

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

DOCUMENTO DE DISEÑO MÓVIL

Nombre del proyecto:
Organización
Fecha:
Versión:

Relax UC
Salud Estudiantil UC
27/10/16
0.1

Historia del Documento

Versión	Fecha	Autor(es)	Razón del Cambio
0.1	15/10/2016	Esteban Meza	Completar descripción general
1.0	27/10/2016	Equipo proyecto	Completar documento
1.1	20/10/2016	Camilo Álvarez	Actualizar documento
2.0	02/12/16	Rodrigo Saavedra	Actualizar correcciones documento

Equipo de Desarrollo

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

Contraparte del Proyecto

Nombres y Apellidos	Rol	Contacto
María Paz Jana	Coordinadora	mpjana@uc.cl 2 2354 1046
María Trinidad Barriga	Psicóloga	mtbarriga@uc.cl 2 2354 1046
María Soledad Zuzulich	Directora Salud Estudiantil	mzuzulich@uc.cl 2 2354 5402

Tabla de Contenidos

Historia del documento.....	2
Equipo de desarrollo.....	3
Contraparte del proyecto.....	3
1. Descripción General	4
1.1 Propósito	4
1.2 Alcance	4
1.3 Visión General	5
1.4 Definiciones, Acrónimos y Abreviaturas.....	5
1.5 Referencias.....	7
2. Vista Lógica.....	8
2.1.1 Arquitectura del Sistema	8
2.1.2 Arquitectura Lógica	9
2.1.3 Arquitectura de los Módulos.....	10
3. Vista de Implementación	20
3.1.1 Estructura de la Aplicación	20
3.1.2 Arquitectura de Implementación	20
4. Vista de Datos.....	22
4.1.1 Modelo de Datos	22
5. Vista de Deployment.....	23
6. VRelaxUC Realidad Virtual.....	24

1. Descripción General

1.1 Propósito

El Sistema de promoción y autocuidado de salud mental 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 y mandalas.

- **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
- Programas semanales de ejercicios de relajación

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 divide sus funcionalidades para dos tipos de usuarios: miembros de la comunidad UC y usuario administrador, mientras que la aplicación móvil es exclusivamente para miembros UC. Por último, la aplicación VR está pensada para ser usada en un centro de atención que aún no se construye.

El servicio principal para miembros UC 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, programas semanales de ejercicios de relajación y un sistema de tomas de hora en los centro de salud UC. Además, estos 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 en 360°, incluyendo audios complementarios.

La sección para usuarios administradores permite subir, editar y borrar contenidos de los siguientes tipos: imaginerías, secuencias de imágenes, videos y programas de ejercicios. También se puede acceder a diversas estadísticas sobre la aplicación, como número de usuarios, reproducciones de cada contenido y promedios de los resultados de los tests psicológicos. Finalmente, se puede editar el calendario de actividades, agregando las fechas y eventos correspondientes. Los miembros UC podrán reservar horas según lo subido por los usuarios administradores. 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

1. APK: *Android Application Package*: paquete del sistema operativo Android que se utiliza para distribuir e instalar componentes en smartphones y tablets que utilizan dicho sistema operativo.
2. Botón SOS: Botón de fácil acceso que lleva directamente a un ejercicio de relajación.

3. Calendario de actividades: Calendario que contiene horas que los usuarios pueden reservar.
4. Examen de ansiedad: Una serie de preguntas con el objetivo de entregar una evaluación del nivel de estrés del usuario.
5. Gear VR: Lentes de realidad virtual.
6. Imagen 360°: Imagen que muestra toda la escena donde fue tomada desde todos los ángulos.
7. Imaginería: Audios de relajación. Pueden ir acompañados de videos o imágenes.
8. Miembro UC: Persona que pertenece a la comunidad de la Pontificia Universidad Católica de Chile.
9. Normal: Vector que sale en ángulo de 90° con relación a la tangente de una curva.
10. Oculus: Empresa desarrolladora de los Gear VR.
11. Perfil de usuario: Cuenta con la cual un usuario puede acceder a las funcionalidades de la aplicación.
12. Prefab: Son los objetos que se insertan en las escenas de unity.
13. Plan de ejercicios: Una serie de ejercicios predeterminados para seguir a lo largo de un determinado tiempo.
14. Realidad Virtual (VR o RV): Es un entorno de escenas u objetos de apariencia real. La acepción más común refiere a un entorno generado mediante tecnología informática, que crea en el usuario la sensación de estar inmerso en él.
15. Respiración alternada: Similar a la respiración diafragmática, pero se alterna la fosa nasal usada en cada inhalación y respiración.
16. 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.
17. 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 se inhalación y exhalación.
18. Respiración guiada: Ejercicio que le indica al usuario cuando inhalar y cuando respirar.
19. SDK: *Software Developer Kit*, son herramientas para trabajar con tecnologías específicas.
20. Secuencias de imágenes: Series de imágenes explicativas de ejercicios de yoga.
21. Unity: Framework utilizado normalmente para la creación de videojuegos y que será utilizado para las aplicaciones de realidad virtual.
22. Textura: Imagen utilizada para cubrir objetos 3D y que estos puedan tener un diseño en la superficie.
23. Video 360°: Video que muestra toda la escena donde fue grabado desde todos los ángulos.

1.5 Referencias

1. Plantilla Diseño IIC2154 v. 1.4

2. Vista Lógica

En esta sección se presentará la vista lógica del Sistema de promoción y autocuidado de salud mental. El sistema cuenta con cuatro subsistemas: Subsistema de toma de horarios de consultas, Subsistema de evaluación psicológica, Subsistema de reproducción de contenido multimedia y Subsistema de usuarios. La vista lógica presenta tres niveles de arquitectura: arquitectura de sistema, arquitectura lógica y arquitectura de módulos. El primer nivel corresponde a la arquitectura del sistema de alto nivel. El segundo nivel muestra la composición de módulos y su interacción para llevar a cabo los requisitos. El último nivel presenta en mayor detalle los módulos participantes de la arquitectura junto a un diagrama que detalla el diseño de cada módulo. El tipo de diagrama utilizado varía según el módulo en cuestión, lo cual se debe principalmente a que los módulos tienen diferentes roles, según su ubicación en el nivel anterior.

2.1.1 Arquitectura del Sistema

En el primer nivel se especifica el patrón de arquitectura para el Sistema, el cual está organizado utilizando el patrón de arquitectura en capas. Se pueden identificar tres capas. En la figura 1 se presenta la Arquitectura del Sistema.

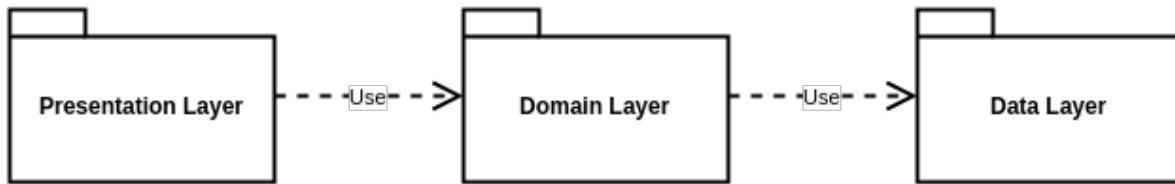


Figura 1: Arquitectura del sistema.

