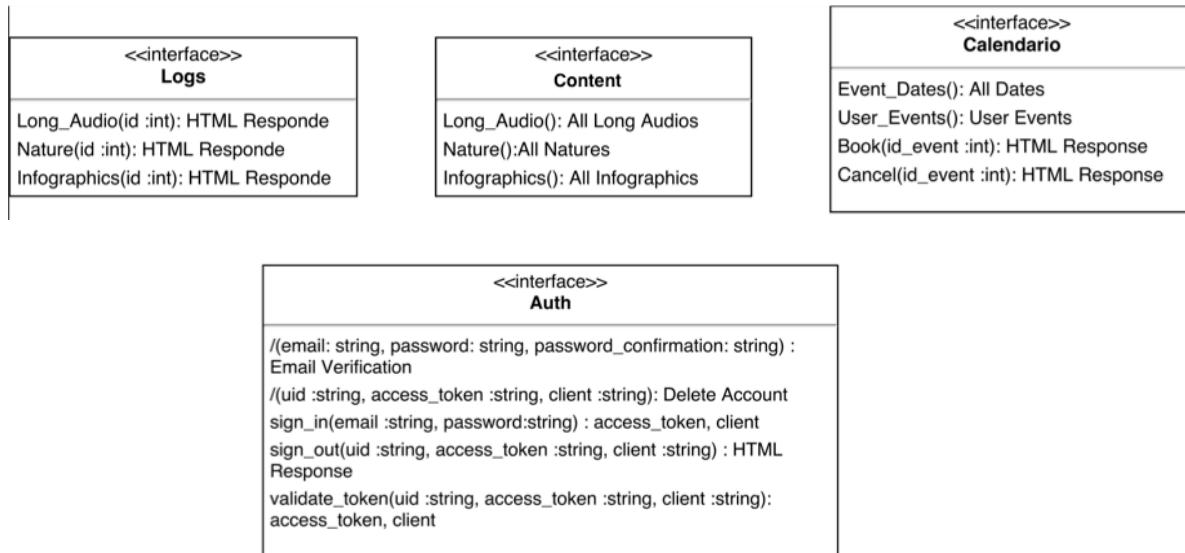


Diagrama 2.16: Interfaces de la aplicación



### Interfaz de Usuario para Subir Contenido

La vista a continuación presenta la vista donde los administradores pueden seleccionar qué contenido multimedia o Media desea subir o editar:

Diagrama 2.17: Vista Seleccionar Contenido



NATURALEZA

IMAGINERÍAS

SECUENCIAS

PROGRAMAS

La siguiente interfaz es equivalente para imaginerías, videos de naturaleza y secuencias de imágenes.

La lógica para subir contenido puede representarse mediante una máquina de estados. Por esta

razón un modelo de estados es el adecuado para describir su diseño interno.

Diagrama 2.18: Vista Subir Contenido

NUEVA IMAGINERÍA

**NOMBRE**  
INGRESAR NOMBRE

**DESCRIPCIÓN**  
INGRESAR DESCRIPCIÓN

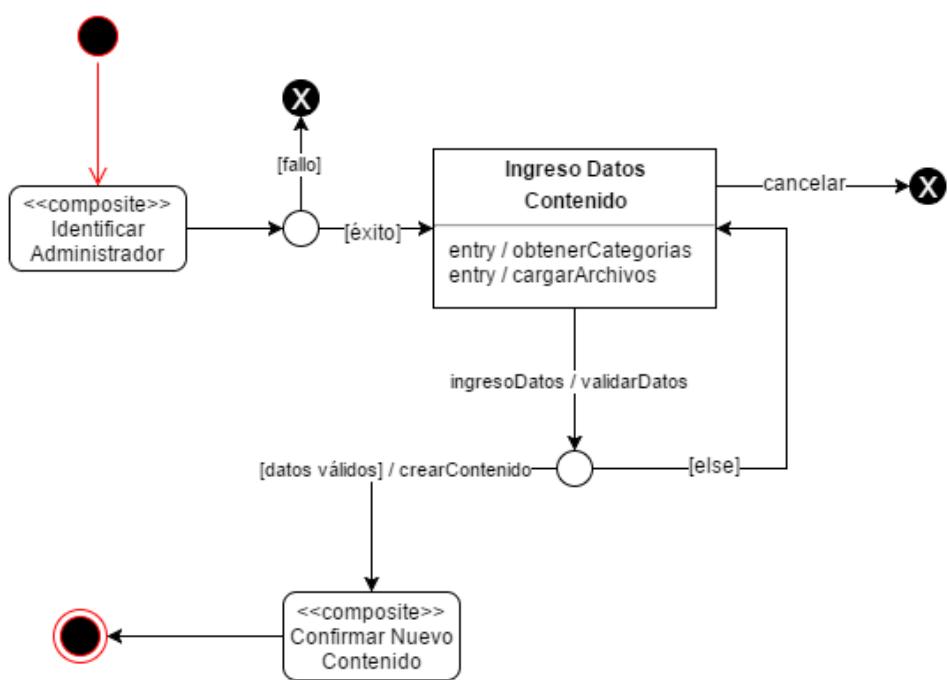
**ARCHIVO**  
Choose File | No file chosen

**CATEGORÍAS** 0  
Playa  
Bosque  
Lago

**SUBIR IMAGINERÍA**

La captura de pantalla muestra una interfaz web para subir contenido. El título es "NUEVA IMAGINERÍA". Los campos son: "NOMBRE" (ingresar nombre), "DESCRIPCIÓN" (ingresar descripción), "ARCHIVO" (selección de archivo, actualmente no elegido), y "CATEGORÍAS" (0 categorías: Playa, Bosque, Lago). Un botón verde "SUBIR IMAGINERÍA" se encuentra en el pie de la página.

