

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Desarrollo de Aplicación Android para la Administración de
Beneficiarios de Fundación “Ser Pa’ Hacer”**

Propuesta tecnológica

Giancarlo Pacini Peláez

Ingeniería en Sistemas

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 10 de mayo de 2018

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍA

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

**Desarrollo de Aplicación Android para la Administración de Beneficiarios de
Fundación “Ser Pa’ Hacer”**

Giancarlo Pacini Peláez

Calificación:

Nombre del profesor, Título académico

Fausto Pasmay, MS

Firma del profesor

Quito, 10 de mayo de 2018

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Giancarlo Pacini Peláez

Código: 00113885

Cédula de Identidad: 1713937918

Lugar y fecha: Quito, 10 de mayo de 2018

RESUMEN

Este documento describe el desarrollo de una aplicación móvil junto a un servidor de aplicaciones y base de datos. La aplicación se desarrolló para la fundación “Ser Pa’ Hacer”, ubicada en la ciudad de Quito. Se pretende presentarle una forma efectiva y organizada de mantener la información de la base de datos para la fundación, para que cambien su forma actual de almacenar la información. Se detalla las tecnologías usadas y como se usaron para llegar al objetivo., la estructura utilizada para construir el servidor y la aplicación, y las ventajas que presenta el usar esta estructura. Se presentan los casos de uso de la aplicación, las mejoras que se deberían implementar a futuro, y el uso completo que se le puede dar a ambos componentes y sus módulos.

Palabras Claves: Programación, Aplicación Móvil, Servidor, Modular.

ABSTRACT

This document describes the development of a mobile app and a server with a database engine. It was developed for “Ser Pa’ Hacer”, a non-profit organization located in Quito, Ecuador. The app is meant to be a simple and organized way for the organization to keep their beneficiaries database. It is an upgrade to their current way of keeping this information. The document specifies the technologies that were used for developing the application. It also details the structure used for developing both the application and the server, and what were the advantages in using this structure. Different use cases of the application are shown, what improvements the organization should do down the road, and the complete usage of the application.

Keywords: Software Development, Server, Application, Modular.

TABLA DE CONTENIDO

Introducción.....	8
Objetivos.....	9
Objetivo General.....	9
Objetivos Específicos.....	9
Desarrollo	10
Servidor.....	10
Base de Datos.....	12
Capa de Persistencia.....	15
Capa Lógica.....	16
Capa de Servicios.....	16
Aplicación Android.....	17
Conexión.....	18
Vistas.....	19
Navegación.....	19
Casos de Uso.....	20
Recomendaciones	23
Conclusiones	25
Anexos	¡Error! Marcador no definido.
Anexo #1 Requerimientos Funcionales	27
Anexo #2 Acta de Entrega-Recepción de la Fundación.....	29
Anexo #3 API de la aplicación	31
Anexo #4 Manual de Usuario.....	36

ÍNDICE DE FIGURAS

Figura 1 Capas y conexiones del servidor	12
Figura 2 Diagrama Entidad-Relación de las personas en base de datos	13
Figura 3 Diagrama Entidad-Relación de los usuarios en base de datos	15
Figura 4 Diagrama de conexión entre la aplicación y el servidor	18
Figura 5 Navegación por los principales módulos de la aplicación.	19
Figura 6 Diagrama de Casos de Uso de Acceso al Sistema	20
Figura 7 Diagrama de Casos de Uso de Manejo de Datos	21

INTRODUCCIÓN

La fundación “Ser Pa’ Hacer” queda ubicada en Quito, en el sector de la Carolina. Esta fundación da hogar a diferentes niños y jóvenes de provincias y de bajos recursos. También, les brinda educación y los ayuda en los deportes o artes en los cuales los jóvenes son más aptos. Tienen diferentes programas en los que pueden ser inscritos los jóvenes, entre esos destacan: Programa para profesionales con el Club el Nacional, el programa de educación para jugadores profesionales y las alianzas estratégicas con diferentes clubes a nivel nacional e internacional. La educación y el entrenamiento que recibe cada joven depende del programa en el que estén inscritos. Se brinda un servicio médico a cada joven, en el cual se hace un seguimiento normal, y una evolución de medidas y estado físico. Alrededor de 200 jóvenes están inscritos en los diferentes programas de la fundación, incluyendo el colegio.

Actualmente, el manejo de base de datos de los beneficiarios de la fundación se realiza en una base de datos de Microsoft Access. Se encuentra instalada en un computador que está dentro de la fundación. Esta base de datos no tiene forma de ser compartida entre distintos dispositivos, y depende de la computadora donde se encuentra instalada. La recolección de datos de los beneficiarios empieza desde los viajes a las provincias, y no se tiene una forma directa de ingresar esta información a la base de datos.

OBJETIVOS

Objetivo General

Crear un servidor de bases de datos para el manejo de la información de los beneficiarios y conectarlo a una aplicación Android donde se pueda visualizar y modificar la información.

Objetivos Específicos

- Acceso remoto a la información de los beneficiarios de la fundación.
- Ingreso remoto de los datos de los beneficiarios o posibles beneficiarios.
- Centralización de la base de datos.
- Acceso desde múltiples dispositivos.
- Seguridad de la información.

DESARROLLO

Se le propuso a la fundación “Ser Pa’ Hacer” el desarrollo de un servidor de información para poder acceder a los datos de los beneficiarios de forma remota. También, se propuso una aplicación Android que tenga acceso a este servidor, y desde la cual se pueda visualizar y modificar los datos de los beneficiarios. La fundación solicitó el acceso con roles y usuarios a diferentes segmentos de la aplicación, y la identificación de beneficiarios dependiendo de los distintos programas en los que están inscritos. La lista completa de requerimientos se puede visualizar en el ANEXO #1.

El desarrollo del proyecto se divide en dos aplicaciones principales. La primera, consta de un servidor el cual contiene toda la información sobre beneficiarios y usuarios de la fundación. A este se puede acceder de forma remota para visualizar la información. Tiene distintos servicios que sirven para modificar, visualizar o agregar información. La segunda, la aplicación móvil, ha sido desarrollada en Android y contiene todas las pantallas y funciones necesarias para acceder al servidor y poder visualizar, agregar y modificar la información de los beneficiarios.

Servidor

El servidor de la aplicación se desarrolló en Java. Es una aplicación de servidor EAR de tres capas. El lanzamiento de la aplicación es en un servidor Wildfly, descendiente de JBOSS. Se usó este debido a la integración que tiene con servicios RESTful mediante sus paquetes JAX-RS. También, las aplicaciones EAR son nativas de JBOSS, y permiten realizar una aplicación modular en la que cada módulo es independiente. Wildfly viene integrado con JPA (Java

Persistence API), que permite el uso de objetos para el manejo de base de datos. JPA está implementado sobre Hibernate que permite la integración con cualquier base de datos relacional. Se usaron todos estos paquetes que vienen integrados con la versión 11 de Wildfly.

El servidor es un paquete EAR compuesto de dos módulos principales. El primer módulo es el de persistencia y el segundo módulo es el de servicios. El módulo de persistencia es un archivo EJB que utiliza JPA para el acceso a base de datos y Beans de java para el manejo de la lógica de la aplicación. El módulo de servicios es un paquete WAR que implementa JAX-RS para crear un API de acceso al servidor. El siguiente gráfico muestra los módulos de la aplicación, cada capa dentro de los módulos, y la conexión de los módulos con el Internet y con la Base de Datos.

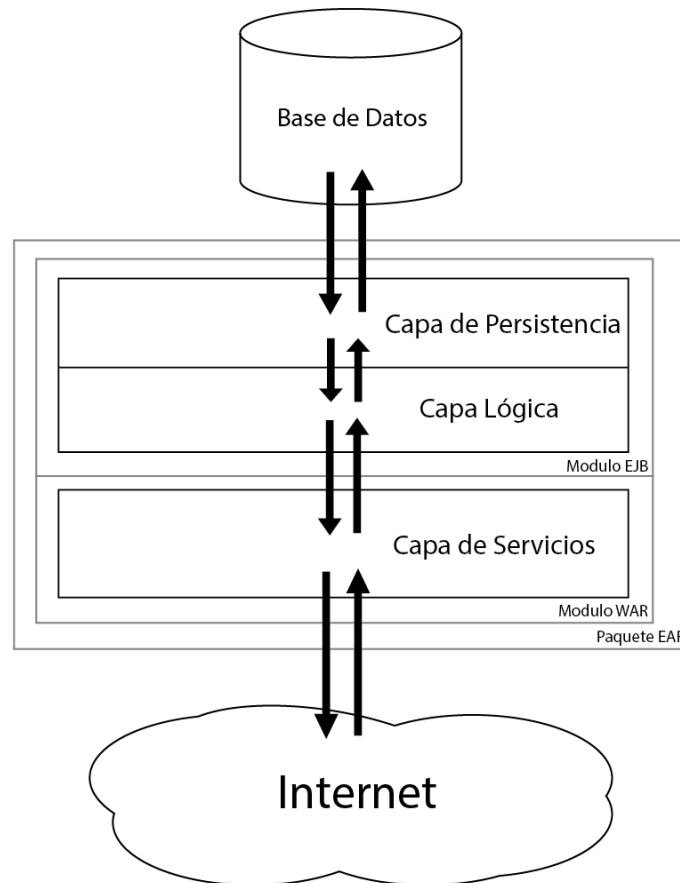


Figura 1 Capas y conexiones del servidor

Base de Datos.

Wildfly permite la integración con cualquier base de datos relacional mediante el uso de Hibernate. En este proyecto se usó el motor de base de datos MySQL, debido a que es gratuito, común y fácil de usar. La base de datos está dividida en dos grandes partes, el control de personas, o beneficiarios de la fundación; y, el control de usuarios que tienen acceso a la aplicación. Las Figuras 2 y 3 muestran los diagramas entidad-relación de cada una de las partes mencionadas.

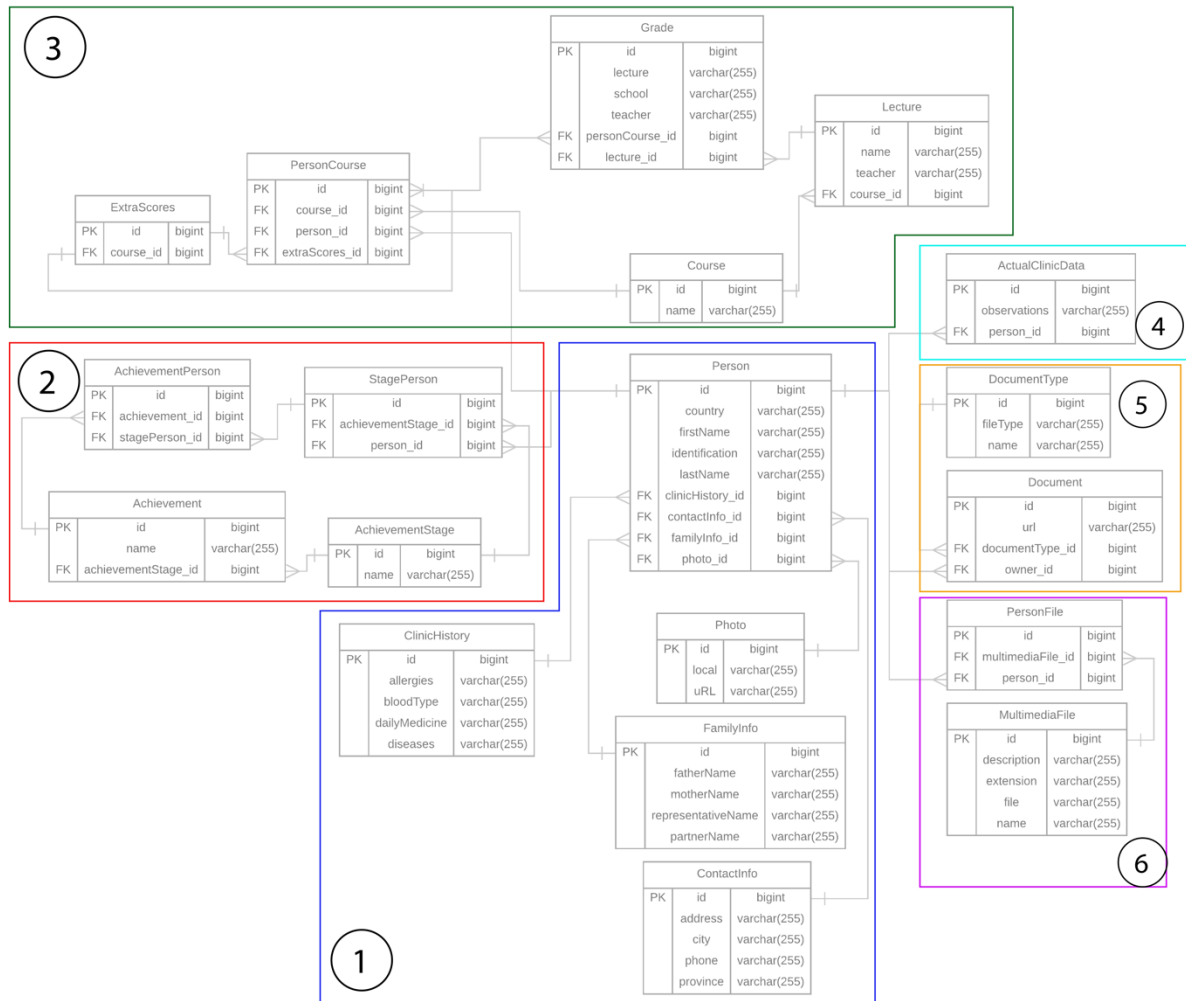


Figura 2 Diagrama Entidad-Relación de las personas en base de datos

La sección 1 corresponde a el manejo de personas en base de datos. Person es la tabla principal, en ella se encuentra la información necesaria para reconocer a una persona. Anexada a ella se encuentran ClinicHistory, que contiene información clínica necesaria de una persona. FamilyInfo contiene la información familiar de una persona. ContactInfo contiene la información de contacto de una persona. Y, Photo contiene el archivo local de la foto de una persona, y la dirección URL a este archivo. Estas cuatro tablas se encuentran separadas de la tabla persona debido a que no es información crucial para el resto de secciones de la aplicación, y enviar esta información junto a la persona implicaría más datos enviados, y un gasto de recursos del servidor.