Las capas del sistema se dividen en: capa de presentación, capa de dominio y capa de datos. Cada capa tiene un rol determinado para los módulos que residen en ella.

La capa de presentación tiene como objetivo presentar la información al usuario y recibir *input* del mismo. Está conformada principalmente por las vistas que corresponden a Fragment y Activity en Android.

La capa de dominio es la capa que contiene los modelos del Sistema, los cuales representan la lógica de negocios de la aplicación. Estos modelos son usados directamente en la capa de presentación.

Por último, la capa de datos es la capa que se encarga de todo lo relacionado con la obtención y guardado de datos (se encarga de la persistencia). Dentro de esta capa, se ocupa el patrón repositorio para abstraer al cliente desde donde se obtienen los datos. Estos datos pueden venir desde el mismo celular (datos locales) o desde el repositorio remoto.

2.1.2 Arquitectura Lógica

La Arquitectura Lógica presenta los módulos utilizados por las distintas capas de la arquitectura de sistemas. En esta parte se analizarán los módulos presentes en cada capa.

2.1.2.1 Capa de presentación

En esta capa están las vistas del sistema. Existen vistas con contenido estático y vistas con contenido dinámico. Desde el punto de vista de la lógica de negocios, se tiene un conjunto de páginas dinámicas generadas a partir de los procesos llevados a cabo por el sistema.

Cada caso de uso indicado en dicha vista requiere, en general, más de una interacción con el usuario. Esta secuencia de páginas es en general variable, dependiendo de las acciones del usuario. La versión expandida de cada caso de uso representa la lógica de dicha interacción.

Esta capa consiste de un módulo por cada caso de uso identificado. Cada módulo contiene la lógica que lleva adelante el caso de uso y un conjunto de páginas dinámicas utilizadas por dicha lógica.

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

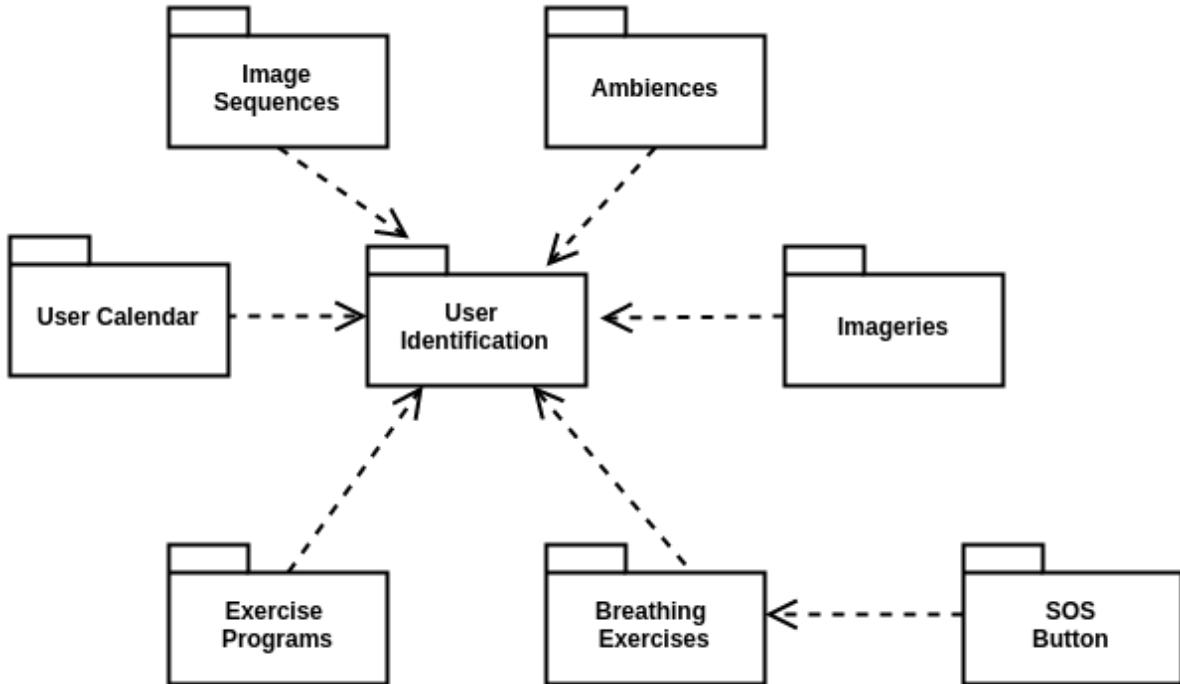


Figura 2: Módulos de capa de presentación.

2.1.2.2 Capa de dominio

Cada módulo de la capa superior utiliza servicios de esta capa. Esta capa tiene los modelos del Sistema, los cuales se pueden ver en la sección de modelo de datos. A grandes rasgos, se identifican 4 módulos, los cuales son representados en el siguiente diagrama,

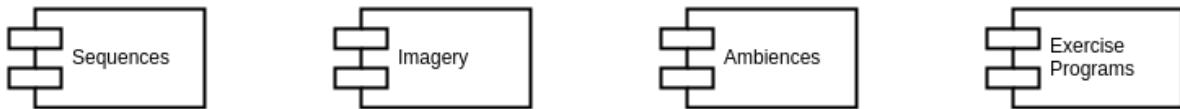


Figura 3: Módulos capa de dominio.

2.1.2.3 Capa de datos

Esta capa contiene varios módulos. Cada módulo corresponde a una implementación del patrón repositorio. Los módulos se pueden ver en el diagrama de abajo:

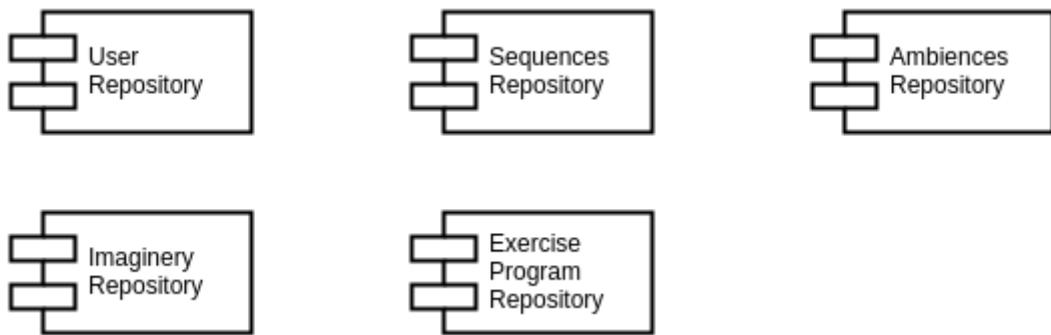


Figura 4: Módulos capa de datos.

El *User Repository*, se encarga de todo lo relacionado con los datos del usuario. Esto incluye su información personal, la toma de horas del calendario, el registro y la obtención del permiso para iniciar sesión.

El *Sequences Repository*, *Imaginery Repository*, *Nature Repository* y *Exercise Program Repository* tienen como objetivo guardar en *caché* la información de las secuencias de imágenes, imaginerías, videos de naturaleza y programas de ejercicios, descargar ese contenido y obtener la lista del servidor.