Diagrama 2.19: Máquina de Estados Subir Contenido



## Interfaz de Usuario para Editar Contenido

La siguiente interfaz es equivalente para imaginerías, videos de naturaleza y secuencias de imágenes. Es acá donde se edita el contenido:

Diagrama 2.20: Vista Editar Contenido

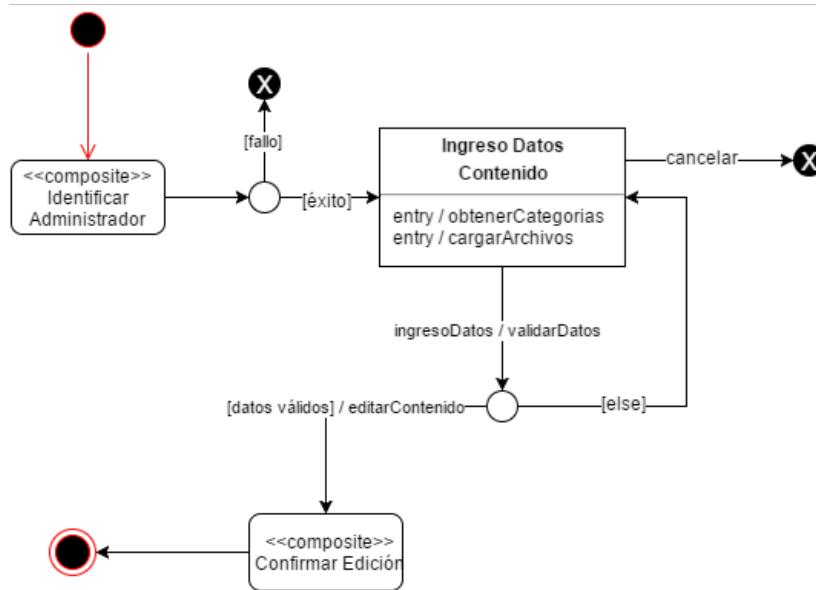
La captura de pantalla muestra una interfaz web titulada 'EDITAR SECUENCIA'. Los campos y sus valores son:

- NOMBRE:** Secuencia de yoga
- DESCRIPCIÓN:** Mejora tu estado mental.
- ARCHIVOS:** [Choose Files] No file chosen
- CATEGORÍAS:** Concéntrate, Desactivate, Empodérate

En la parte inferior hay un botón verde 'SUBIR SECUENCIA'.

La lógica para editar contenido se puede representar mediante una máquina de estados. Por esta razón un modelo de estados es el adecuado para describir su diseño interno.

Diagrama 2.21: Máquina de Estados Editar Contenido



#### **2.1.3.1.4 Calendar**

##### **Interfaz de Administrador para Crear Horario**

Al crear un evento en su calendario, el administrador debe ir a “Calendario” y luego a “Crear horario” dentro de su barra de navegación de administrador. Luego, deberá completar un formulario para la actividad, y en la misma vista podrá ver el evento creado.

Diagrama 2.22: Vista Crear Horario

The screenshot shows a form for creating a schedule entry. It consists of several input fields and a green 'AGREGAR' (Add) button at the bottom. The fields are labeled as follows:

- PROFESOR: (Teacher)
- CAMPUS: (Campus)
- SALA: (Room)
- CAPACIDAD: (Capacity)
- DÍA: (Day)
- HORA DE INICIO: (Start Time) with a small clock icon to its right.
- HORA DE TÉRMINO: (End Time) with a small clock icon to its right.
- REPETIR: (Repeat) with a circular arrow icon above it.

Below the fields is a large green rectangular button labeled "AGREGAR".