La sección 2 contiene la información de logros de un beneficiario. AchievementStage corresponde a una etapa de logros y Achievement es cada logro dentro de una etapa. StagePerson es una etapa que está siendo ejecutada, o lo fue por una persona, y AchievementPerson es un logro dentro de un AchievementStage.

La información académica de un beneficiario se encuentra en la sección 3. Course corresponde a un año lectivo y Lecture a una materia dictada en ese año lectivo. PersonCourse es el curso que está cursando o cursó una persona, ExtraScores son sus notas de disciplina, presencia, puntualidad y responsabilidad; Grade son las notas correspondientes a una materia específica dentro del curso.

La sección 4 contiene la información de la evolución medica de una persona. Aquí se ingresan datos como la altura y el peso, las medidas de la persona, su presión arterial y observaciones del chequeo.

La sección 5 es la documentación de una persona. DocumentType es un documento específico que es requerido para una persona, y Document es la foto del documento.

La sección 6 contiene multimedia sobre un beneficiario. MultimediaFile contiene la información sobre un archivo que se encuentra en el servidor, así como un nombre y una descripción para el archivo. PersonFile asocia un archivo a una persona. Un archivo puede ser asociado a varias personas.

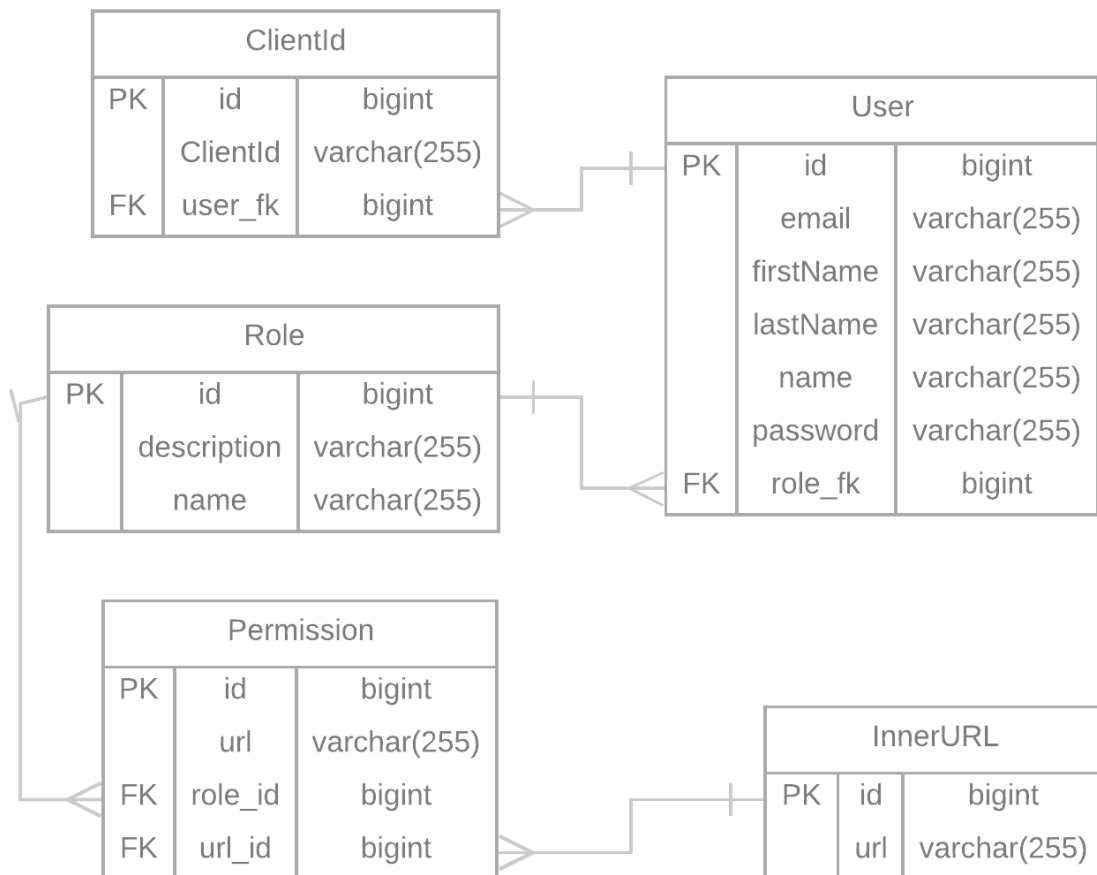


Figura 3 Diagrama Entidad-Relación de los usuarios en base de datos

User contiene la información de usuario. Role contiene la información específica de un Role, y se usa para dar los permisos de acceso a diferentes módulos de la aplicación. InnerURL representa una URL relativa de la aplicación que está ligada a un servicio. Permission es el permiso que tiene un rol a acceder a una InnerURL específica. ClientID se usa como un API_KEY para que el usuario no tenga que volver a ingresar su contraseña, y para evitar el robo de la contraseña en casos de baja seguridad.

Capa de Persistencia.

La capa de persistencia hace uso de JPA y de Hibernate. Dos herramientas que son proporcionadas por Wildfly como parte del servidor. JPA permite el uso de objetos como

representación lógica de las tablas en la base de datos. Modificando las propiedades de un objeto se puede persistir o modificar su entrada correspondiente en la base de datos. Hibernate es un servicio que permite la conexión con la base de datos, sin necesidad de conocer que motor de base de datos se está usando, y tampoco la sintaxis del mismo. La capa de persistencia está dividida en dos paquetes java principales. El primero contiene todas las clases que representan las distintas tablas en base de datos. El segundo, contiene todos los DAO (Data Access Object). Un DAO es una clase que instancia el manejador de persistencia de JPA, permitiendo guardar, modificar o eliminar objetos de base de datos. Existe un DAO por cada entidad; para poder identificar errores de forma más sencilla y para tener un código más limpio.

Capa Lógica.

La capa lógica hace uso de JavaEE y sus Beans para el manejo lógico de la aplicación. Esta capa se encarga de procesar los datos obtenidos en la capa de servicios y ajustarlos de forma necesaria para ser persistidos por la capa de persistencia. Esta capa se encarga de filtrar la información no necesaria y de validar la información recibida. Hace uso de excepciones cuando la información no está completa, o cuando existe un error en ella.

Existe un solo paquete principal que contiene todos los Beans utilizados por la aplicación. Cada uno de los objetos que pueden ser persistidos tienen su Bean correspondiente, así cada Bean se encarga de validar la información de un objeto en específico. Esto permite mayor facilidad de lectura del código, y encontrar problemas de forma más sencilla.

Capa de Servicios.

La capa de servicios está implementada con JAX-RS. Un módulo de Wildfly que permite la creación de servicios RESTful con el uso de anotaciones. Viene integrado con Wildfly y es

sencillo de usar. En el ANEXO #3 se puede encontrar la descripción del API de la aplicación, cada una de las URLs de acceso y los datos que recibe y retorna cada uno de los accesos.

Aplicación Android

El cliente es una aplicación Android. Esta desarrollada en React-Native, una librería desarrollada por Facebook para crear aplicaciones iOS y Android de forma simultánea. La aplicación está desarrollada en JSX, una sintaxis que mezcla JavaScript junto a XML para la generación de las vistas de la aplicación. La aplicación tiene 3 partes principales. La primera parte es el módulo de conexión con el servidor. La segunda parte son las vistas de la aplicación. La tercera parte es la navegación de la aplicación. La figura 4 muestra el diagrama de la conexión entre la aplicación y el servidor.

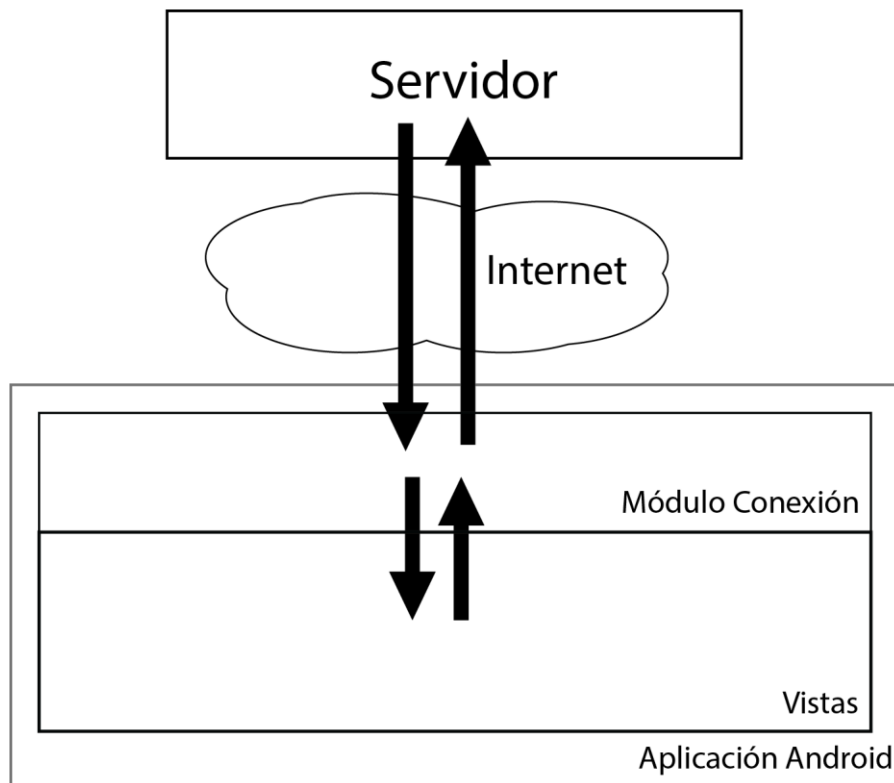


Figura 4 Diagrama de conexión entre la aplicación y el servidor

Conexión.

El módulo de conexión está hecho con la interfaz de JavaScript llamada Fetch. Esta permite abrir una conexión http o https con un servidor remoto mediante POST o GET. El módulo de conexión transforma el mensaje de la aplicación al formato requerido por el servidor, de la misma forma transforma el mensaje del servidor a un formato que pueda ser leído por el cliente. Las vistas proporcionan funciones de retorno por si la conexión fue un éxito, o si es que hubo un error. El módulo de conexión adjunta automáticamente los HEADERS requeridos por el servidor y se encarga de persistir la información del usuario de la aplicación.

Vistas.

Las vistas visualizan la información, y son el método de interacción con el usuario. Existen vistas para las diferentes secciones de la aplicación que permiten la creación y modificación de nuevo contenido. La descripción de las vistas se encuentra en el manual de usuario, el ANEXO #4.

Navegación.

La navegación entre vistas se realizó con la librería de navegación React Native Navigation, desarrollada por Wix. Aquí cada una de las vistas se registra bajo un seudónimo que luego puede ser accedido desde cualquier otra vista. La figura 5 muestra el flujo de la navegación en las principales vistas del sistema. El ANEXO #4 contiene el manual de usuario, el cual contiene la navegación completa por la aplicación.

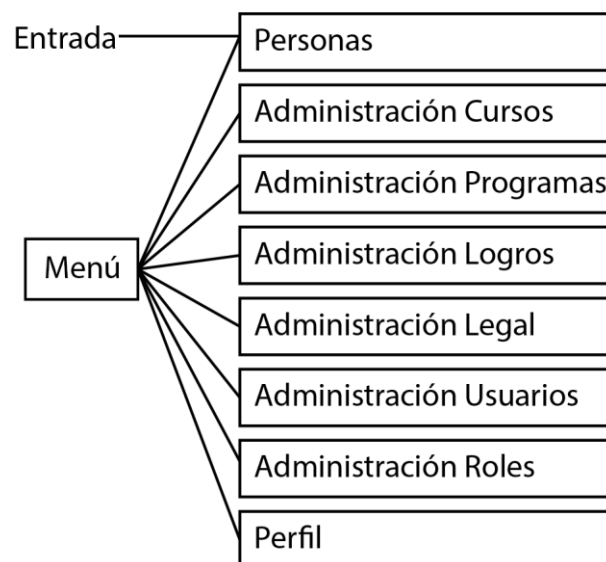


Figura 5 Navegación por los principales módulos de la aplicación.

Casos de Uso

Los casos de uso para el sistema se dividen en dos: acceso al sistema y manejo de información. Las dos figuras a continuación describen cada uno de los casos de uso.