2.1.3 Arquitectura de los Módulos

La Arquitectura de los Módulos presenta un refinamiento de la Arquitectura Lógica. Esta incluye, para cada módulo, una vista que mejor define su diseño. Para cada tipo de módulo, i.e. para los módulos de cada capa, se utilizará una vista diferente.

2.1.3.1 Interfaz de Usuario

Modelo de Navegación del Sistema

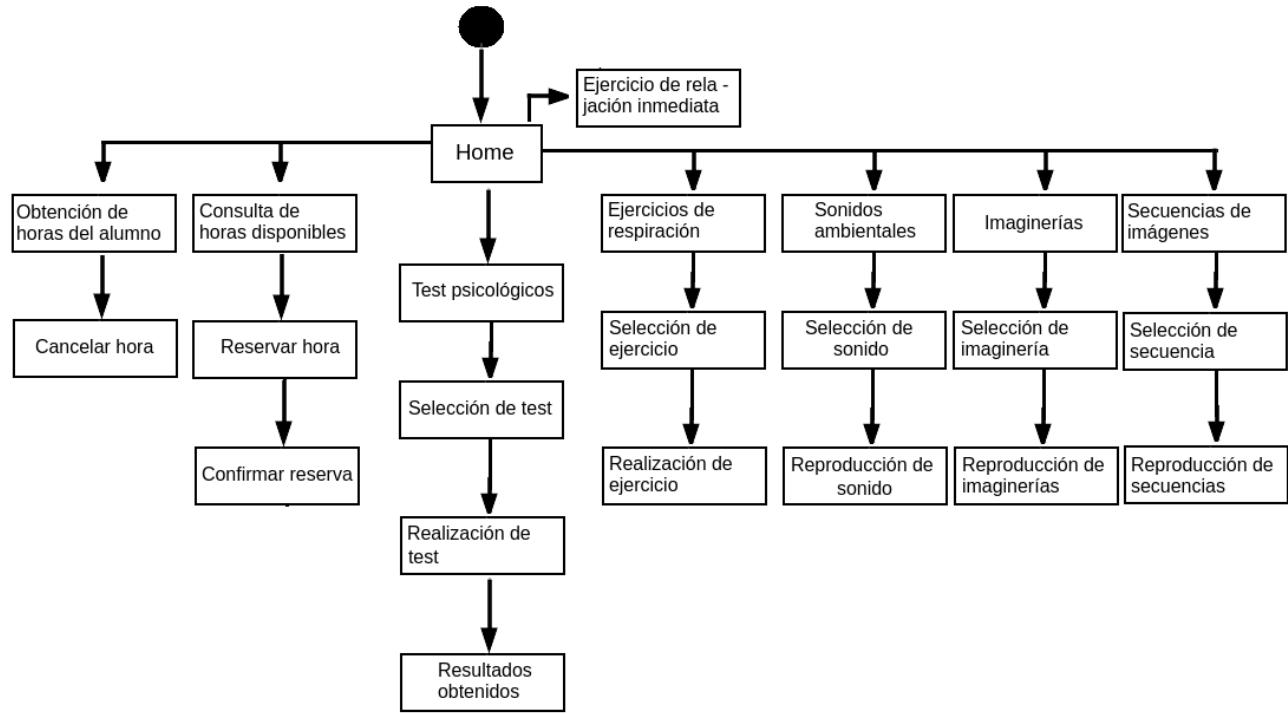


Figura 5: Flujo de navegación del sistema.

Interfaz de Usuario para reservar horas

A continuación, se muestra un diagrama que permite ver la lógica de la reserva de horarios de consulta, junto con las correspondientes imágenes de la interfaz gráfica.

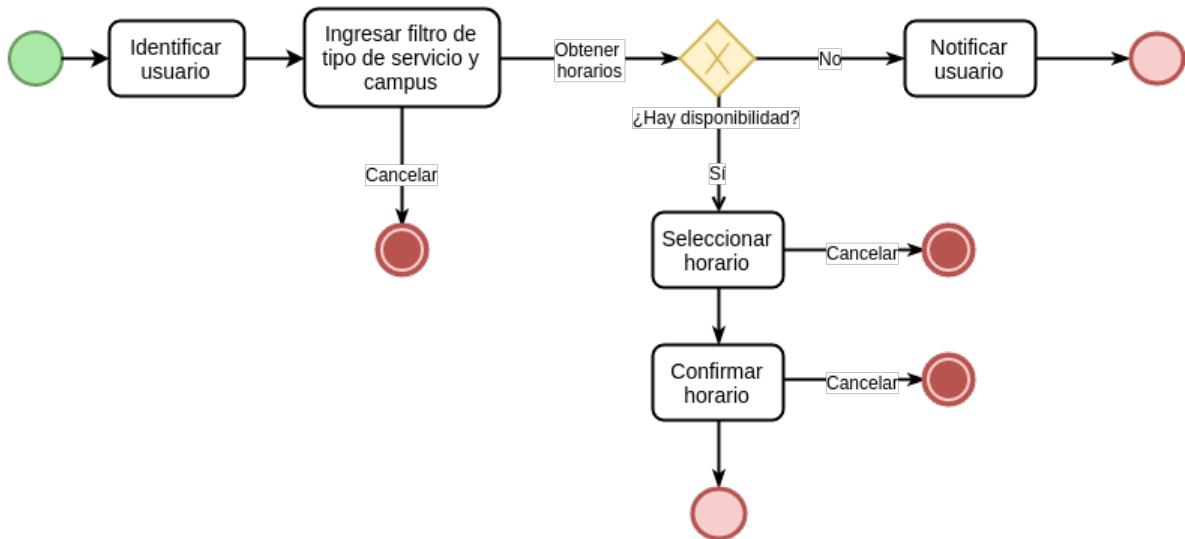


Figura 6: Flujo de navegación de reserva de horas.

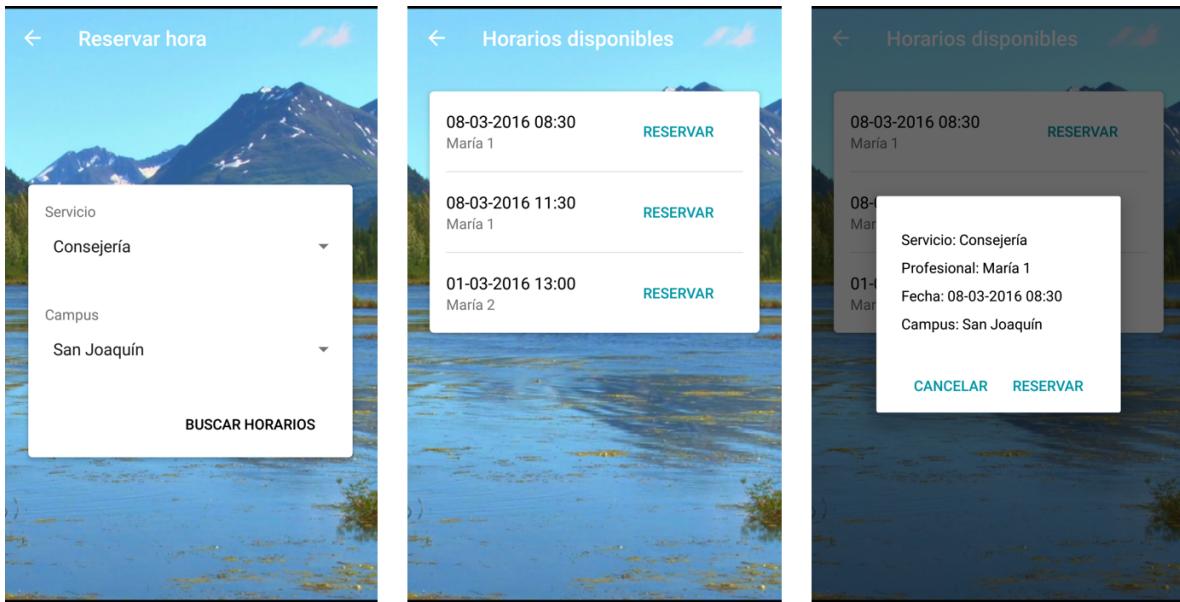


Figura 7: Interfaz de reserva de horas.