Diagrama 2.23: Vista Evento Creado

The screenshot displays a weekly calendar grid from Monday to Friday. The vertical axis represents time from 09:00 to 18:00. A blue event block is visible for each day, spanning from 10:00 to 11:00. The event is labeled "Consejería". The grid shows the following pattern:

Lunes	Martes	Miércoles	Jueves	Viernes
09:00				
10:00	10:00 - 11:00 Consejería			
11:00				
12:00				
13:00		13:00 - 14:00 Consejería		
14:00				
15:00				
16:00			16:00 - 17:00 Consejería	
17:00				
18:00				

## Interfaz de Administrador para Ver Horario General

El administrador debe ir a “Calendario” y luego a “Horario General” dentro de su barra de navegación de administrador. Luego, puede observar todos los eventos registrados:

Diagrama 2.24: Vista Horario General

		Noviembre 2016							
		Lunes	Martes	Miércoles	Jueves	Viernes	Hoy	<	>
		31	1	2	3	4			
		7	8	9	10	11			
		14	15	16	17	18			
		21	22	23	24	25			
		28	29	30	1	2			
		9:30a Apoyo Grupal	9:30a Apoyo Grupal	9:30a Apoyo Grupal	9:30a Consejería	9:30a Consejería			
		9:30a Consejería	9:30a Consejería	9:30a Consejería	9:30a Taller de Mindfulness	9:30a Taller de Mindfulness			
		9:30a Taller de Mindfulness	9:30a Taller de Mindfulness	9:30a Taller de Mindfulness	9:30a Taller de Yoga	9:30a Taller de Yoga			
		9:30a Taller de Yoga	9:30a Taller de Yoga	9:30a Taller de Yoga					

## Interfaz de Usuario para Reservar Horas

El usuario debe ir a “Reserva de Horas” y luego a “Reservar Horas” dentro de su barra de navegación de principal. Luego, puede filtrar las posibles horas y registrar una.

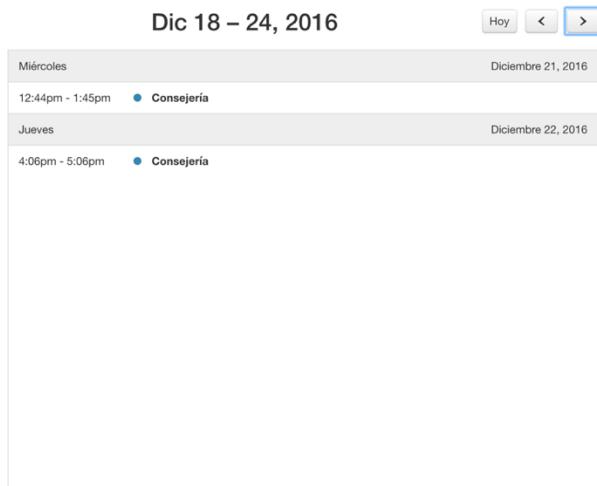
Diagrama 2.25: Vista Reserva de Horas

PROFESOR:	Elige la actividad adecuada para ti.
CAMPUS:	<input type="checkbox"/> Consejerías Son consultas de 60 minutos que te ayudarán a identificar sintomatología puntual que se presente e interfiere en un contexto académico, social o familiar y aprende a enfrentarla.
TIPO DE ACTIVIDAD:	<input type="checkbox"/> Apoyo Grupal Son talleres en grupos de un máximo de 12 personas. Son 4 sesiones mensuales de 90 minutos. Las sesiones tienen como objetivo: (a) entender la ansiedad, (b) aprender a calmar el cuerpo, (c) aprender a calmar la mente y (d) construir un estilo de vida para manejar la ansiedad y el estrés.
BUSCAR	

## Interfaz de Usuario para Reservar Horas

El usuario debe ir a “Reserva de Horas” y luego a “Horas Reservadas” dentro de su barra de navegación de principal. Luego, puede ver en su calendarios sus actuales horas reservadas.

Diagrama 2.26: Vista Horas Reservadas

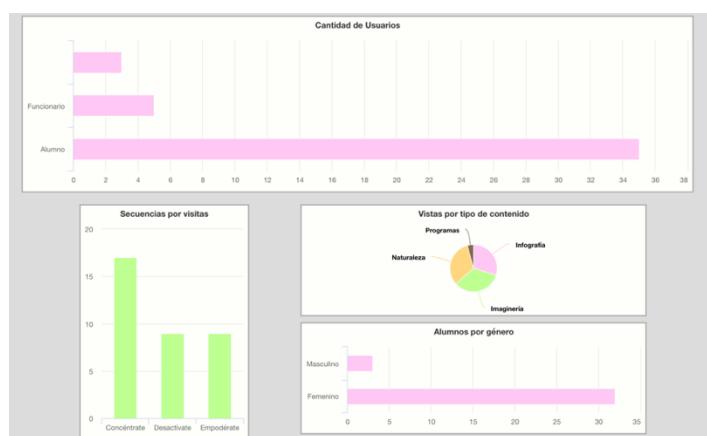


### 2.1.3.1.5 Admin

## Interfaz de Administrador para Visualizar Estadísticas

El administrador, dentro de su menú de navegación, se puede dirigir a “Estadísticas” para acceder a información general de la aplicación:

Diagrama 2.20: Vista Estadísticas



### 3. Vista de Implementación

La vista de implementación presenta los archivos y las dependencias necesarias para que una instancia del servidor interprete y ejecute el código base de RoR.

Ruby implementa módulos como gemas. Estas son colecciones de funcionalidades unitarias. Una gema puede contener varios archivos y son una manera de extender la funcionalidad de Ruby.