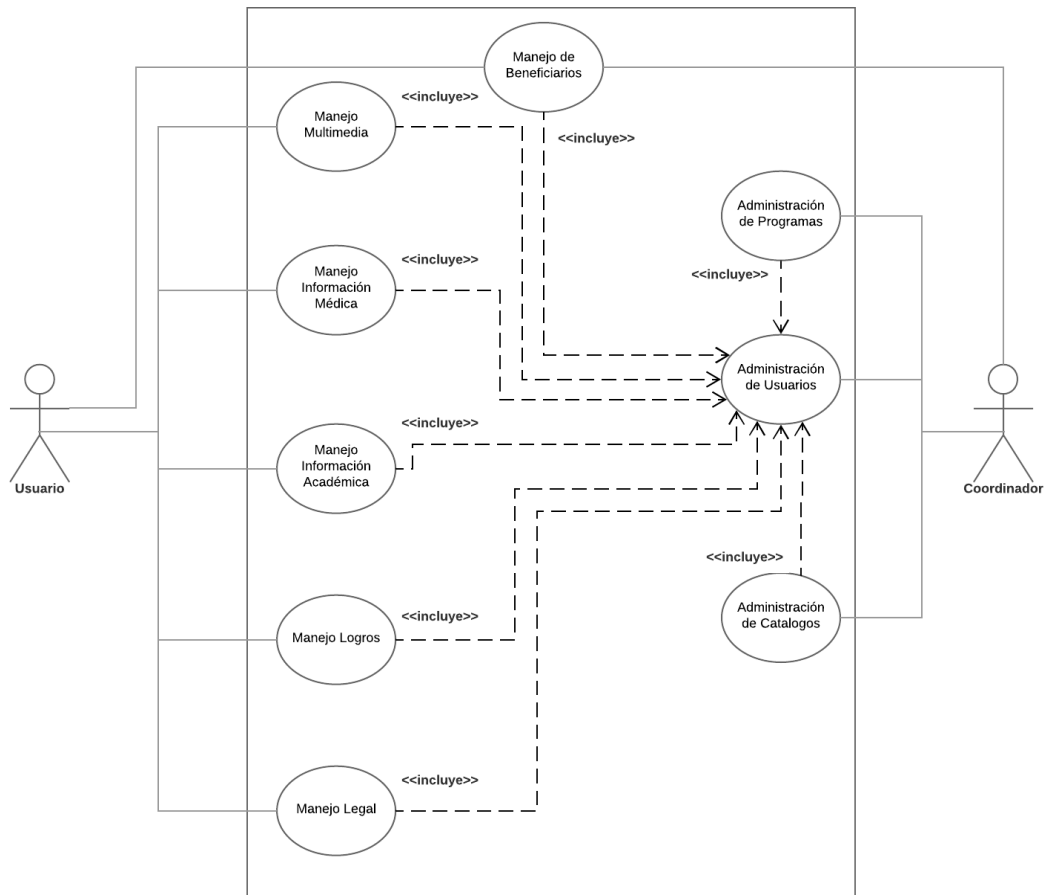


Figura 6 Diagrama de Casos de Uso de Acceso al Sistema

Como podemos ver en la figura, el acceso a cualquier sección de la aplicación depende de la administración de usuarios. El usuario coordinador, o administrador del sistema, crea roles y usuarios. Cada rol creado tiene acceso a diferentes secciones del sistema.

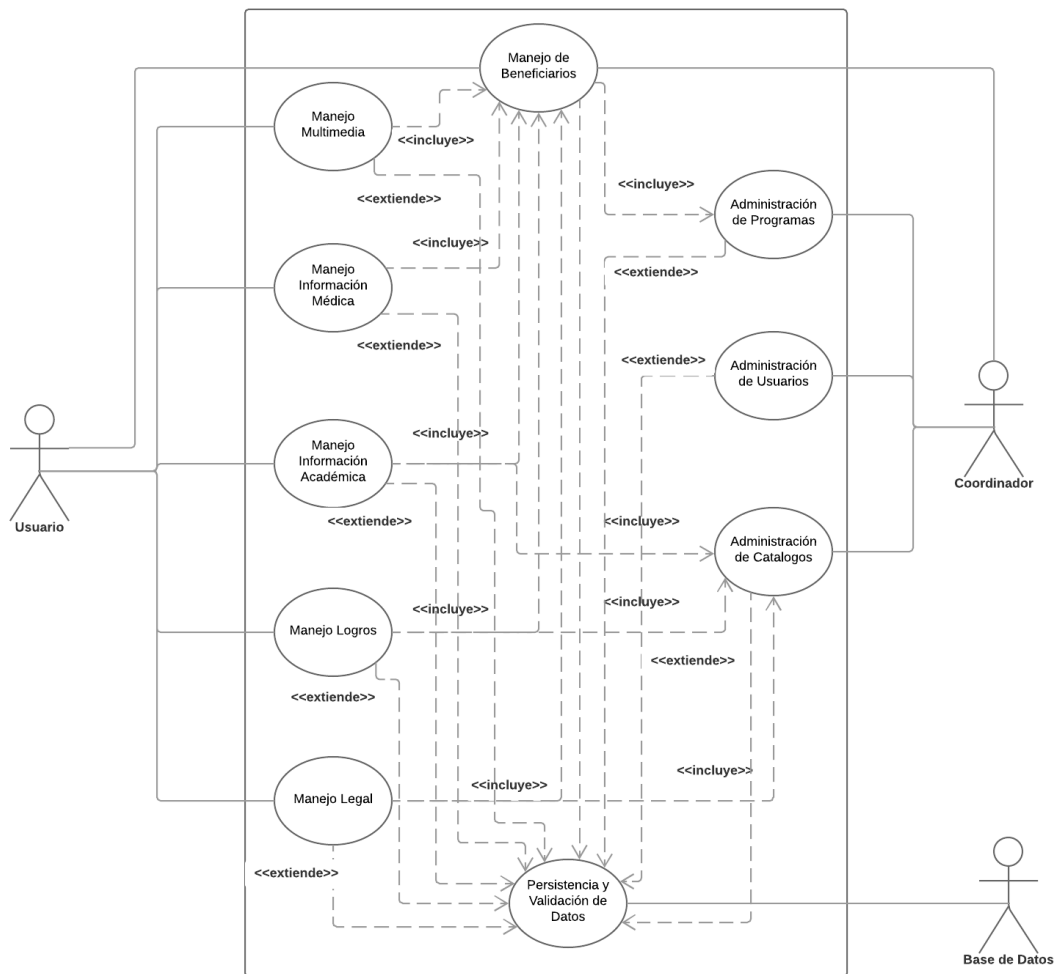


Figura 7 Diagrama de Casos de Uso de Manejo de Datos

Todas las secciones manejadas por el usuario dependen del manejo de beneficiarios. Esto es porque todos los datos manipulados son por beneficiario. El beneficiario depende de la administración de programas, debido a que la identificación de beneficiarios por programa es la función principal del sistema. Algunos módulos dependen de la administración de catálogos, debido a que necesitan tener catálogos predefinidos para poder funcionar correctamente.

Cada uno de los procesos del sistema extiende la persistencia y validación de datos, la cual es manejada por la Base de Datos. Las acciones y cambios realizados en cada uno de los

beneficiarios son guardados en base de datos para ser visualizada posteriormente. Todos estos datos se validan antes de ser persistidos.

RECOMENDACIONES

La aplicación presenta medidas de seguridad informática básicas. Las contraseñas son guardadas como un texto HASH de 512 caracteres. Implementa un API KEY, también llamado Client ID para evitar que la información de usuario de un cliente sea robada. A pesar de esto, se recomienda que la instalación del servidor sea implementada bajo SSL. Wildfly trae sus propias herramientas para la instalación de un certificado SSL, o también se puede adquirir un certificado SSL firmado por diferentes proveedores de seguridad informática reconocidos.

La implementación del servidor está desarrollada con una base de datos MySQL. Esto es debido a que este motor de base de datos es sencillo de usar, es gratuito, y fácil de implementar. Se recomienda que la implementación final sea bajo el motor de base de datos de preferencia del administrador del sistema, y no necesariamente MySQL. Existen diferentes motores que proveen ventajas sobre otros, y es responsabilidad del administrador identificar cual es el mejor motor para su caso. Si ya tiene un motor previamente instalado, puede usarlo. Wildfly tiene la flexibilidad de acoplarse al motor de base de datos que el administrador desee. Lo único que se requiere es agregar el driver java a la carpeta de configuraciones del servidor.

La aplicación está desarrollada para funcionar junto a un servidor Wildfly. Si el administrador del sistema tiene actualmente una instalación de otro servidor JavaEE, la aplicación puede ser instalada en este servidor. Para hacerlo, tiene que descargarse los paquetes de JavaEE, Hibernate, JPA y JAX-RS de sus dominios oficiales e integrarlos a la instalación del servidor.

Wildfly es un servidor JavaEE desarrollado por Red Hat Enterprise Linux. Se recomienda que el servidor sea instalado en un sistema operativo RHEL para evitar problemas de compatibilidad. Si es que el administrador del sistema tiene un equipo con otra distribución, Linux, Mac o Windows, existen versiones de Wildfly compatibles para cada una de ellas en la [página web oficial de Red Hat](#).

CONCLUSIONES

El desarrollo de una aplicación modular permite que los problemas en la aplicación se identifiquen de manera más sencilla. Los cambios son más fáciles de realizar, debido a que se identifica en que modulo se debe hacer, y cuál es el cambio que se debe hacer. Los cambios en base de datos son más largos de implementar, pero se puede asegurar que no va a haber consecuencias en el resto de información debido al cambio, ya que cada tabla maneja su información en un DAO y un Bean independientes.

Un servidor modular y una aplicación externa permiten la extensión de la aplicación a otras plataformas. Se puede implementar un nuevo paquete WAR, conectado al paquete EJB, para la creación de una página web. Se puede crear una aplicación especializada para dispositivos iOS que se conecte a los mismos servicios de la aplicación Android. Inclusive, si el administrador del sistema desea, se pueden crear una web Backend para el manejo de catálogos y datos administrativos, y eliminar este manejo de la aplicación web. El tiempo de desarrollo de estas aplicaciones se ve reducido debido a la estructura actual del servidor.

La implementación de la aplicación presenta un beneficio significativo para la fundación. Con ella, se puede realizar el levantamiento de datos en campo, con garantía de que la información va a persistir. La información puede ser accedida inmediatamente en la ciudad, para que el personal de la fundación pueda determinar quién puede ser un beneficiario de la fundación. Ahorrándose el trabajo de traer la información a la ciudad y transcribirla en el computador.

Los beneficiarios de la fundación pueden participar en los programas aliados a la fundación de manera más sencilla. Teniendo toda su información en un solo lugar, recopilada y presentada de una manera práctica, pueden presentar sus datos ante las alianzas de una mejor manera. Los filtros por programa también permiten al personal de la fundación identificar los beneficiarios más rápidamente.

El proceso de ingreso de datos de los beneficiarios es más óptimo. Los médicos pueden actualizar la evolución en consulta, y no tienen que transcribirla luego. Los profesores pueden agregar la información académica de los beneficiarios al final del año, ahorrando a la fundación el trabajo de transcribirla para cada estudiante. Cuando es necesitada, el acceso a la documentación del beneficiario es de forma inmediata y no hay que buscarlo en carpetas.

ANEXO #1 REQUERIMIENTOS FUNCIONALES

Requerimientos Funcionales Aplicación Android “Ser Pa Hacer”

- Aplicación desarrollada para Android.
- Ingreso con usuario y contraseña.
- Roles de Administración de usuarios y de colaboradores.
- Visualizar todas las personas inscritas en cualquiera de los programas de la fundación.
- Visualización de la ficha personal, deportiva, de salud y académica de las personas inscritas.
- Identificar a que programa pertenece la persona inscrita.
- Identificar si vive en la residencia de la fundación.
- Identificar si pertenece a alguna alianza estratégica.
- Poder mover a las personas entre los programas de la fundación.
- Poder crear nuevos beneficiarios e ingresar los datos de las fichas.
- Poder modificar las fichas en los casos necesarios.

Yo, Galo Flores, director de la fundación “Ser Pa Hacer”, estoy de acuerdo con los requerimientos ofrecidos para la aplicación móvil.

Galo Flores

CI:

**ANEXO #2 ACTA DE ENTREGA-RECEPCIÓN DE LA
FUNDACIÓN**

Quito, 2 de abril de 2018

Acta de Entrega-Recepción de Software

Sr. Galo Flores, director de la fundación "Ser Pa' Hacer"

Certifico:

Que han sido recibidos con entera satisfacción los siguientes artículos, desarrollados por el Sr. Giancarlo Pacini.

Número	Descripción
1	Código del servidor aplicación "Ser Pa' Hacer".
2	Código de la aplicación Android "Ser Pa' Hacer".
3	Manual sobre instalación y uso de la aplicación.

Galo Flores

Director "Ser Pa' Hacer"

CI:

Giancarlo Pacini

Desarrollador

CI:

ANEXO #3 API DE LA APLICACIÓN

URL	Método	Módulo	Descripción	Recibe	Retorna
/extrascorress/update	POST	Académico	Actualizar Notas de Curso	JSON	JSON
/grades/.*?	GET	Académico	Ver Calificaciones	JSON	JSON
/grades/.*?/possible	GET	Académico	Ver Materias Por coger un Curso	JSON	JSON
/grades/create	POST	Académico	Anadir Materia a Persona	JSON	JSON
/grades/update	POST	Académico	Actualizar Calificaciones	JSON	JSON
/personcourses/.*?	GET	Académico	Ver Cursos	JSON	JSON
/personcourses/.*?/possible	GET	Académico	Ver Cursos Disponibles	JSON	JSON
/personcourses/create	POST	Académico	Agregar curso a persona	JSON	JSON
/personcourses/single/.*?	GET	Académico	Ver curso	JSON	JSON
/personcourses/update	POST	Académico	Actualizar curso de persona	JSON	JSON
/courses	GET	Admin/Cursos	Ver Cursos	JSON	JSON
/courses/.*?	GET	Admin/Cursos	Ver Información de Curso	JSON	JSON
/courses/changeActive	POST	Admin/Cursos	Des/Activar Cursos	JSON	JSON
/courses/create	POST	Admin/Cursos	Crear Cursos	JSON	JSON
/courses/update	POST	Admin/Cursos	Actualizar Cursos	JSON	JSON