Interfaz de Usuario para reproducir contenido

A continuación, se muestra un diagrama que permite ver la lógica de descarga y reproducción de contenido, junto con las correspondiente imágenes de la interfaz gráfica.

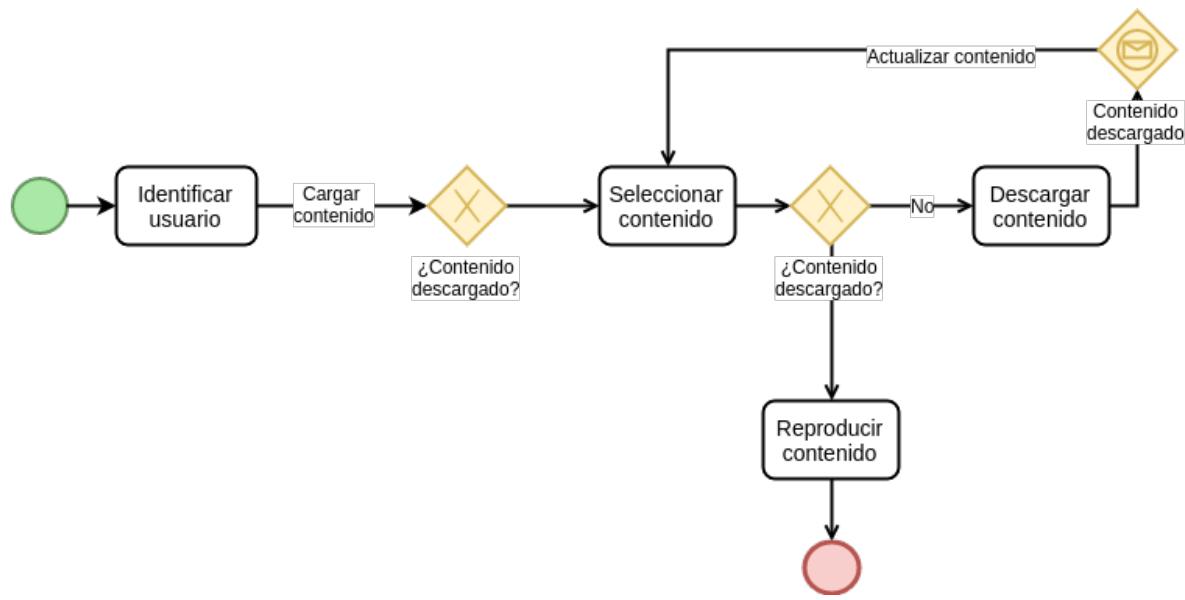


Figura 8: Flujo de navegación de reproducción de contenido.

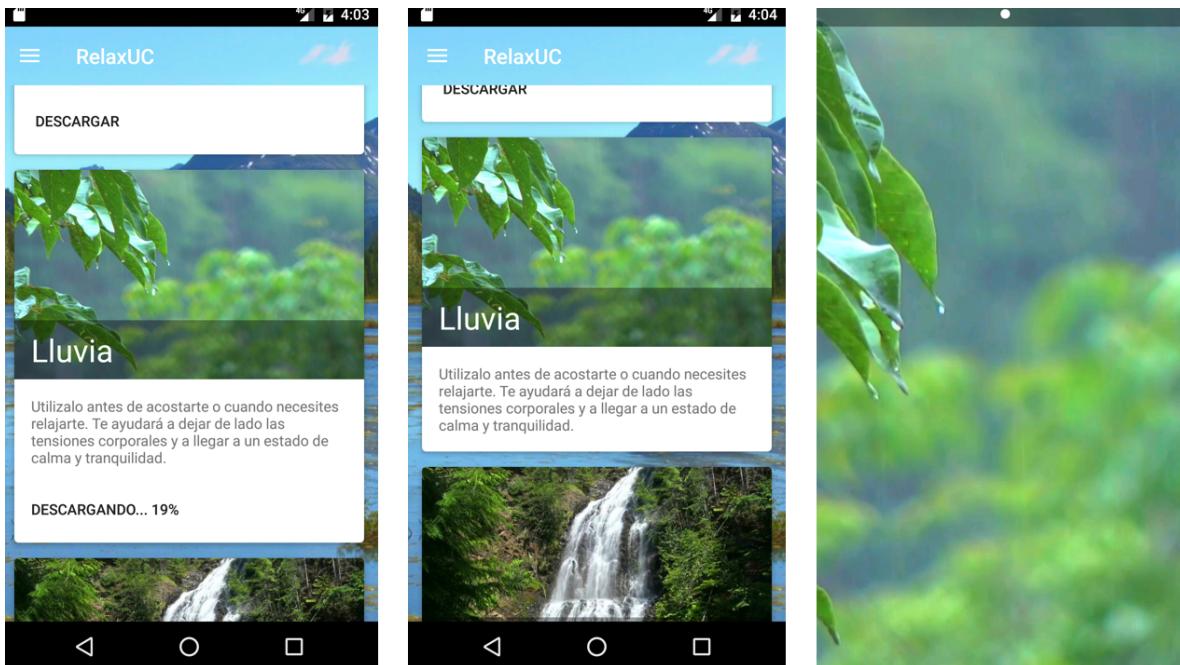


Figura 9: Interfaz de reproducción de contenido.

Las secuencias de imágenes, los sonidos ambientales y las imaginerías funcionan bajo el mismo diagrama de flujo.

Interfaz de Usuario para realización de ejercicios de relajación

A continuación, se muestra un diagrama que permite ver la lógica de reproducción y realización de ejercicios de relajación, junto con las correspondiente imágenes de la interfaz gráfica.