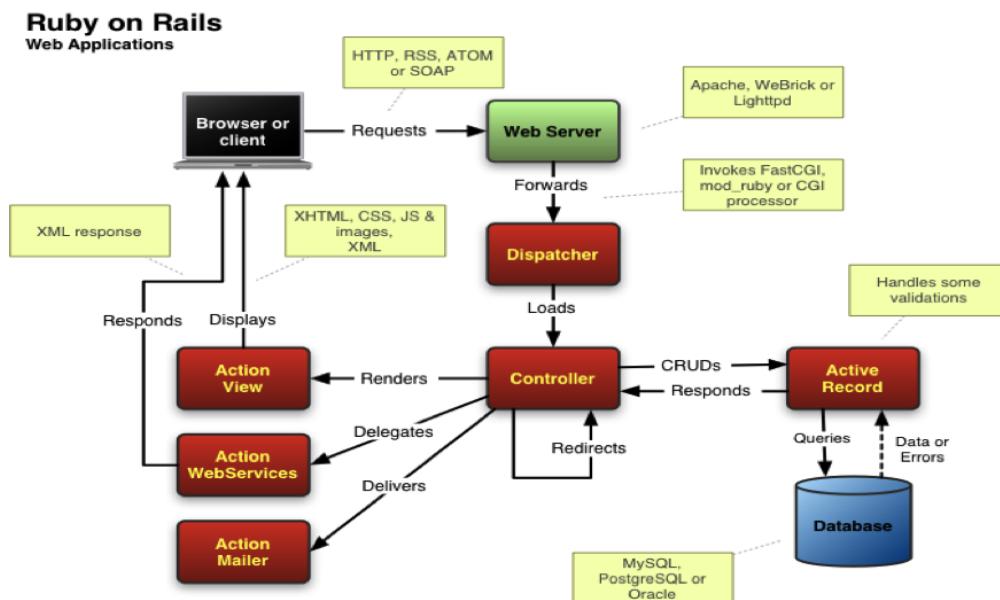
#### 3.1 Estructura de la Aplicación

RoR no compila los archivos en ejecutables. Todos las dependencias y módulos se encuentran dentro de la raíz.

En anexos se presenta un diagrama de componentes que indica las dependencias entre las gemas mencionados anteriormente. Cada entidad representa una gema. Aquellas entidades sin color son parte central de RoR. Aquellas en color son dependencias propias del proyecto (Anexo 6.4).

No obstante, debido a la complejidad del diagrama anterior, su lectura se hace difícil. Un diagrama que explica la implementación de RoR en forma genérica es el siguiente:

Diagrama 3.1: Diagrama de implementación (Fuente: <http://adrianmejia.com/blog/2011/08/11/ruby-on-rails-architectural-design/>)



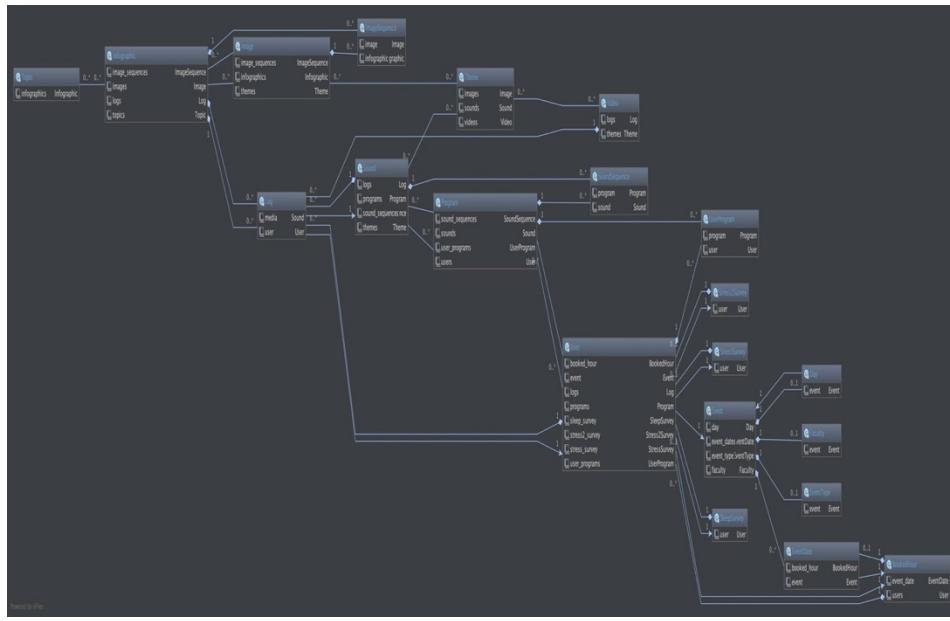
## 4. Vista de Datos

En esta vista se presenta el modelo de datos utilizado para el desarrollo de la aplicación web además de su distribución.

### 4.1 Modelo de Datos

Toda instancia utiliza una única base de datos relacional manejada por el motor PostgreSQL. La 6.1 en la sección de Anexos, presenta el modelo de datos en partes para mayor comprensión (6.1, 6.2 y 6.3).