/lectures/.*?	GET	Admin/Cursos	Ver Materias	JSON	JSON
/lectures/changeActive	POST	Admin/Cursos	Des/Activar Materias	JSON	JSON
/lectures/create	POST	Admin/Cursos	Crear Materias	JSON	JSON
/lectures/update	POST	Admin/Cursos	Actualizar	JSON	JSON
/documenttypes/.*?	GET	Admin/Legal	Ver Tipo Documentos	JSON	JSON
/documenttypes/create	POST	Admin/Legal	Crear Tipos Documentos	JSON	JSON
/documenttypes/update	POST	Admin/Legal	Actualizar Tipos Documentos	JSON	JSON
/achievements/.*?	GET	Admin/Logros	Ver Logros	JSON	JSON
/achievements/changeActive	POST	Admin/Logros	Des/Activar Logros	JSON	JSON
/achievements/create	POST	Admin/Logros	Crear Logros	JSON	JSON
/achievements/update	POST	Admin/Logros	Actualizar Logros	JSON	JSON
/achievementstages	GET	Admin/Logros	Ver Todas las Etapas	JSON	JSON
/achievementstages/.*?	GET	Admin/Logros	Ver Etapa	JSON	JSON
/achievementstages/changeActive	POST	Admin/Logros	Des/Activar Etapas	JSON	JSON
/achievementstages/create	POST	Admin/Logros	Crear Etapas	JSON	JSON

/achievementstages/update	POST	Admin/Logros	Actualizar Etapas	JSON	JSON
/programs/changeActive	POST	Admin/Programas	Des/Activar Programas	JSON	JSON
/programs/create	POST	Admin/Programas	Crear Programas	JSON	JSON
/programs/update	POST	Admin/Programas	Actualizar Programas	JSON	JSON
/admin/roles	GET	Admin/Roles	Ver Roles	JSON	JSON
/admin/roles/changeActive	POST	Admin/Roles	Des/Activar Roles	JSON	JSON
/admin/roles/create	POST	Admin/Roles	Crear Roles	JSON	JSON
/admin/roles/update	POST	Admin/Roles	Actualizar Roles	JSON	JSON
/admin/users	GET	Admin/Usuarios	Ver Usuarios	JSON	JSON
/admin/users/.*?	GET	Admin/Usuarios	Ver Por ID	JSON	JSON
/admin/users/changeActive	POST	Admin/Usuarios	Des/Activar Usuarios	JSON	JSON
/admin/users/create	POST	Admin/Usuarios	Crear Usuarios	JSON	JSON
/admin/users/resetPassword	POST	Admin/Usuarios	Resetear Contraseñas	JSON	JSON
/admin/users/update	POST	Admin/Usuarios	Actualizar Usuarios	JSON	JSON
/actualclinicdatas/.*?	GET	Clínica	Ver Evolución Clínica	JSON	JSON
/actualclinicdatas/create	POST	Clínica	Crear Evolución Clínica	JSON	JSON

/actualclinicdatas/update	POST	Clínica	Actualizar Evolución Clínica	JSON	JSON
/documents/.*?	GET	Legal	Ver Documentos	JSON	JSON
/documents/upload	POST	Legal	Subir Documentos	JSON	JSON
/documenttypes	GET	Legal	Ver Tipos Documentos	JSON	JSON
/login	POST	Login	Login	JSON	JSON
/achievementpersons/.*?	GET	Logros	Ver Logros por Etapa	JSON	JSON
/achievementpersons/complete	POST	Logros	Completar Logro	JSON	JSON
/stagepersons/.*?	GET	Logros	Ver Etapas	JSON	JSON
/stagepersons/complete	POST	Logros	Completar Etapa	JSON	JSON
/multimediafiles/.*?	GET	Multimedia	Ver Archivo Multimedia	JSON	JSON
/multimediafiles/upload	POST	Multimedia	Subir Multimedia	JSON	JSON
/profile	GET	Perfil	Ver Perfil de Usuario	JSON	JSON
/profile/password	POST	Perfil	Cambiar Contraseña	JSON	JSON
/profile/update	POST	Perfil	Actualizar Perfil de Usuario	JSON	JSON
/permissions/.*?	GET	Permisos	Ver Permisos Por Rol	JSON	JSON
/permissions/save	POST	Permisos	Guardar permisos de un rol	JSON	JSON
/clinichistorys	GET	Personas	Ver Historias Clínica	JSON	JSON
/clinichistorys/.*?	GET	Personas	Ver Historia Clínica	JSON	JSON
/clinichistorys/create	POST	Personas	Crear Historia Clínica	JSON	JSON
/clinichistorys/update	POST	Personas	Actualizar Historia Clínica	JSON	JSON
/contactinfos	GET	Personas	Ver Contactos	JSON	JSON

/contactinfos/.*?	GET	Personas	Ver Contacto	JSON	JSON
/contactinfos/create	POST	Personas	Crear Contacto	JSON	JSON
/contactinfos/update	POST	Personas	Actualizar Contacto	JSON	JSON
/familyinfos	GET	Personas	Ver Familias	JSON	JSON
/familyinfos/.*?	GET	Personas	Ver Familia	JSON	JSON
/familyinfos/create	POST	Personas	Crear Familia	JSON	JSON
/familyinfos/update	POST	Personas	Actualizar Familia	JSON	JSON
/persons	GET	Personas	Ver Personas	JSON	JSON
/persons/.*?	GET	Personas	Ver Persona	JSON	JSON
/persons/create	POST	Personas	Crear Personas	JSON	JSON
/persons/update	POST	Personas	Actualizar Personas	JSON	JSON
/persons/upload	POST	Personas	Subir Foto	JSON	JSON
/programpersons/.*?	GET	Programas	Ver Programas	JSON	JSON
/programpersons/complete	POST	Programas	Completar Programas	JSON	JSON
/programpersons/persons/.*?	GET	Programas	Ver Personas por programa	JSON	JSON
/programs	GET	Programas	Ver Programas	JSON	JSON
/programs/.*?	GET	Programas	Ver Un Programa	JSON	JSON

ANEXO #4 ÍNDICE DE CÓDIGO FUENTE

serphacer-ejb	
com.gpacini.serpahacer.data.admin	
ClientId.java	Entidad de ClientID
InnerURL.java	Entidad que representa una URL del sistema
Permission.java	Entidad que representa un permiso por rol y url
ProjectVariable.java	Entidad para guardar variables del proyecto
Role.java	Entidad que representa un rol en la aplicacion
User.java	Entidad que representa un usuario en la aplicacion
com.gpacini.serpahacer.data.app.academic	
Course.java	Entidad que representa un curso académico.
ExtraScores.java	Entidad que representa las notas extras de un curso académico.
Grade.java	Entidad que representa una materia tomada.
Lecture.java	Entidad que representa una materia dictada.
PersonCourse.java	Entidad que representa un curso tomado por una persona.
com.gpacini.serpahacer.data.app.achievements	
Achievement.java	Entidad que representa un logro.
AchievementPerson.java	Entidad que representa un logro de una persona.
AchievementStage.java	Entidad que representa un logro en una etapa.
StagePerson.java	Entidad que representa una etapa de una persona.
com.gpacini.serpahacer.data.app.contact	
ContactInfo.java	Entidad que representa la información de contacto de una persona.
FamilyInfo.java	Entidad que representa la información familiar de una persona.
Person.java	Entidad que representa una persona.
Photo.java	Entidad que representa la foto de una persona.
com.gpacini.serpahacer.data.app.contact.enumeration	

CivilStatusEnum.java	Enumeración de estados civiles.
com.gpacini.serpahacer.data.app.health	
ActualClinicData.java	Entidad que representa una visita clínica de una persona.
ClinicHistory.java	Entidad que representa la historia clínica de una persona.
com.gpacini.serpahacer.data.app.legal	
Document.java	Entidad que representa un documento legal.
DocumentType.java	Enumeración de los tipos de documentos.
com.gpacini.serpahacer.data.app.multimedia	
MultimediaFile.java	Entidad que representa un archivo multimedia
com.gpacini.serpahacer.data.app.program	
Program.java	Entidad que representa un programa de la fundación
ProgramPerson.java	Entidad que representa la relación entre un programa y una persona
com.gpacini.serpahacer.model.admin	
DAOJPAClientId.java	DAO para manipular ClientIds
DAOJPAInnerURL.java	DAO para manipular URLs
DAOJPAPermission.java	DAO para manipular permisos
DAOJPAProjectVariable.java	DAO para manipular variables del sistema
DAOJPARole.java	DAO para manipular roles
DAOJPAUser.java	DAO para manipular Usuarios
com.gpacini.serpahacer.model.app.academic	
DAOJPACourse.java	DAO para manipular cursos
DAOJPAExtraScores.java	DAO para manipular notas de cursos
DAOJPAGrade.java	DAO para manipular notas de materias
DAOJPAlecture.java	DAO para manipular materias
DAOJPAPersonCourse.java	DAO para manipular cursos tomados por personas
com.gpacini.serpahacer.model.app.achievements	
DAOJPAAchievement.java	DAO para manipular logros
DAOJPAAchievementPerson.java	DAO para manipular logros por persona
DAOJPAAchievementStage.java	DAO para manipular etapas de logros por persona
DAOJPASTagePerson.java	DAO para manipular etapas de logros
com.gpacini.serpahacer.model.app.contact	
DAOJPAContactInfo.java	DAO para manipular la información de contacto de una persona.

DAOJPFamilyInfo.java	DAO para manipular la información familiar de una persona
DAOJPAPerson.java	DAO para manipular una persona.
DAOJPAPhoto.java	DAO para manipular la foto de una persona.
com.gpacini.serpahacer.model.app.health	
DAOJPAActualClinicData.java	DAO para manipular consultas médicas.
DAOJPAClinicHistory.java	DAO para manipular historia clínica.
com.gpacini.serpahacer.model.app.legal	
DAOJPADocument.java	DAO para manipular documentos legales.
com.gpacini.serpahacer.model.app.multimedia	
DAOJPAMultimediaFile.java	DAO para manipular archivos multimedia
com.gpacini.serpahacer.model.app.program	
DAOJPAProgram.java	DAO para manipular programas.
DAOJPAProgramPerson.java	DAO para manipular programas por personas.
com.gpacini.serpahacer.model.common	
DAOGeneric.java	Interfaz de un DAO genérico
DAOJPABase.java	DAO Abstracto con métodos básicos
com.gpacini.serpahacer.service.admin	
ClientIDBean.java	Lógica de los ClientID.
InnerURLBean.java	Lógica de los URLs.
PermissionBean.java	Lógica de los permisos.
ProjectVariableBean.java	Lógica de las variables del sistema
RoleBean.java	Lógica de los roles.
UserBean.java	Lógica de los usuarios.
com.gpacini.serpahacer.service.app.academic	
CourseBean.java	Lógica de los cursos.
ExtraScoresBean.java	Lógica de las notas de un curso.
GradeBean.java	Lógica de las notas de una materia.
LectureBean.java	Lógica de las materias.
PersonCourseBean.java	Lógica de los cursos por personas.
com.gpacini.serpahacer.service.app.achievements	
AchievementBean.java	Lógica de los logros.
AchievementPersonBean.java	Lógica de los logros por persona.
AchievementStageBean.java	Lógica de las etapas de logros.
StagePersonBean.java	Lógica de las etapas por persona.
com.gpacini.serpahacer.service.app.contact	
ContactInfoBean.java	Lógica de la información de contacto.
FamilyInfoBean.java	Lógica de la información familiar.
PersonBean.java	Lógica de la información de una persona.

PhotoBean.java	Lógica de una foto.
com.gpacini.serpahacer.service.app.health	
ActualClinicDataBean.java	Lógica de las consultas médicas.
ClinicHistoryBean.java	Lógica de la historia clínica.
com.gpacini.serpahacer.service.app.legal	
DocumentBean.java	Lógica de los documentos legales.
com.gpacini.serpahacer.service.app.multimedia	
MultimediaFileBean.java	Lógica de los archivos multimedia.
com.gpacini.serpahacer.service.app.program	
ProgramBean.java	Lógica de los programas.
ProgramPersonBean.java	Lógica de los programas por persona.
com.gpacini.serpahacer.service.login	
LoginBean.java	Lógica para el Login al sistema de un usuario.
com.gpacini.serpahacer.util.ejb	
DateUtilities.java	Utilidades para manipulación de fechas.
JSONReturnable.java	Interfaz que representa un objeto que puede ser convertido a un objeto JSON.
NumberUtilities.java	Utilidades para la manipulación de números.
StringUtilities.java	Utilidades para la manipulación de Strings.
serphacer-web	
com.gpacini.serpahacer.filters	
DebugFilter.java	Filtro de Salida del sistema. Se agregan Headers y se guardan Logs.
PermissionFilter.java	Filtro de entrada al sistema. Verifica permisos del usuario.
com.gpacini.serpahacer.rest	
JaxRsActivator.java	Activador para que funcione REST.
com.gpacini.serpahacer.services.admin	
PermissionService.java	Servicios para los permisos.
ProfileService.java	Servicios del perfil de usuario.
RoleService.java	Servicios de roles.
UserService.java	Servicios de usuarios
com.gpacini.serpahacer.services.app.academic	
CourseService.java	Servicios de cursos.
ExtraScoresService.java	Servicios de notas adicionales.
GradeService.java	Servicios de notas por materia.
LectureService.java	Servicios de materias.
PersonCourseService.java	Servicios de personas por curso.

com.gpacini.serpahacer.services.app.achievements	
AchievementPersonService.java	Servicios de logros por persona.
AchievementService.java	Servicios de logros.
AchievementStageService.java	Servicios de etapas de logros.
StagePersonService.java	Servicios de etapas por personas.
com.gpacini.serpahacer.services.app.contact	
ContactInfoService.java	Servicios de información de contacto.
FamilyInfoService.java	Servicios de información familiar.
PersonService.java	Servicios de personas.
com.gpacini.serpahacer.services.app.health	
ActualClinicDataService.java	Servicios de consultas médicas.
ClinicHistoryService.java	Servicios de historias clínicas.
com.gpacini.serpahacer.services.app.legal	
DocumentService.java	Servicios de documentos legales.
com.gpacini.serpahacer.services.app.multimedia	
MultimediaFileService.java	Servicios de archivos multimedia.
com.gpacini.serpahacer.services.app.program	
ProgramPersonService.java	Servicios de programas por personas.
ProgramService.java	Servicios de programas.
com.gpacini.serpahacer.services.login	
LoginService.java	Servicios para el login al sistema.
com.gpacini.serpahacer.util.web	
CommonJSONInput.java	Interfaz que representa una entrada a un servicio en forma de JSON.
CommonJSONResponse.java	Interfaz que representa una respuesta de un servicio en forma de JSON.
FileSaver.java	Utilidad para guardar archivos en el disco duro.