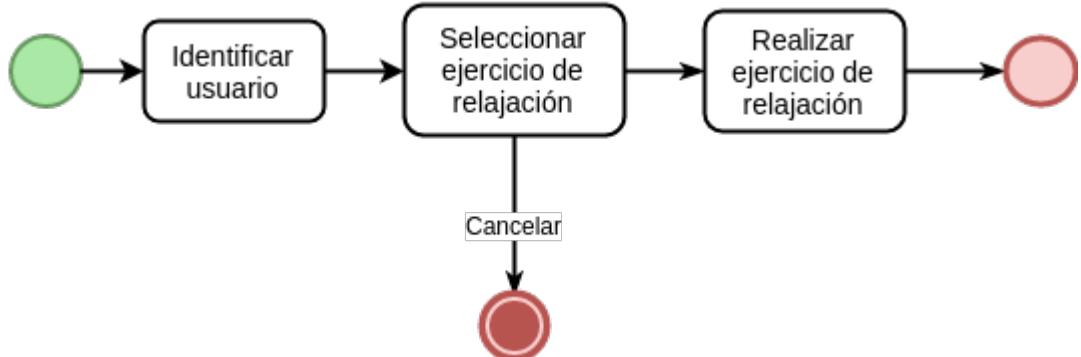


Figura 10: Flujo de navegación de realización de ejercicios de relajación.

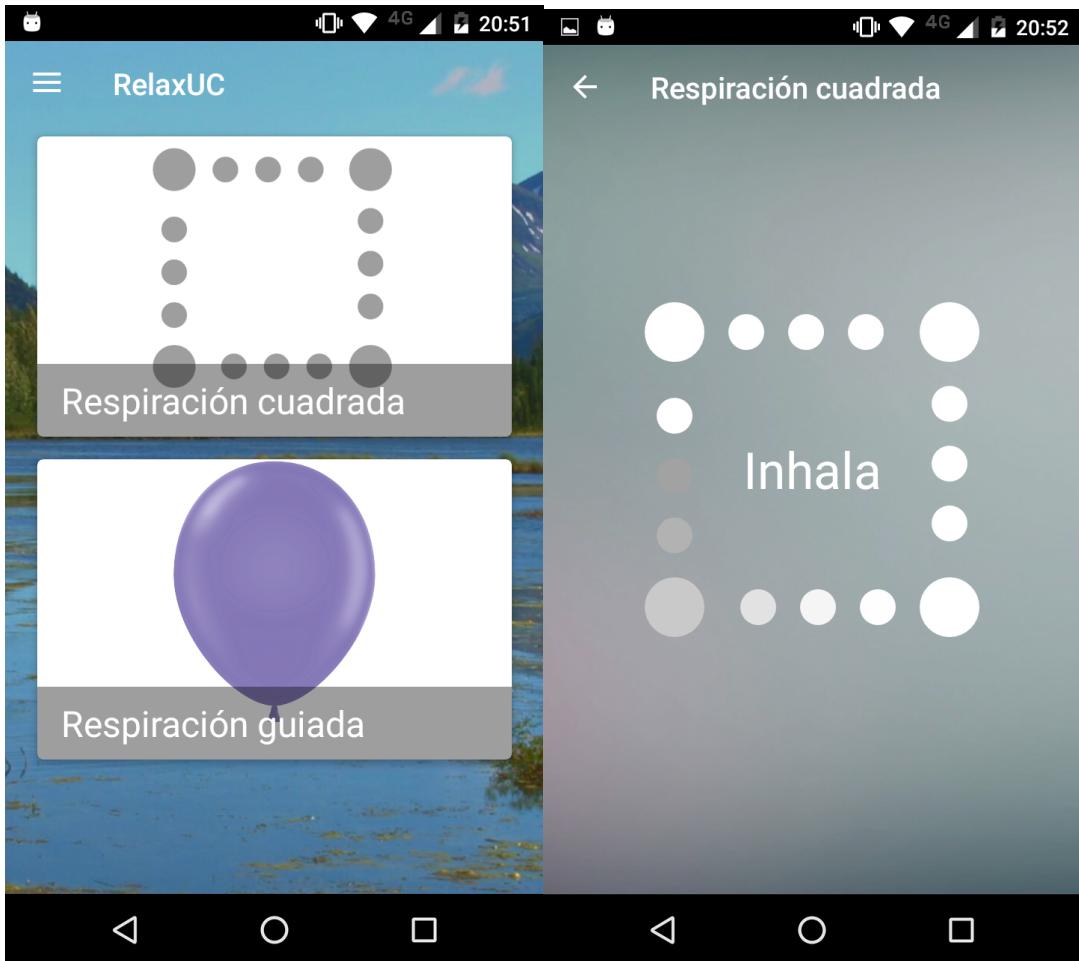


Figura 11: Interfaz de ejercicios de relajación.

Interfaz de Usuario para realizar ejercicios de programas semanales

A continuación, se muestra un diagrama que permite ver la lógica de descarga, selección y reproducción de ejercicios de los programas, junto con las correspondiente imágenes de la interfaz gráfica.

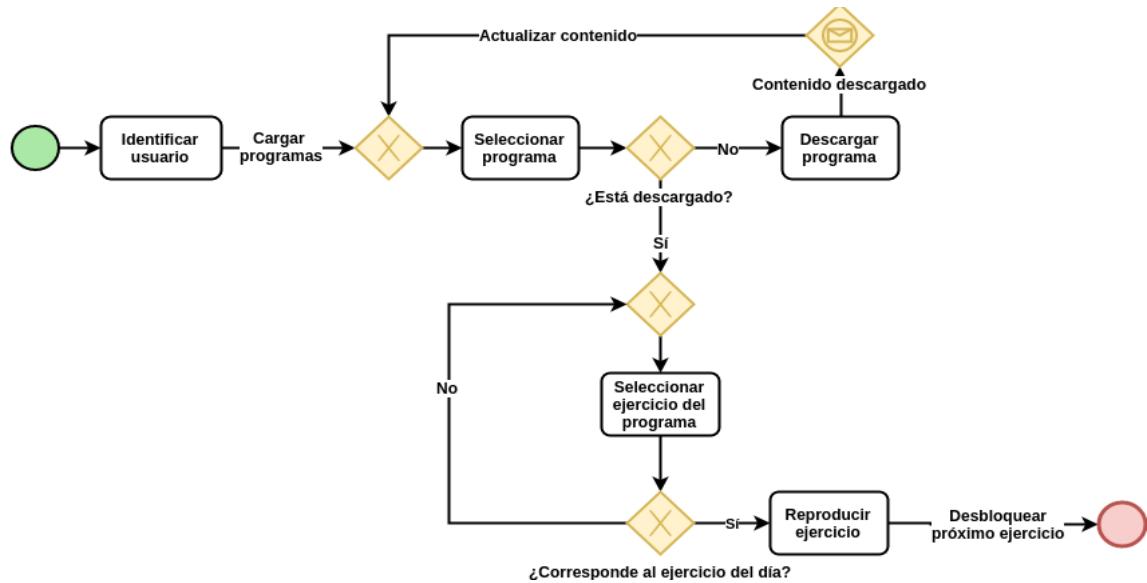


Figura 12: Flujo de navegación de realización de programas de ejercicios.