Diagrama 4.1: Modelo de Datos



### 4.2 Distribución

El diseño de la aplicación tiene una distribución horizontal de datos. La DB PostgreSQL esta alojada actualmente en una instancia EC2 de AWS pero a futuro estará en un servidor del DCC. Por otra parte, el contenido multimedia, que incluye videos, imágenes y audio, está almacenado en Amazon S3 para facilitar las descargas del contenido por parte de los usuarios. Además, el uso de gemas implementadas para el manejo de contenido multimedia como Paperclip permiten hacer esta distribución de manera fácil y conveniente.

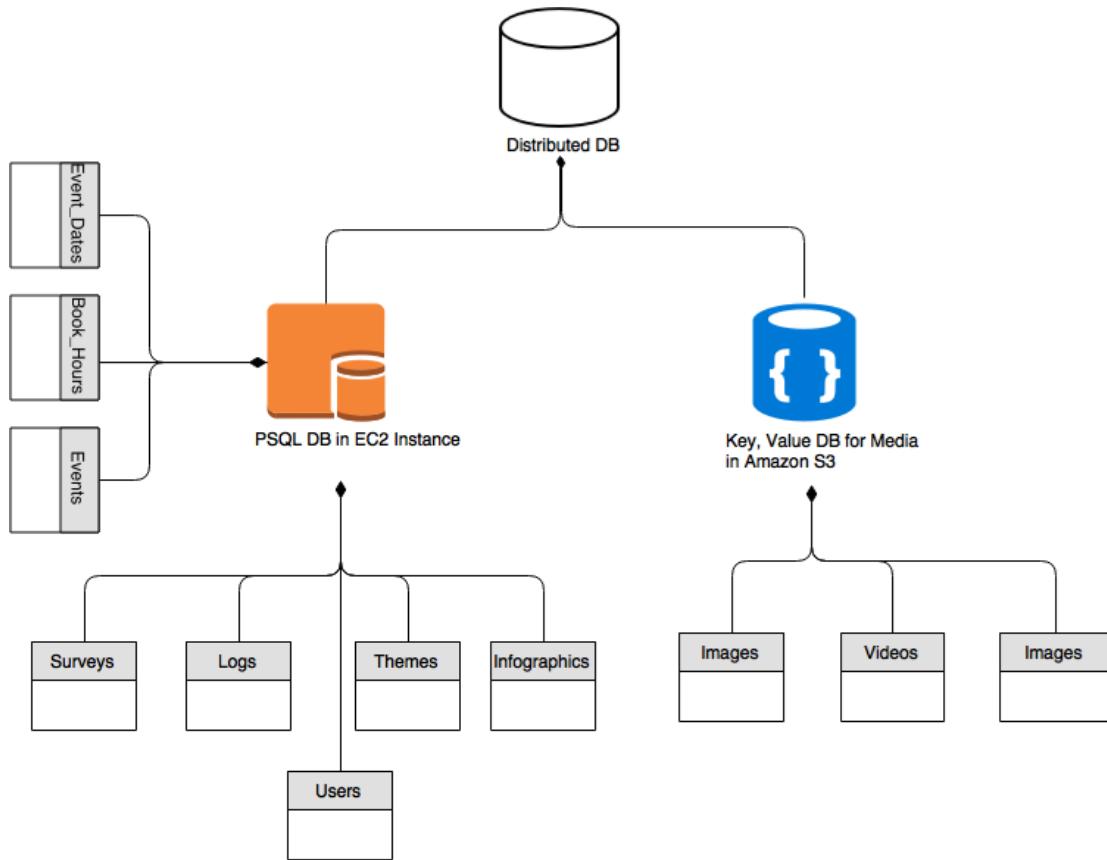
A futuro se maneja la opción de distribuir la tabla de Logs en otra instancia para no bloquear procesos del usuario mientras se ejecutan procesos del administrador que requieren muchos recursos.

## 5. Vista de Deployment

### 5.1 Arquitectura Técnica

Analizando los procesos que sigue la aplicación, es posible identificar los siguientes nodos representativos de la arquitectura: Web Client, Android Client, Server y Server Amazon.

Diagrama 5.1: Diagrama de arquitectura técnica



El nodo Web Client se refiere a los usuarios que se conectan al servidor mediante un navegador web usando la página de la aplicación. Esto incluye tanto Miembros UC como Administradores de la aplicación. Los Android Client son los usuarios que se conectan usando la aplicación de Android. Estos nodos se distinguen por cómo se conectan al servidor. La aplicación web se conecta directamente al backend mientras que la aplicación móvil utiliza la API provista. Además, en web se pueden acceder funcionalidades disponibles para administradores.

El nodo Server está relacionado con el nodo Amazon S3 ya que Amazon decide cómo distribuir la carga en sus servidores. Para función de la aplicación, existe un servidor que recibe los requests de ambos tipos de cliente.