ANEXO #5 MANUAL DE USUARIO

MANUAL DE USUARIO: APLICACIÓN “SER PA’ HACER”

Requerimientos

- Contar con un dispositivo Android.
- Descargarse la aplicación desde <http://104.131.25.120:8080/serpahacer.apk>
- Solicitar un usuario y contraseña al administrador del sistema.

Índice

Manual de Usuario: Aplicación “Ser Pa’ Hacer”	41
Requerimientos	41
Índice	42
Acceso al sistema	47
Menú y Permisos	47
Personas.....	50
Filtrar Por Programa.....	51
Agregar o Editar Persona.....	52
Flujo para la edición o creación de beneficiarios.....	52
Información de Contacto.....	53
Información Familiar	54
Historia Clínica.....	55
Foto del Beneficiario	56
Información de Beneficiario	57
Historial Clínico.....	61
Historia Académica	63
Agregar Un Curso.....	64
Ver Curso.....	65
Notas generales del curso.....	66
Agregar Materia.....	67
Ver Materia.....	68
Calcular Nota Final.....	68

Logros.....	69
Legal.....	71
Multimedia.....	74
Programas Inscrito	76
Administración del Sistema	77
Administración de Roles.....	77
Administración de Usuarios.	79
Administración Legal.	82
Administración de Logros.....	84
Administración de Programas.	88
Administración de Cursos.....	90
Perfil	94

Índice De Figuras

Figura 1 Pantalla de Ingreso al Sistema	47
Figura 2 Menú del sistema.....	48
Figura 3 Visualización de Personas	50
Figura 4 Pantalla para filtrar beneficiarios por Programa	51
Figura 5 Agregar Persona.....	52
Figura 6 Información de Contacto de Beneficiario	53
Figura 7 Información Familiar de Beneficiario.....	54
Figura 8 Historia Clínica del Beneficiario	55
Figura 9 Pantalla Para Subir Foto de Un Beneficiario	56
Figura 10 Pantalla de Información de Beneficiario.....	57
Figura 11 Botones de la información de beneficiario.....	58
Figura 12 Información de Beneficiario.....	60
Figura 13 Pantalla Historial Clínico	61
Figura 14 Consulta Médica	62
Figura 15 Historial Académico de Beneficiario	63
Figura 16 Pantalla para agregar un curso	64
Figura 17 Visualización de Curso	65
Figura 18 Notas Generales del Curso.....	66

Figura 19 Agregar Materia a Curso	67
Figura 20 Visualización de Materia	68
Figura 21 Visualización Logros del Beneficiario	69
Figura 22 Visualización Etapa.....	70
Figura 23 Visualización Documentos Legales de Beneficiario	71
Figura 24 Seleccionar Documento	72
Figura 25 Subir Nuevo Documento.....	73
Figura 26 Visualización Multimedia	74
Figura 27 Subir Archivo Multimedia	75
Figura 28 Visualización de Programas Inscritos del Beneficiario.....	76
Figura 29 Administración de Roles	77
Figura 30 Pantalla de Edición de Roles	78
Figura 31 Administración de Usuarios	79
Figura 32 Pantalla de Creación de Usuarios	80
Figura 33 Pantalla de Edición de Usuarios.....	81
Figura 34 Pantalla de Administración Legal.....	82
Figura 35 Pantalla de Edición y Creación de Documentos Legales.....	83
Figura 36 Pantalla de Administración de Logros	84
Figura 37 Pantalla de Creación de Etapas.....	85

Figura 38 Pantalla de Visualización de Etapas	86
Figura 39 Pantalla de Creación y Edición de Logros	87
Figura 40 Administración de Programas.....	88
Figura 41 Pantalla de Edición y Creación de Programas.....	89
Figura 42 Administración de Cursos	90
Figura 43 Pantalla de Creación de Cursos.....	91
Figura 44 Pantalla de Visualización de Curso	92
Figura 45 Pantalla de Creación y Edición de Materias.....	93
Figura 46 Vista superior de la pantalla de Perfil	94
Figura 47 Vista inferior de la pantalla de Perfil	95

Acceso al sistema

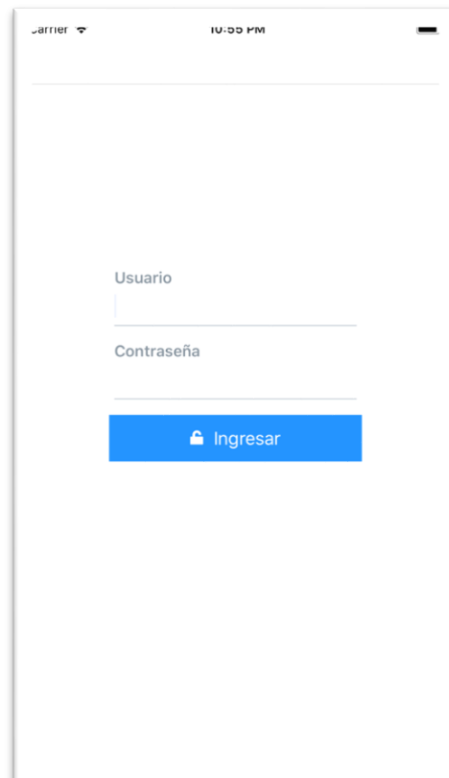
The image shows a mobile application login screen. At the top, the status bar displays 'Carrier' and '10:55 PM'. The main content area is white and contains two text input fields. The first field is labeled 'Usuario' and the second is labeled 'Contraseña'. Below these fields is a blue button with a white lock icon and the text 'Ingresar'.

Figura 8 Pantalla de Ingreso al Sistema

En la pantalla de ingreso al sistema se deben ingresar el usuario y contraseña suministrados por el administrador del sistema. Una vez ingresado con éxito, la aplicación redirige al usuario a la vista de personas.

Menú y Permisos

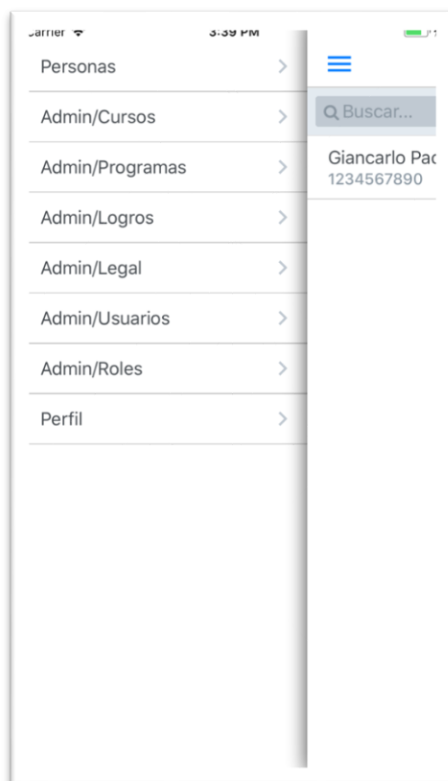


Figura 9 Menú del sistema

Las ocho vistas principales tienen el icono de menú en la esquina izquierda superior, al seleccionar este icono el menú de la aplicación se despliega en la parte izquierda de la pantalla. La siguiente tabla muestra las ocho opciones, una descripción de cada una y los permisos que debe tener el rol de usuario para acceder a cada una de ellas.

Menú	Descripción	Permisos
Personas	Despliega una lista de los beneficiarios de la fundación. Permite el acceso a cada uno de ellos y la visualización de su información.	Personas y Programas
Admin/Cursos	Despliega la lista de cursos disponibles. Se pueden activar o desactivar y modificar cada	Admin/Cursos

	uno de ellos. También se pueden acceder a las materias y modificarlas.	
Admin/Programas	Despliega la lista de programas disponibles en la fundación. Se pueden activar o desactivar, crear o modificar nuevos programas.	Admin/Programas
Admin/Logros	Despliega una lista de las etapas de logros disponibles. Se pueden activar o desactivar, agregar o modificar y agregar nuevos logros a cada etapa.	Admin/Logros
Admin/Legal	Despliega una lista de los documentos legales disponibles. Se pueden crear o modificar nuevos documentos.	Admin/Legal
Admin/Usuarios	Despliega una lista de los usuarios con acceso al sistema. Se pueden agregar usuarios, modificarlos y cambiar el rol al que pertenecen, activarlos o desactivarlos.	Admin/Usuarios
Admin/Roles	Despliega una lista de los roles disponibles del sistema. Se pueden agregar o modificar los roles, y cambiar los permisos de acceso de cada uno de ellos.	Admin/Roles
Perfil	Aquí se pueden realizar cambios de contraseña para el usuario actual, cerrar sesión en la	Perfil

	aplicación, o cambiar los datos del usuario.	
--	--	--

Personas

Esta es la sección principal de la aplicación. Aquí se pueden visualizar todos los beneficiarios y la información específica de cada uno de ellos.

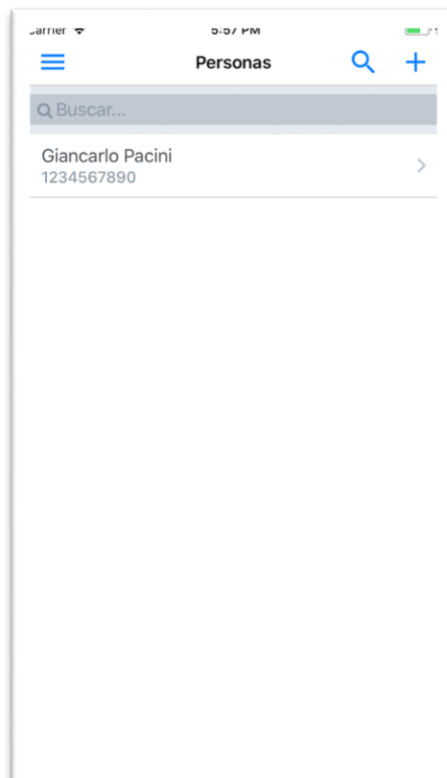


Figura 10 Visualización de Personas

La pantalla despliega la lista de beneficiarios de la fundación. En negro está el nombre del beneficiario y en gris el número de cedula del beneficiario. La barra de búsqueda en la parte superior de la lista permite filtrar los beneficiarios por nombre, apellido o número de cedula.

El botón de lupa en la esquina superior derecha despliega la pantalla de filtro por programa.

El botón “+” permite agregar un nuevo beneficiario.

Filtrar Por Programa

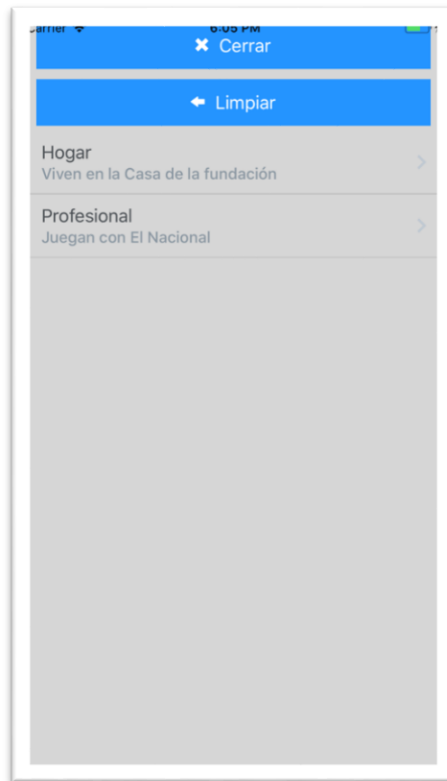
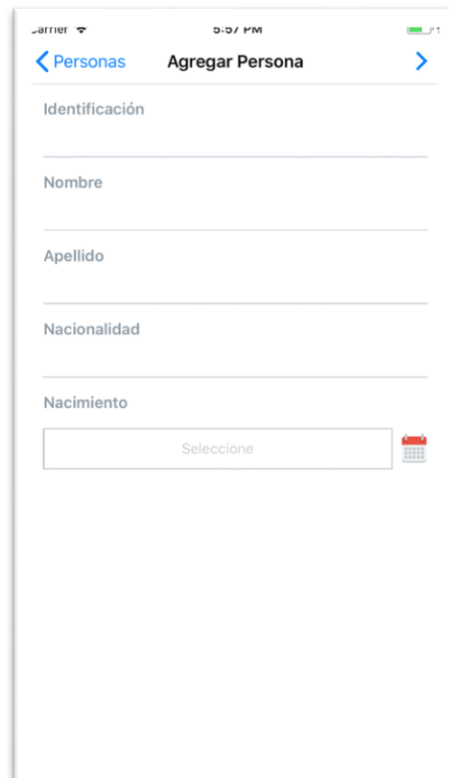


Figura 11 Pantalla para filtrar beneficiarios por Programa

Cuando se selecciona la lupa en la pantalla de visualización de personas, se despliega la pantalla de la figura 4. Aquí se puede seleccionar un programa para visualizar todos los beneficiarios que pertenecen a ese programa. Si es que un programa ya fue seleccionado, el botón “Limpiar” permite regresar a la lista completa de beneficiarios. El botón “Cerrar” cancela la acción y regresa a la lista de beneficiarios en su último estado.

Agregar o Editar Persona



The screenshot shows a mobile application interface for adding a person. The title bar at the top reads 'Agregar Persona' with a back arrow on the left and a forward arrow on the right. Below the title bar, there are several input fields: 'Identificación', 'Nombre', 'Apellido', 'Nacionalidad', and 'Nacimiento'. The 'Nacimiento' field is a date picker with the text 'Seleccione' and a calendar icon. The status bar at the top shows 'Carrier', '9:57 PM', and a battery icon.