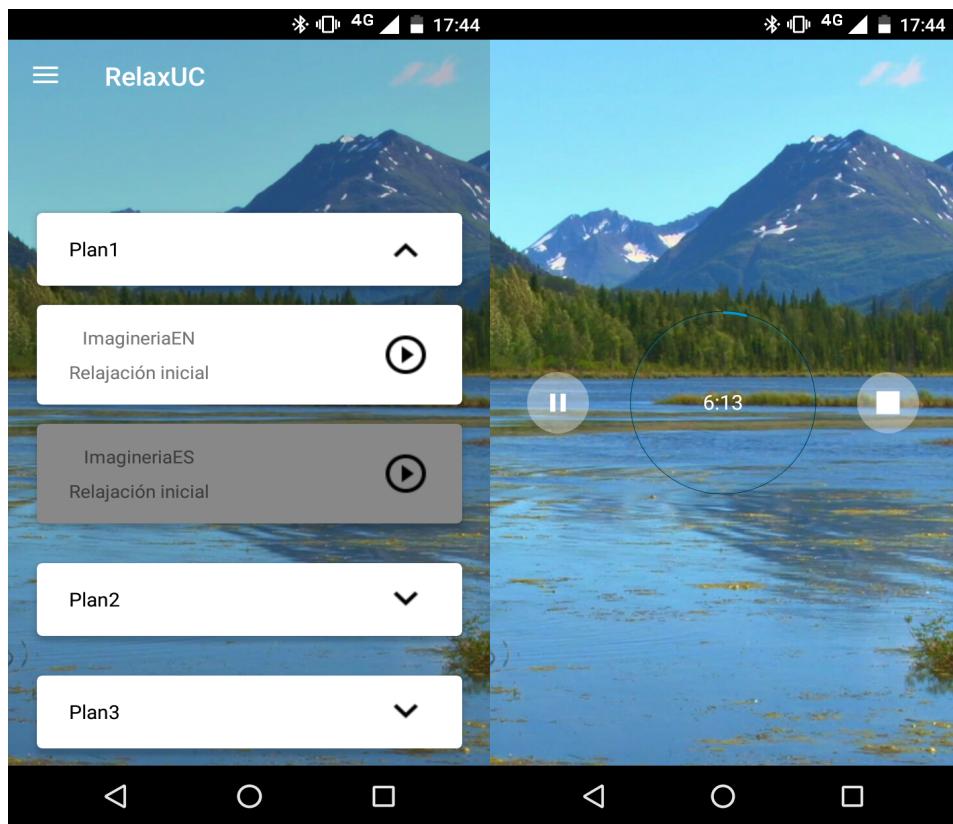


Figura 13: Interfaz de programas de ejercicios.

Interfaz de Usuario para realizar evaluaciones

A continuación, se muestra un diagrama que permite ver la lógica de las evaluaciones, junto con las correspondiente imágenes de la interfaz gráfica.

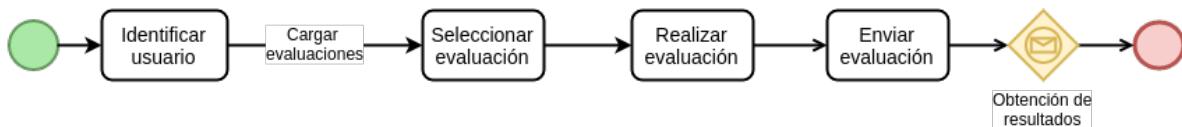


Figura 14: Flujo de navegación de realización de evaluaciones.

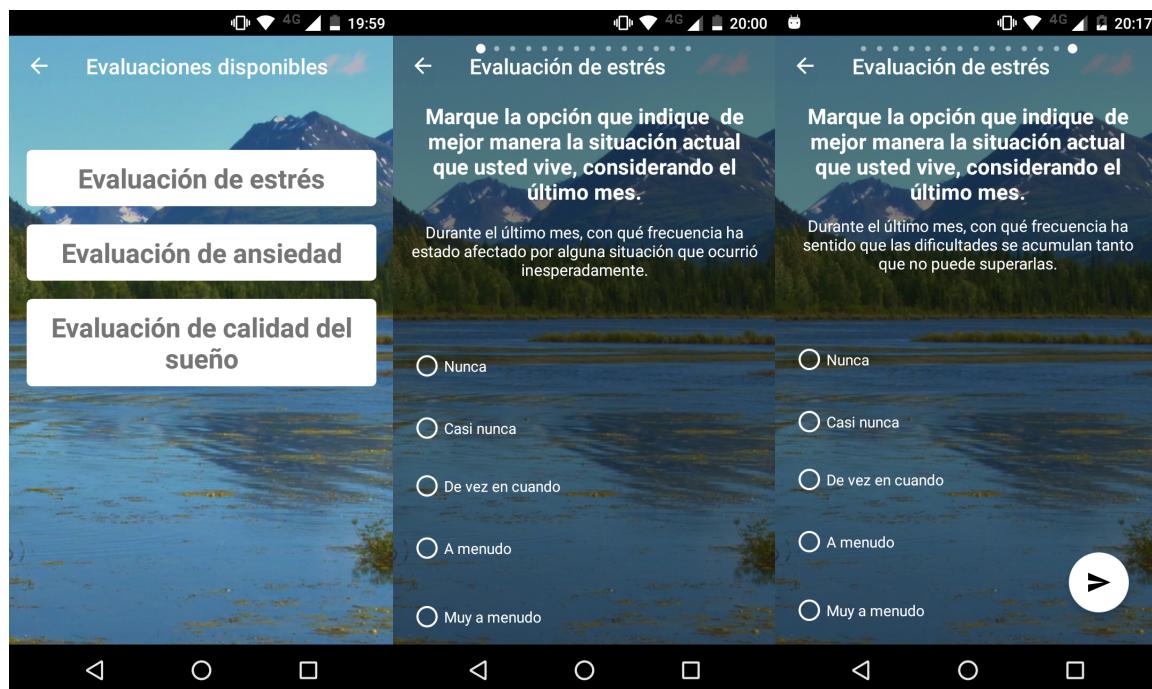


Figura 15: Interfaz de evaluaciones.

Capa de datos

Como se dijo anteriormente, la capa de datos maneja todo lo que es la persistencia y obtención de datos por parte del resto de los módulos. Esta capa encapsula la forma en que los datos están almacenados, la distribución de estos datos y el acceso a ellos. Cada módulo tiene una interfaz que tiene los métodos que provee el respectivo repositorio.

La arquitectura general de cada repositorio se muestra en la figura 16.

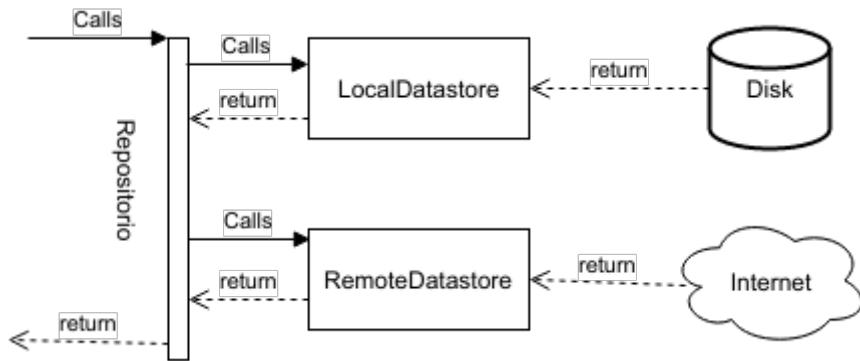


Figura 16: Arquitectura de repositorios.

Es decir, cada repositorio encapsula datos que se almacenan localmente y, opcionalmente, datos que se almacenan en remoto. Luego, la implementación del repositorio junta los datos locales y los que están en el servidor.