## 5.2 Tecnología requerida

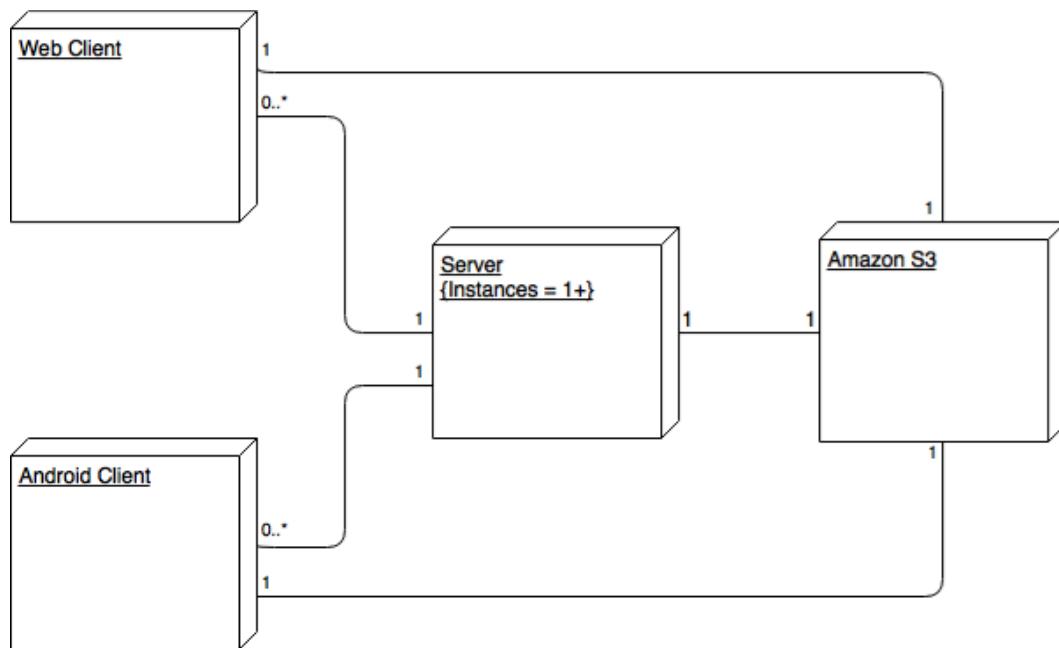
Los usuarios web necesitan un navegador para conectarse a la página web. Por ejemplo, Chrome, Safari y Firefox. El navegador debe ser capaz de ejecutar JavaScript y contar con soporte de HTML 5. Los usuarios de móvil deben contar con Android 4.1 como mínimo. Además, requieren de conexión a internet para descargar la aplicación y descargar contenidos, pero pueden acceder a contenido ya descargado sin estar conectados a la red.

El nodo servidor debe correr una instancia de Linux 16.04 Ubuntu, en la cual estará una instancia de RoR 5.0.0.1. Los servidores de Amazon S3 son externos a la aplicación.

## 5.3 Deployment

Las dependencias presentadas en la Vista de Implementación residen en el nodo de tipo Server presentado en las secciones anteriores.

Diagrama 5.2: Diagrama de dependencias técnicas



La configuración del servidor Amazon S3 debe estar referenciada en el servidor principal de la aplicación para generar la conexión de manera correcta.