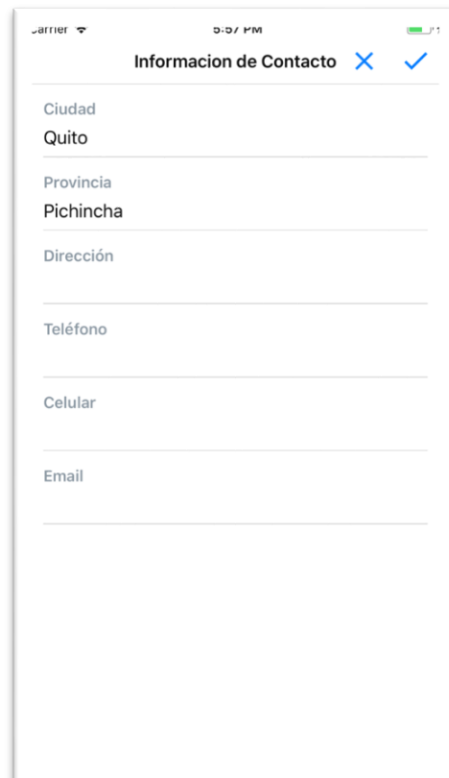
Figura 12 Agregar Persona

Esta pantalla permite agregar un nuevo beneficiario al sistema. Una vez llenados todos los campos es necesario presionar el botón “>” en la esquina superior derecha para continuar a la pantalla de información de contacto del beneficiario.

Flujo para la edición o creación de beneficiarios.

Agregar o editar persona -> Información de Contacto -> Información Familiar -> Historia Clínica -> Foto -> Visualización de Beneficiario

Información de Contacto



Carrier 9:57 PM 100%

Información de Contacto X ✓

Ciudad
Quito

Provincia
Pichincha

Dirección

Teléfono

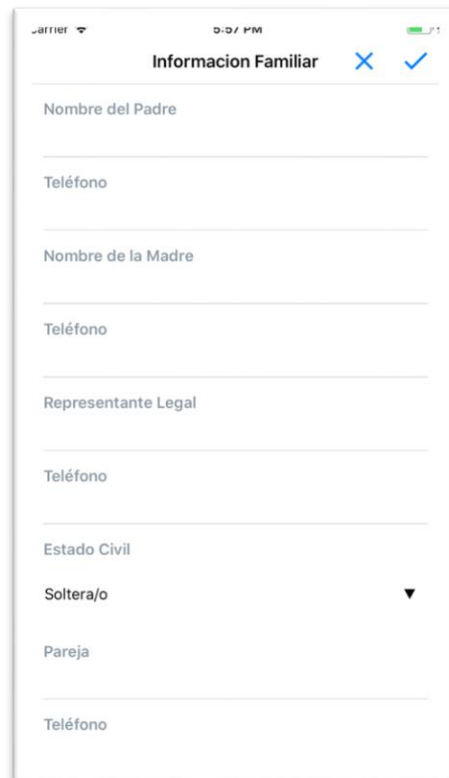
Celular

Email

Figura 13 Información de Contacto de Beneficiario

Una vez agregado un beneficiario al sistema, la aplicación redirige al usuario a la información de contacto. Aquí el usuario tiene la opción de llenar los campos para el beneficiario, esta información es opcional y puede ser llenada posteriormente. El botón “x” cancela la acción y redirige al usuario a la pantalla de información del beneficiario. El botón “check” en la esquina superior derecha guarda la información del contacto y redirige al usuario a la pantalla de información familiar del beneficiario.

Información Familiar



Carrier 9:57 PM 100%

Información Familiar X ✓

Nombre del Padre

Teléfono

Nombre de la Madre

Teléfono

Representante Legal

Teléfono

Estado Civil

Soltera/o ▼

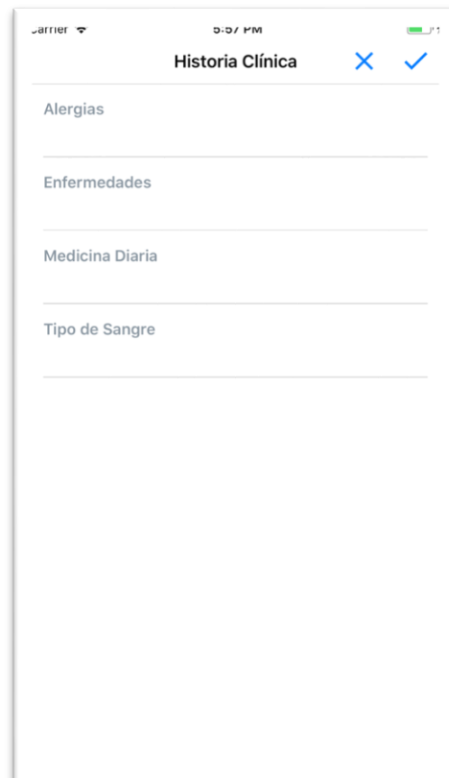
Pareja

Teléfono

Figura 14 Información Familiar de Beneficiario

Aquí el usuario puede modificar la información familiar del beneficiario. Todos los campos son opcionales y pueden ser llenados posteriormente. El botón “X” cancela la acción y redirige al usuario a la información del beneficiario. El botón “check” guarda la información y redirige al usuario a la información clínica del beneficiario.

Historia Clínica



The screenshot displays a mobile application interface for entering clinical history. The title bar at the top reads 'Historia Clínica' and includes a blue 'X' button for cancellation and a blue checkmark button for saving. Below the title, there are four text input fields, each preceded by a label: 'Alergias', 'Enfermedades', 'Medicina Diaria', and 'Tipo de Sangre'. The status bar at the very top shows 'Carrier', the time '9:57 PM', and a battery icon.

Figura 15 Historia Clínica del Beneficiario

Aquí el usuario puede ingresar la historia clínica del beneficiario. Estos datos son opcionales y pueden ser llenados posteriormente. El botón “X” cancela la acción y dirige al usuario a la información del beneficiario. El botón “check” guarda la información y dirige al usuario a la pantalla para subir foto del beneficiario.

Foto del Beneficiario

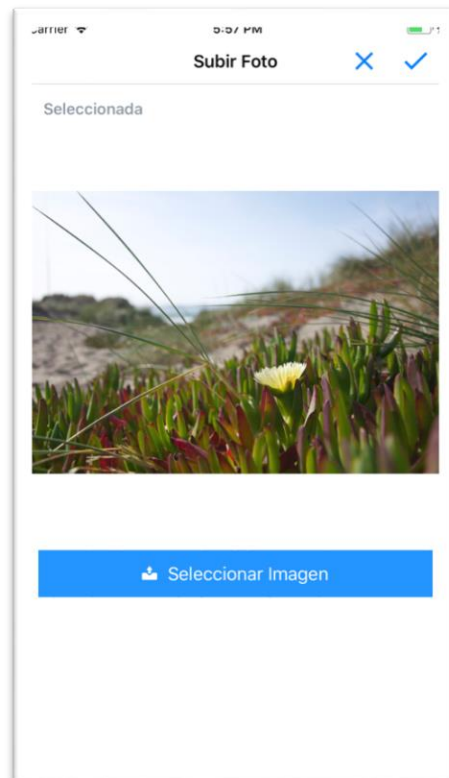


Figura 16 Pantalla Para Subir Foto de Un Beneficiario

Aquí se puede subir la foto de un beneficiario. El botón “Seleccionar Imagen” permite seleccionar una imagen de la librería de fotos del teléfono o tomar una nueva foto. Una vez seleccionada la imagen el botón “check” permite guardar la foto del beneficiario y redirige al usuario a la pantalla de información del beneficiario. El botón “X” cancela la acción y redirige al usuario a la pantalla de información del beneficiario.

Información de Beneficiario



Figura 17 Pantalla de Información de Beneficiario

Esta es la pantalla principal de la información de un beneficiario. Aquí se puede visualizar la foto del beneficiario, la información principal, botones que redirigen a diferentes secciones y la información de contacto, familiar y clínica.

En la parte superior derecha está el botón de editar beneficiario. Este botón redirige a la pantalla de creación de beneficiario con la información del beneficiario ya llena, esta sigue el mismo flujo de pantallas que sigue la creación del beneficiario.

Cuando se selecciona la foto del beneficiario, esta se expande para mostrarla en la pantalla completa.

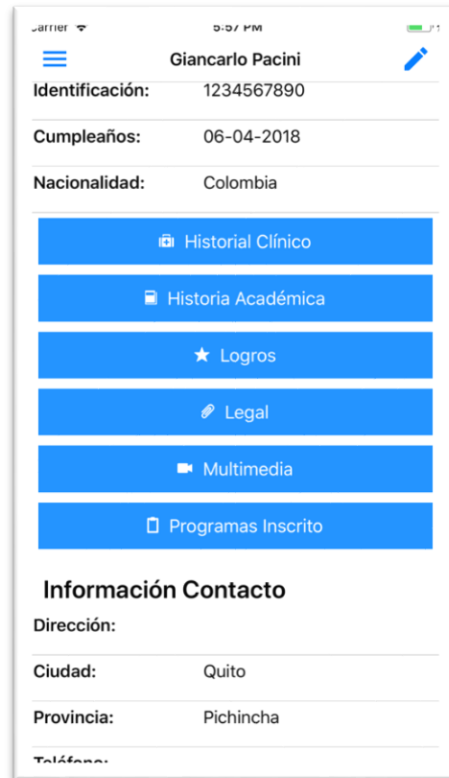


Figura 18 Botones de la información de beneficiario

Debajo de la información principal de un beneficiario se encuentran los botones. La siguiente lista muestra una descripción de cada botón y los permisos que se necesitan para acceder a cada uno de ellos.

Botón	Descripción	Permisos
Historial Clínico	Visualiza la evolución clínica del beneficiario, con gráficas y descripción de cada consulta.	Clínica
Historia Académica	Visualiza la historia académica del beneficiario. Cursos tomados, si aprobó o reprobó y las notas obtenidas.	Académico

Logros	Visualiza las etapas completadas y la etapa actual de logros del beneficiario.	Logros
Legal	Visualiza los documentos legales y da la opción de subir nuevos documentos para el beneficiario.	Legal
Multimedia	Visualiza todas las fotos del beneficiario y permite subir nuevas fotos.	Multimedia
Programas Inscrito	Visualiza los programas en los que está inscrito el beneficiario y permite inscribirlo en nuevos programas.	Programas

Carrier 5:07 PM 100%

☰ Giancarlo Pacini ✎

Información Contacto

Dirección:

Ciudad: Quito

Provincia: Pichincha

Teléfono:

Celular:

Correo:

Condiciones Clínicas

Alergias:

Enfermedades:

Medicina:

Tipo Sangre:

Información Familiar

Padre:

Teléfono:

Figura 19 Información de Beneficiario

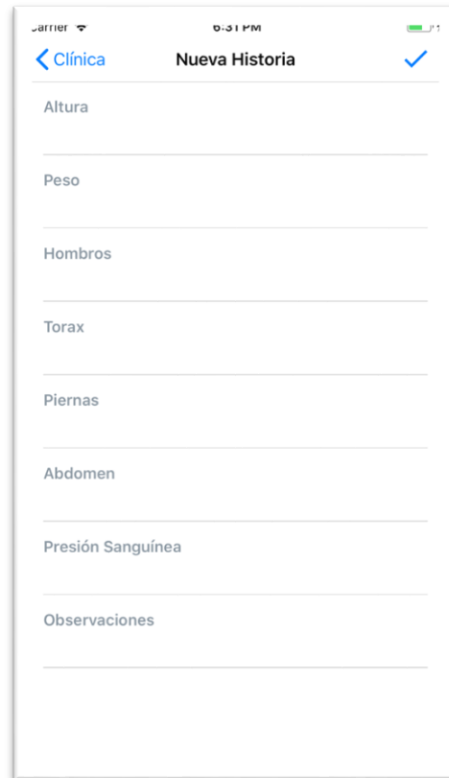
Debajo de los botones del beneficiario, se encuentra la información de contacto, información clínica e información familiar del beneficiario.

Historial Clínico



Figura 20 Pantalla Historial Clínico

Cuando el usuario selecciona el botón “Historial Clínico” en la pantalla de información del beneficiario, la aplicación lo redirige a esta pantalla. Aquí el usuario puede ver las curvas de la evolución de cada una de las medidas del beneficiario, puede visualizar dos curvas a la vez. En la parte inferior se despliega una lista de las consultas médicas, con sus fechas y las observaciones de la consulta. El botón “+” permite el ingreso de una nueva consulta, y seleccionar una consulta permite la edición de ella.



The image shows a mobile application interface for entering medical history. The screen is titled "Nueva Historia" and features a list of input fields for "Altura", "Peso", "Hombros", "Torax", "Piernas", "Abdomen", "Presión Sanguínea", and "Observaciones". A blue checkmark icon is visible in the top right corner.

Figura 21 Consulta Médica

En esta pantalla se pueden ingresar las medidas y las observaciones para la consulta actual. Si el usuario selecciono una consulta, los campos son llenados con la información de esa consulta. El botón "check" en la parte superior guarda la información de la consulta.

Historia Académica

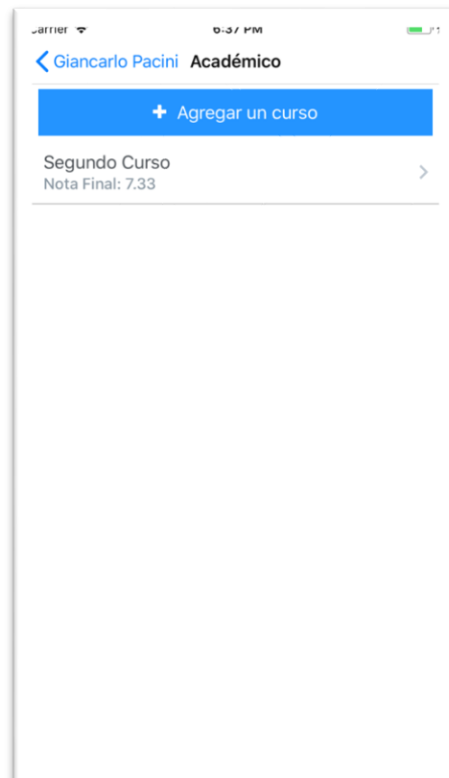


Figura 22 Historial Académico de Beneficiario

Aquí el usuario puede visualizar la historia académica del beneficiario. El botón “Agregar un Curso” despliega la lista de cursos disponibles en los que se puede inscribir el beneficiario. En la lista inferior se encuentran los cursos terminados o actuales del beneficiario y su nota final, si es que existe. Cuando se selecciona un curso, se puede ver la información del curso y las materias que tomó el beneficiario en ese curso.