User Repository

La interfaz soportada por este módulo es la siguiente.

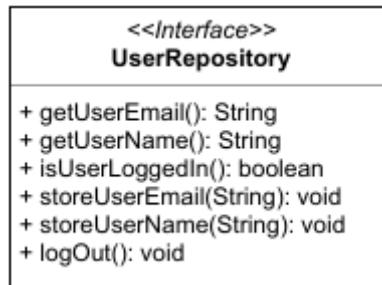


Figura 17: Interfaz de *UserRepository*.

Sequence Repository

La interfaz soportada por este módulo se presenta a continuación.

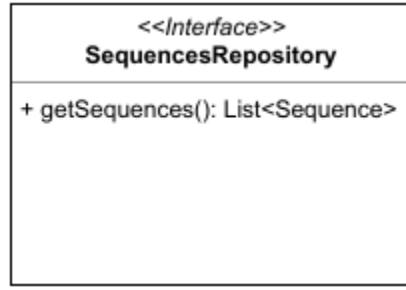


Figura 18: Interfaz de *SequencesRepository*.

El modelo de información asociado al módulo se muestra en la sección del Modelo de Datos.

AmbiencesRepository

La interfaz soportada por este módulo es la siguiente.

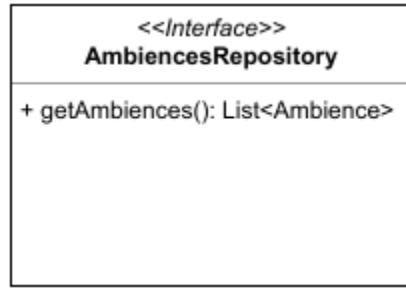


Figura 19: Interfaz de *AmbiencesRepository*.

El modelo de información asociado al módulo se muestra en la sección del Modelo de Datos.

ImageriesRepository

La interfaz soportada por este módulo es la siguiente.

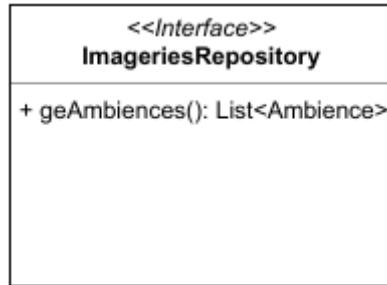


Figura 20: Interfaz de *ImageriesRepository*.

El modelo de información asociado al módulo se muestra en la sección del Modelo de Datos.

ExerciseProgramRepository

La interfaz soportada por este módulo es la siguiente.

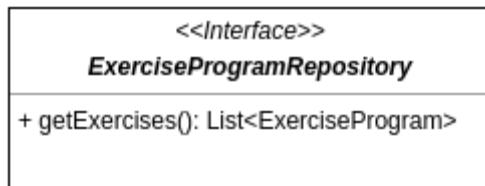


Figura 21: Interfaz de *ExerciseProgramRepository*.

El modelo de información asociado al módulo se muestra en la sección del Modelo de Datos.

3. Vista de Implementación

Bajo el SDK de Android, la unidad de deployment es un APK, que es una colección de funcionalidades construida, versionada e instalada como una unidad de implementación, y contiene, en general, múltiples archivos.

3.1.1 Estructura de la Aplicación

La aplicación se compone de un solo APK que soporta todo el Sistema móvil.

3.1.2 Arquitectura de Implementación

La arquitectura de implementación se compone de 10 packages

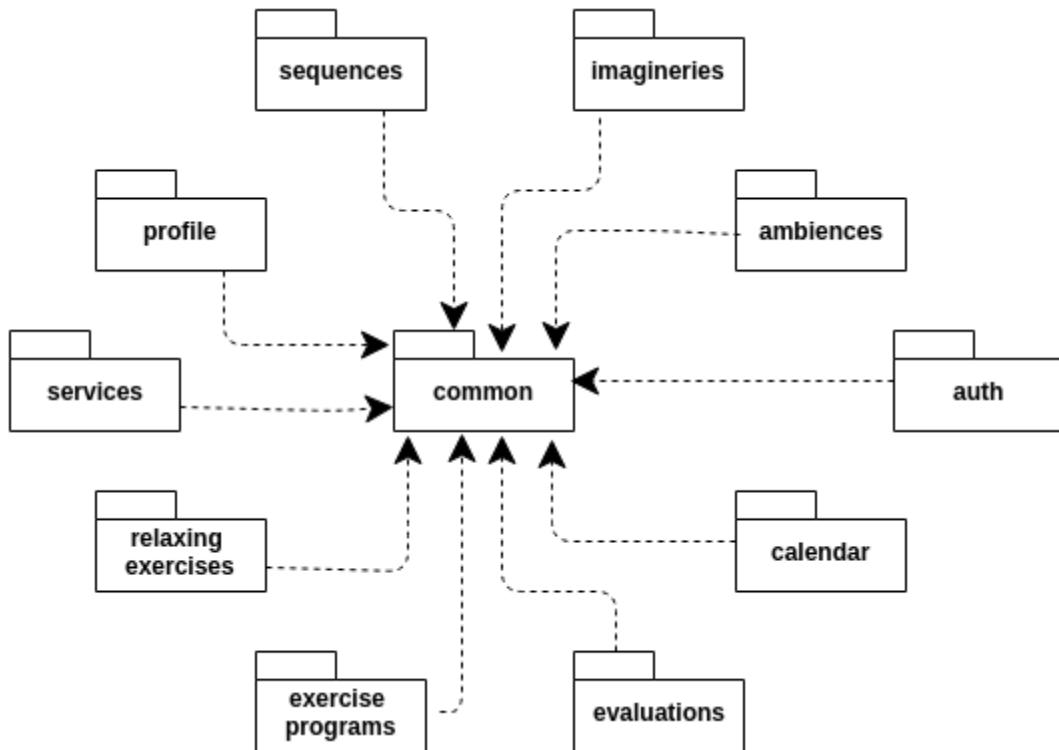


Figura 22: Arquitectura de implementación.

Capa de presentación

La capa de presentación de la aplicación cuenta con tres elementos fundamentales para la correcta representación gráfica de esta. Para el manejo de las 12 actividades, todas heredan una clase padre llamada “ *BaseActivity*” que permite tener un manejo común de ciertos elementos como la configuración del contenido, la carga del background, manejo de servicios, despliegue de notificaciones, etc. En cuanto al manejo de los fragments, todos heredan una clase padre llamada “ *BaseFragment*” que permite el manejo de los servicios asociados al fragment, despliegue de las notificaciones, y manejo de la actividad a la que se asocian. Finalmente, para el manejo de los *ListAdapters*, todas heredan de “ *BaseListAdapter*”, que permite que todos los adaptadores de las listas tengan un comportamiento en común, sin importar el origen de los datos a ser adaptados.

A continuación se presenta un diagrama que refleja la relación entre los tres elementos bases.