## 6. Anexos

Diagrama 6.1: Modelo de Datos Parte 1

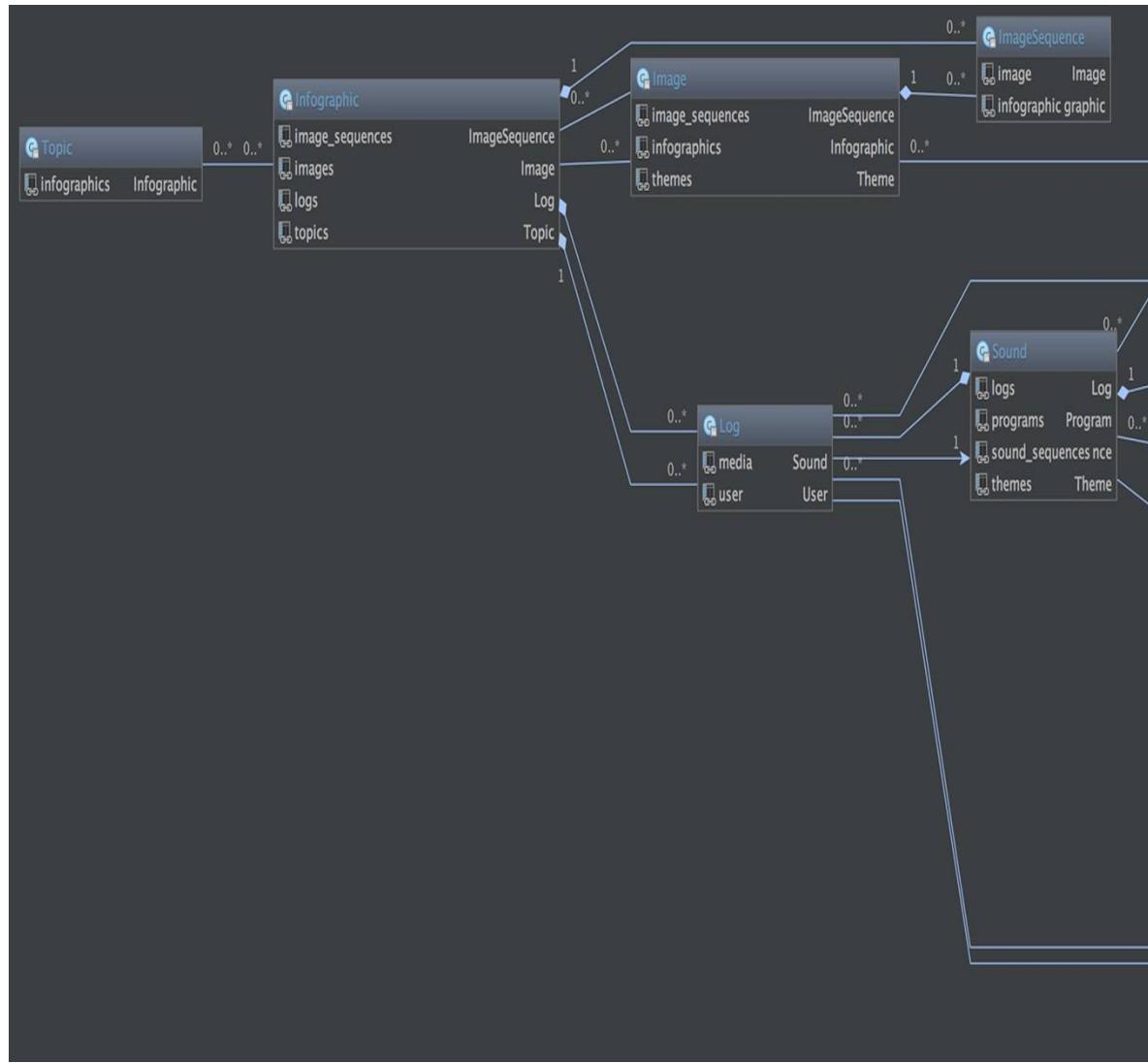


Diagrama 6.2: Modelo de Datos Parte 2

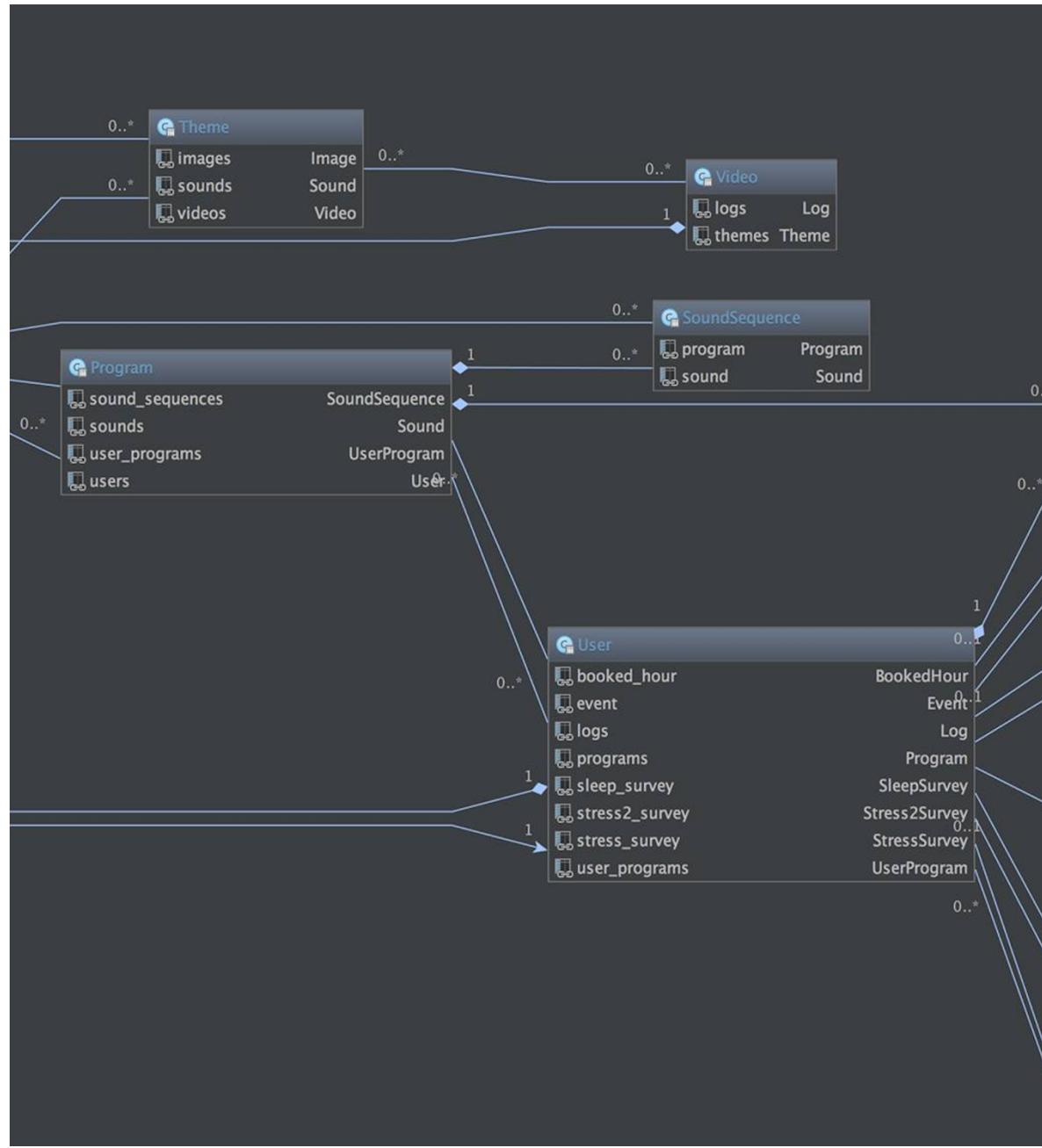