Agregar Un Curso.

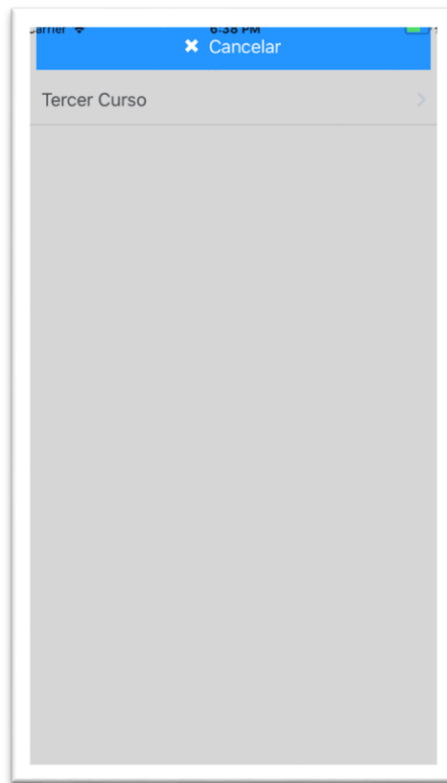


Figura 23 Pantalla para agregar un curso

Cuando se selecciona el botón de “Agregar Un Curso” en la historia académica de un beneficiario, se despliega la pantalla de la figura 16. Aquí el usuario puede seleccionar un curso disponible en el que se puede inscribir a un beneficiario. El botón “cancelar” cancela la acción y cierra esta pantalla.

Ver Curso.

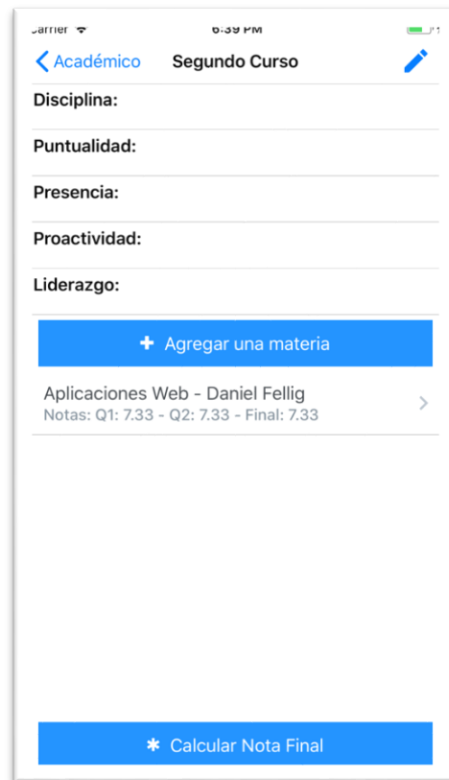
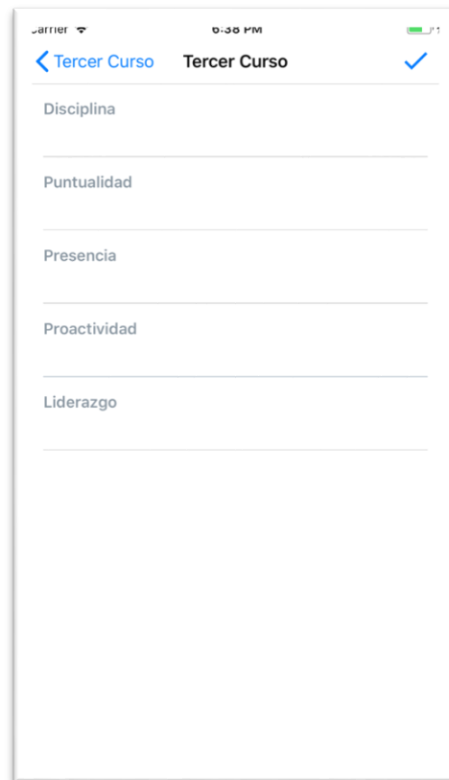


Figura 24 Visualización de Curso

Cuando se selecciona un curso de la lista de cursos del beneficiario se despliega esta pantalla. Aquí se pueden visualizar las notas generales del curso, agregar materias al curso, visualizar las materias actuales con sus notas, y calcular la nota final del curso.

El botón de lápiz en la esquina superior permite modificar las notas generales del curso. El botón "Agregar una materia" permite agregar una materia disponible para el beneficiario en el curso. Cuando se selecciona una materia la aplicación redirige a la visualización de la materia. El botón "Calcular Nota Final" calcula la nota final del curso en base a los promedios de cada materia.

Notas generales del curso.



The screenshot shows a mobile application interface for entering course notes. At the top, there is a status bar with the carrier name 'Carrier', the time '0:30 PM', and a battery icon. Below the status bar, there is a navigation bar with a back arrow, the text 'Tercer Curso', the title 'Tercer Curso', and a checkmark icon. The main content area contains five text input fields, each with a label above it: 'Disciplina', 'Puntualidad', 'Presencia', 'Proactividad', and 'Liderazgo'. Each input field is currently empty.

Figura 25 Notas Generales del Curso

Cuando se selecciona el botón de lápiz en un curso, la aplicación redirige al usuario a esta pantalla. Aquí el usuario puede ingresar las notas generales de un curso para el beneficiario. El botón “check” guarda la información y redirige al usuario a la visualización del curso.

Agregar Materia.

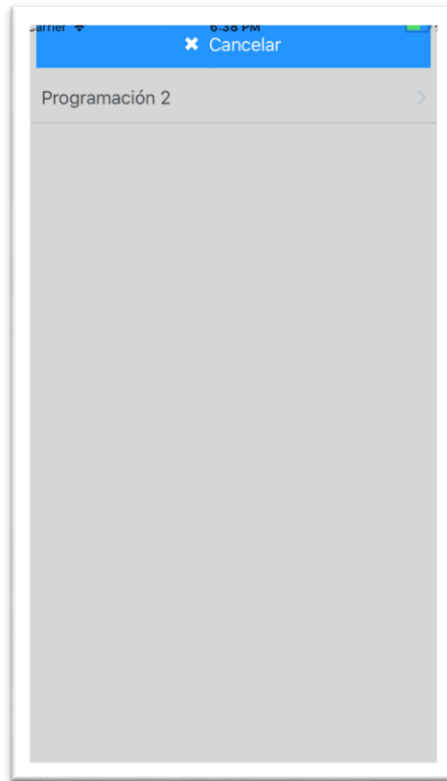


Figura 26 Agregar Materia a Curso

Si se selecciona el botón de “Agregar una materia” en la información de curso, se despliega esta pantalla con todas las materias disponibles para agregar a un curso. Si se selecciona una materia, esta es agregada. El botón de “Cancelar” cancela la acción y cierra la pantalla.

Ver Materia.



Figura 27 Visualización de Materia

Si se selecciona una materia de la lista de materias de un curso, se abre la pantalla de información de la materia. Aquí se puede ver la información de la materia y se pueden agregar las notas para cada uno de los bloques de los dos quimestres. La nota del quimestre es calculada con los promedios de los bloques, y la nota final es calculada con el promedio de los quimestres. El botón “check” de la esquina superior guarda la información de la materia.

Calcular Nota Final.

Una vez ingresadas todas las notas de las materias del curso, se puede presionar el botón de “Calcular Nota Final” en la pantalla principal del curso. Este botón calcula el promedio de las materias y redirige al usuario a la Historia Académica del beneficiario.

Logros

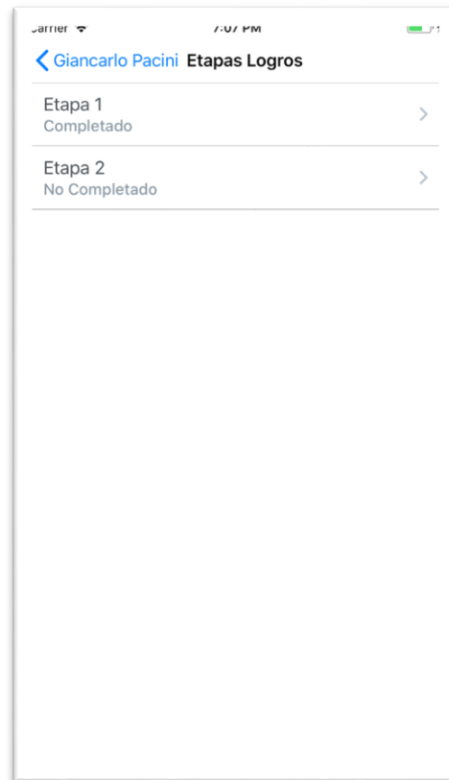


Figura 28 Visualización Logros del Beneficiario

Aquí el usuario puede visualizar las etapas de logros completadas y la etapa en curso del beneficiario. La primera vez que un usuario ingresa a esta pantalla el sistema automáticamente inicializa la etapa con menor orden. Cada vez que se cierra una etapa, el sistema inicializa la siguiente etapa de la lista. Cuando una etapa se selecciona, se pueden visualizar los logros completados y pendientes del beneficiario en esa etapa.

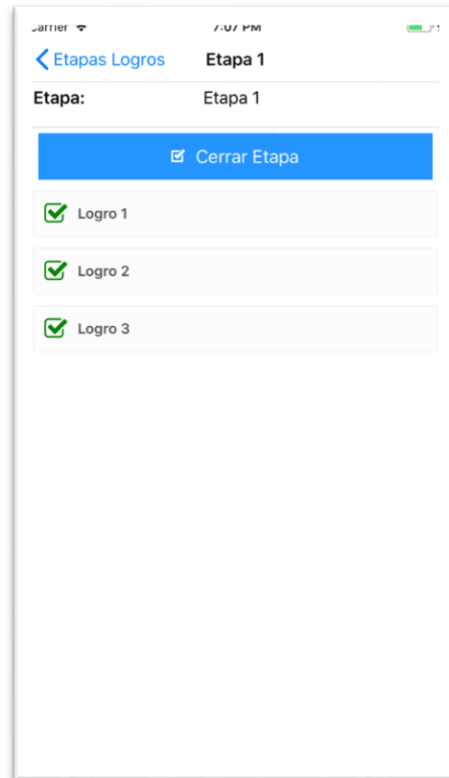


Figura 29 Visualización Etapa

Cuando una etapa es seleccionada de la lista de etapas de un beneficiario, la aplicación redirige al usuario a esta pantalla. Aquí se puede ver una lista de los logros completados y pendientes del beneficiario en esta etapa. Si un logro es seleccionado este se marca como completado o se desmarca dependiendo de su estado actual. Una vez completados todos los logros, se puede seleccionar el botón “Cerrar Etapa”, el sistema completa la etapa e inicializa una nueva etapa y redirige al usuario a la pantalla de etapas del beneficiario.

Legal

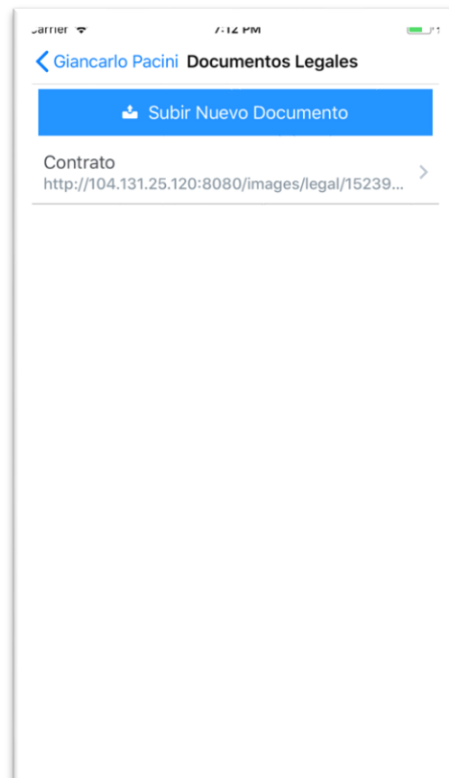


Figura 30 Visualización Documentos Legales de Beneficiario

Aquí se pueden visualizar los documentos legales subidos para un beneficiario. En la lista esta el tipo de documento y la dirección URL donde se encuentra. Si se selecciona el documento se abre la imagen del documento. Si se selecciona el botón “Subir Nuevo Documento” se despliega una lista con las opciones de documentos a subir.

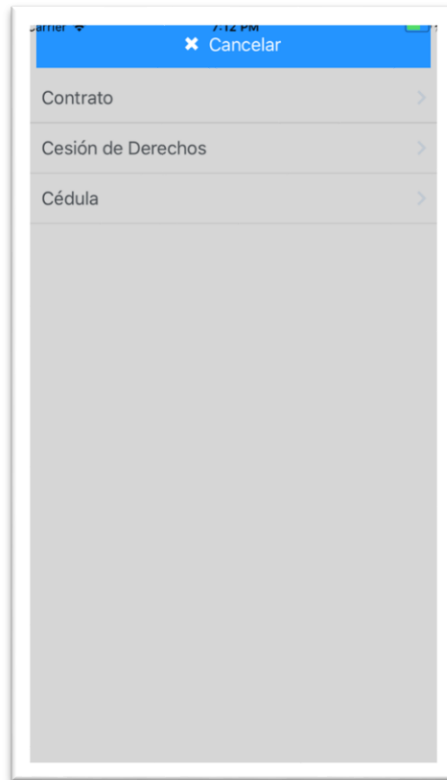


Figura 31 Seleccionar Documento

Cuando se selecciona el botón “Subir Nuevo Documento” de la lista de documentos, se despliega esta pantalla. Aquí el usuario puede seleccionar el documento a subir de un beneficiario. El botón “Cancelar” cierra la pantalla y regresa al listado de documentos.