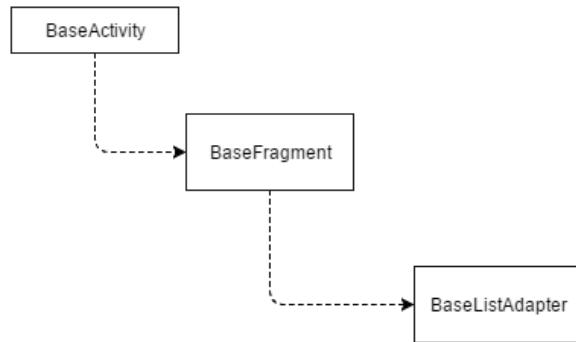


Figura 23: Estructura de clases base de la capa de presentación.

4. Vista de Datos

En esta sección se presenta el modelo de datos utilizado.

4.1.1 Modelo de Datos

La aplicación de Android, cuenta con su propio modelo de datos que se conecta con las base de datos almacenada en el servidor. Los diversos modelos utilizados en la aplicación, se muestran a continuación.

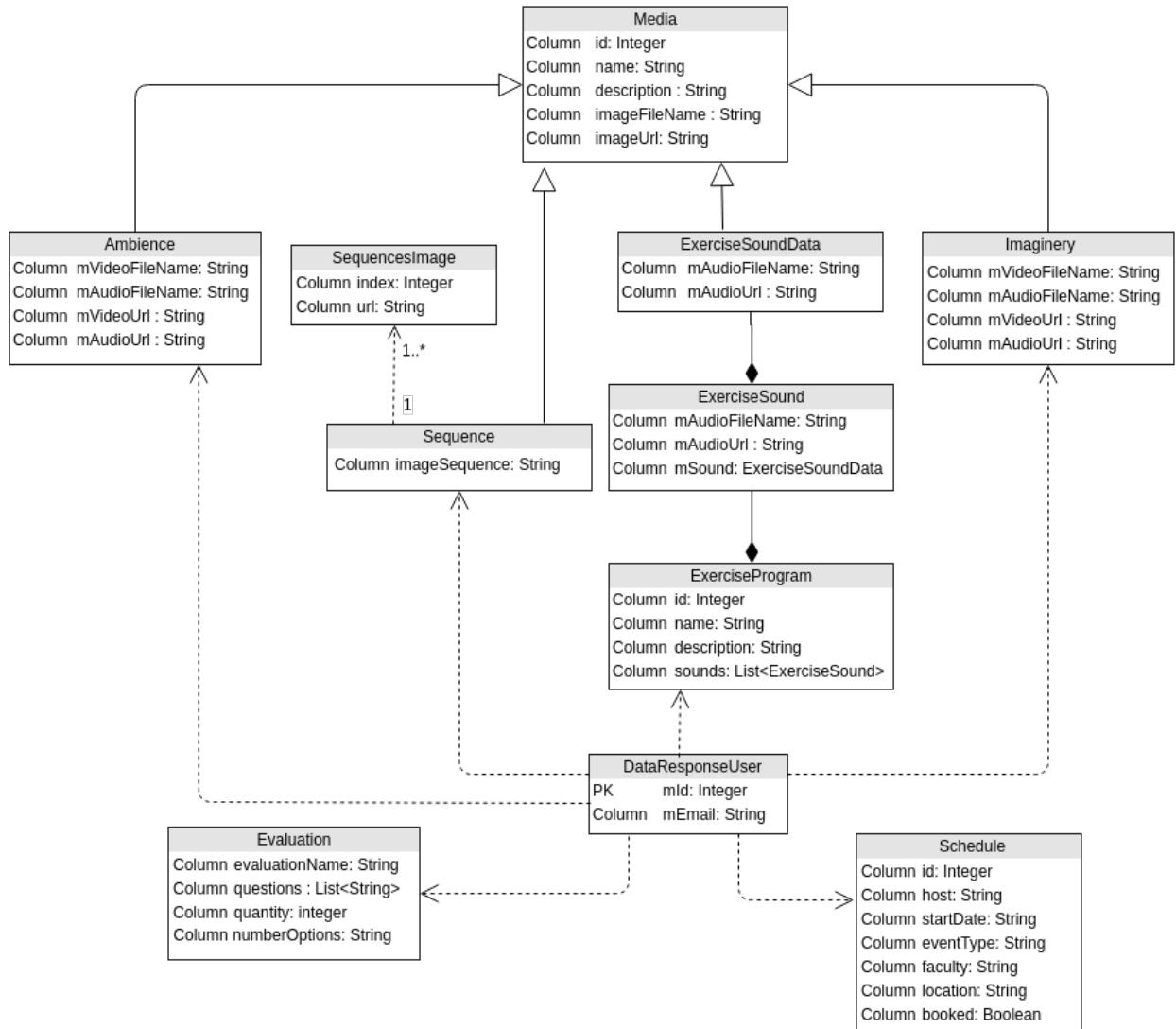


Figura 24: Modelo de datos de la aplicación.

5. Vista de Deployment

El deployment se hace en un celular Android con versión de Android SDK de 16 o mayor. La instalación se hace por medio de Google Play. Para ver la vista de deployment completa, ver el documento de diseño de web.

6. VRelaxUC Realidad Virtual

Para la realidad virtual se utilizó Unity 5.4.0f3 en conjunto con un SDK de oculus llamado Oculus Utilities V 1.8.

Primero se creó un proyecto en Unity que con una escena que estos Prefabs:

- OVRCameraRig: Es la cámara otorgada por el SDK de Oculus que permite visualizar a través de los lentes de manera correcta siguiendo la mirada del usuario. En esta cámara también se le carga un audio 3D para dar sonido ambiente usando un audio source y un audio listener.
- Sphere: Es una esfera que tiene un script que le volteo las normales, a esta esfera se le da como textura un video 360°. Este video al ser 360° queda envolviendo la esfera perfectamente y permite crear el ambiente, para hacer estas texturas especiales, se utilizo un asset de Unity llamado Easy Movie Texture.
- Unos canvas de unity que poseen botones con opciones, estos canvas poseen un script especial llamado OVR Raycaster que permite convertir la visión del usuario en un puntero para interactuar con los botones. Estos canvas son:
 - Menú Principal: Posee 3 botones para ir a las distintas secciones de la aplicación, estos botones son para videos 360°, imaginerías y ejercicios de respiración.
 - Menú de Videos: Este permite visualizar los distintos videos mientras vas cambiando el sonido ambiente. Para esto existen unos botones Siguiente y Atrás que te permiten navegar entre los distintos videos y audios.
 - Menú de Imaginerías: Este permite escuchar distintas imaginerías para relajarte mientras ves el paisaje. Para esto existen unos botones Siguiente y Atrás que te permiten navegar entre las distintas imaginerías.
 - Menú de Ejercicios de Respiración: Con esto podrás ver los distintos ejercicios de respiración y poder guiarte según los ejercicios para inhalar y exhalar adecuadamente mientras ves los videos 360°. Existen 2 botones, uno por cada ejercicio.

Además de esto cada menú posee un botón ocultar que te permite quitar el menú. Para poder volver al menú simplemente aprietas Tap y aparecerá. Cabe recalcar que todos los botones se aprietan mirando el botón hasta que aparezca el cursor arriba del botón y luego dando Tap.

Para la interacción de los menús con el paisaje se usa un script llamado Video Control que tiene funciones para manipular el audio, video y los ejercicios de respiración. Básicamente las funciones están dentro del script y luego los botones llaman funciones de este script.