Diagrama 6.3: Modelo de Datos Parte 3

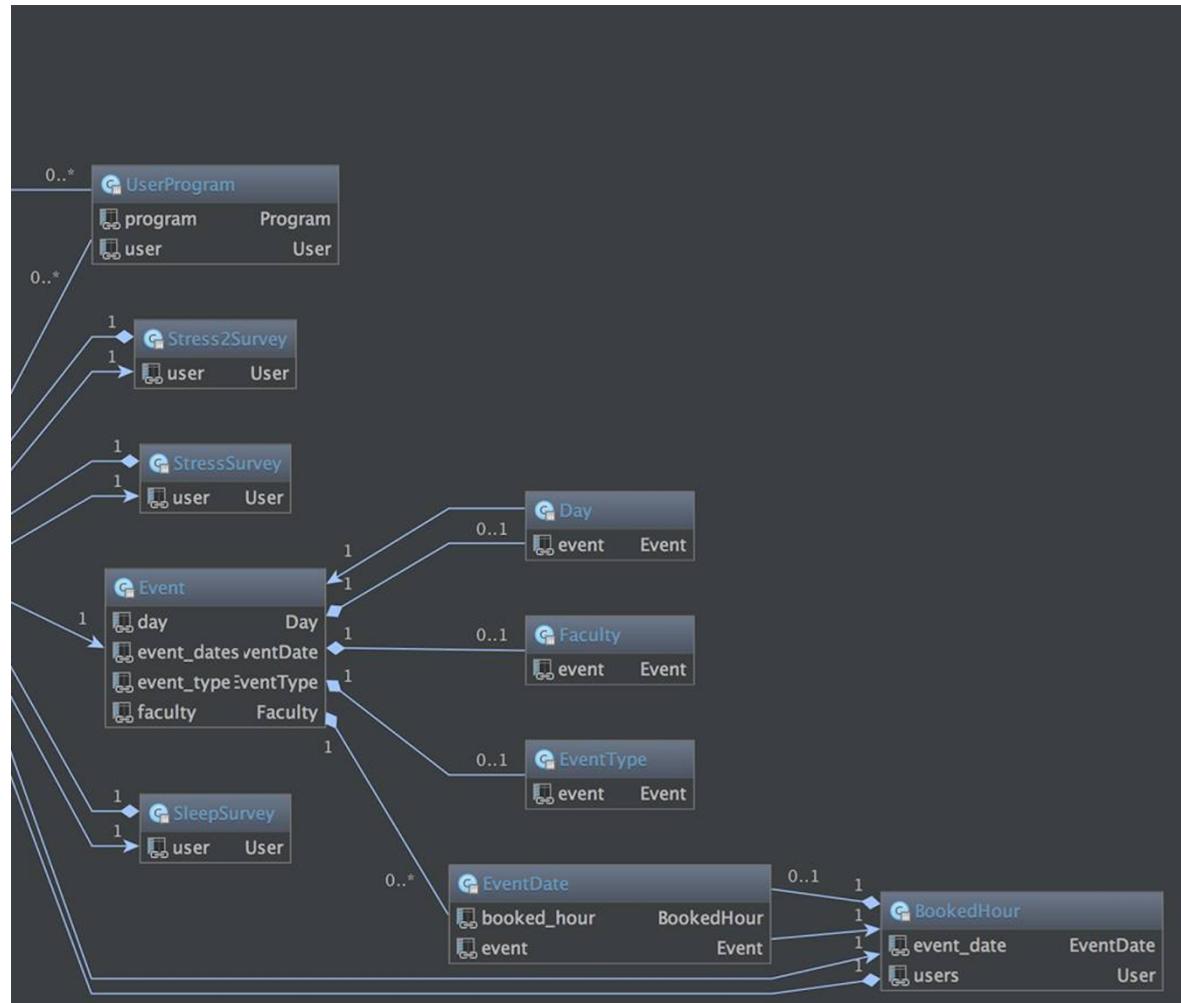


Diagrama 6.4: Modelo de Dependencia de Gemas Completo