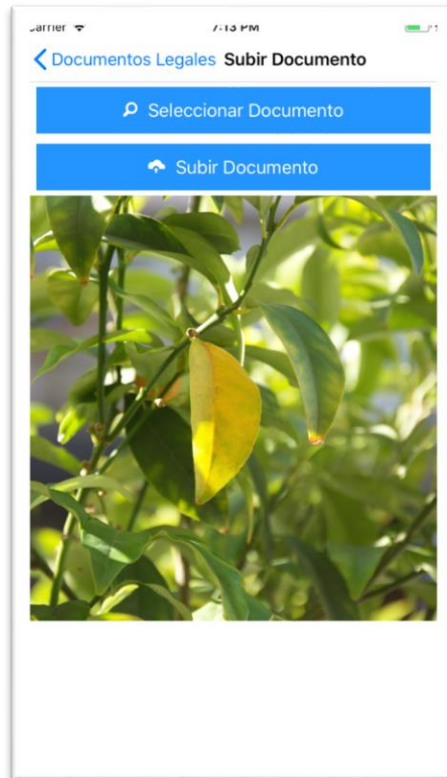


Figura 32 Subir Nuevo Documento

Cuando se selecciona un documento para subir, la aplicación despliega esta pantalla. Con el botón “Seleccionar Documento” el usuario puede seleccionar de su librería un documento para subir. Una vez seleccionado el documento el botón “Subir Documento” carga el documento al sistema y dirige al usuario a la pantalla de documentos.

Multimedia

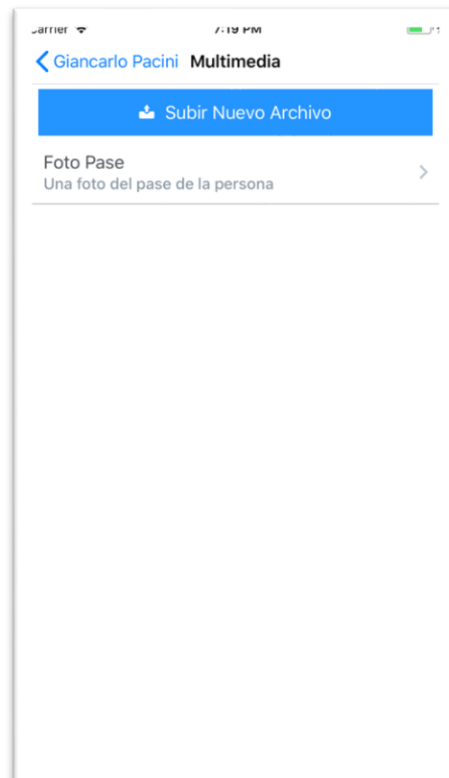


Figura 33 Visualización Multimedia

Aquí se pueden visualizar todas las imágenes subidas con respecto a un beneficiario. La lista despliega los documentos con su nombre y su descripción. Cuando se selecciona un documento se puede visualizar la imagen. El botón "Subir Nuevo Archivo" se puede subir una foto del beneficiario.

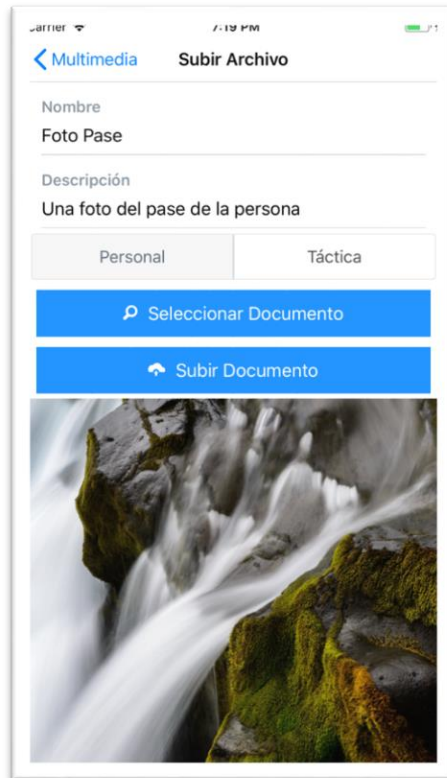


Figura 34 Subir Archivo Multimedia

Cuando se selecciona el botón “Subir Nuevo Archivo” de la pantalla de Visualización Multimedia se despliega esta pantalla. Aquí el usuario debe ingresar la información sobre la imagen que va a subir y si la imagen es personal o táctica. Una vez ingresada la información, con el botón “Seleccionar Documento” se puede seleccionar un archivo para subirlo. Con el archivo seleccionado, el botón “Subir Documento” guarda el archivo y redirige al usuario a la pantalla de Visualización Multimedia.

Programas Inscrito

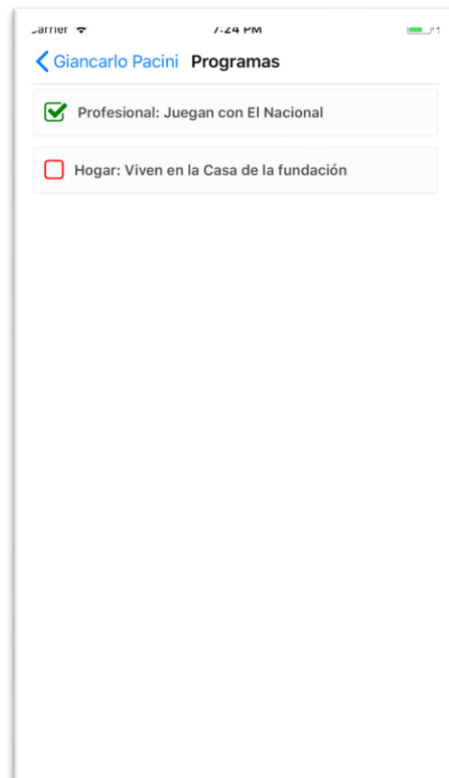


Figura 35 Visualización de Programas Inscritos del Beneficiario

Aquí se pueden visualizar los programas inscritos de un beneficiario. Se ve toda la lista de programas disponibles y se puede inscribir o des inscribir a un beneficiario de un programa. Los programas seleccionados en esta pantalla permiten filtrar al beneficiario posteriormente en la lista de beneficiarios.

Administración del Sistema

Administración de Roles.

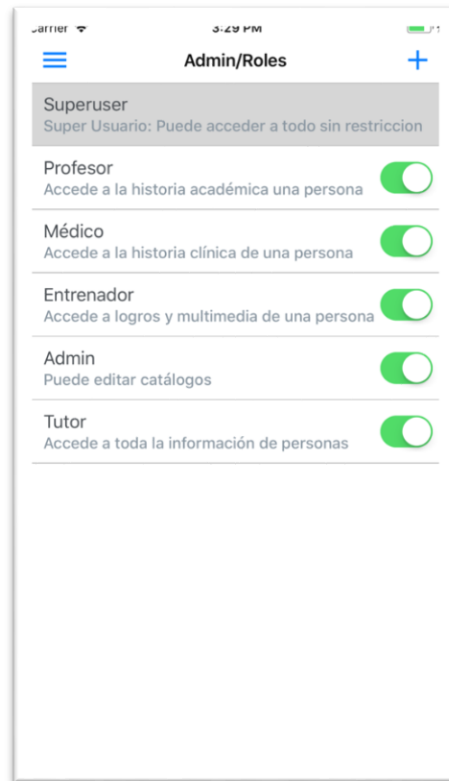


Figura 36 Administración de Roles

En esta pantalla se pueden administrar los roles del sistema y los accesos de cada uno de ellos. El rol Superuser no puede ser modificado, debido a que este rol es el administrador del sistema.

El botón a la derecha de cada uno de los roles permite activarlo o desactivarlo. Cuando un rol está desactivado los usuarios que pertenecen a ese rol no pueden acceder al sistema, tampoco se puede asignar el rol a nuevos usuarios.

El botón +, ubicado en la esquina superior derecha, permite el ingreso de nuevos roles. Seleccionar un rol de la lista permite la edición del rol.

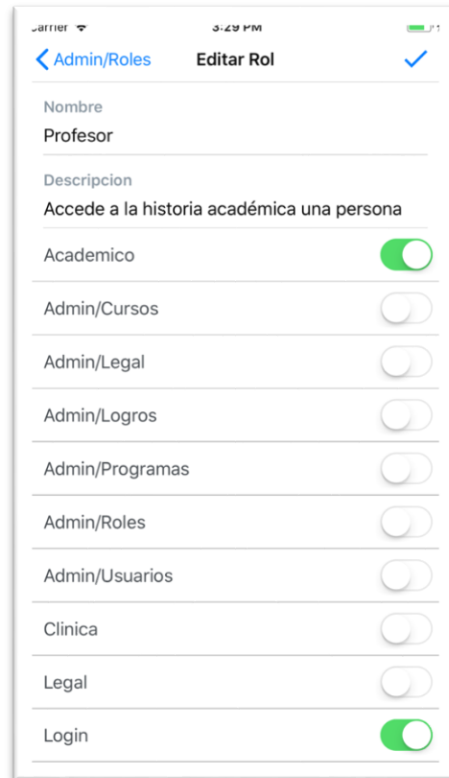


Figura 37 Pantalla de Edición de Roles

Cuando se agrega un nuevo rol, o cuando se selecciona uno para la edición, se despliega la pantalla de edición de roles. En la pantalla existen dos campos: Nombre, una cadena de texto para identificar el rol; y, Descripción, una breve descripción para saber que permisos tiene el rol. En la parte inferior de los campos de edición se despliega una lista de los permisos disponibles en el sistema, estos se pueden activar o desactivar a conveniencia. El visto en la esquina superior derecha guarda el rol.

Administración de Usuarios.

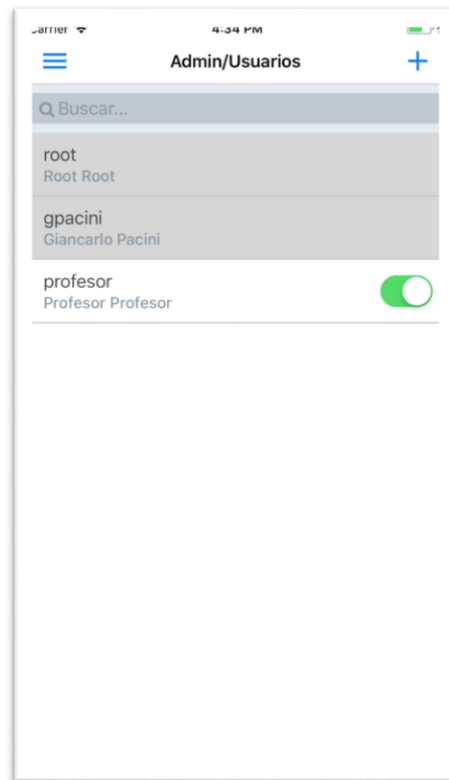
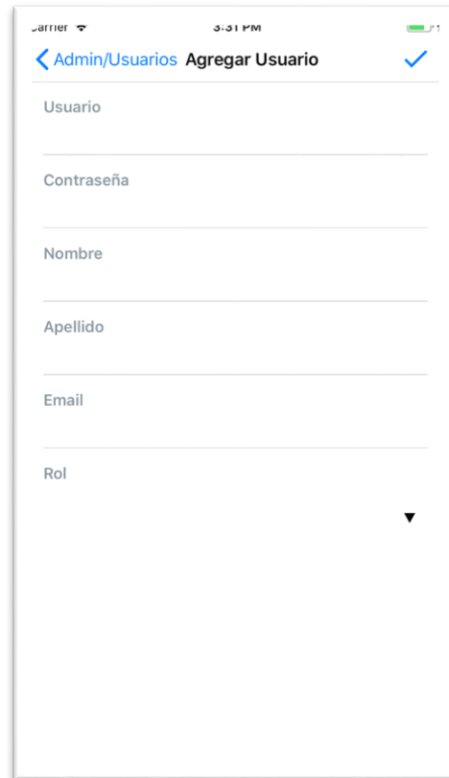


Figura 38 Administración de Usuarios

En esta pantalla se pueden visualizar los usuarios del sistema. Los usuarios en gris son súper usuarios y no pueden ser modificados. Para cada usuario se presenta el nombre de usuario en negro, y el nombre y apellido del usuario en gris. El botón de la derecha permite desactivar o activar el usuario, un usuario desactivado no tiene acceso al sistema.

El botón + de la esquina superior derecha permite el ingreso de un nuevo usuario al sistema.

Seleccionar un usuario permite la edición del usuario.



The image shows a mobile application interface for adding a user. At the top, there is a status bar with 'Carrier', '3:31 PM', and a battery icon. Below that, a navigation bar contains a back arrow, the text 'Admin/Usuarios', and 'Agregar Usuario' with a blue checkmark icon. The main content area consists of a form with the following fields: 'Usuario', 'Contraseña', 'Nombre', 'Apellido', 'Email', and 'Rol'. Each field has a horizontal line for input. The 'Rol' field has a small black downward-pointing triangle to its right, indicating a dropdown menu.

Figura 39 Pantalla de Creación de Usuarios

En esta pantalla se ingresan los datos para agregar un usuario. En la opción de rol se despliega una lista de los roles activados en el sistema. El campo usuario y el campo contraseña son los valores que usa un usuario para ingresar al sistema. El visto en la esquina superior derecha guarda el usuario.

Carrier 4:38 PM

< Admin/Usuarios Editar Usuario ✓

Reseteo Contraseña

Usuario
profesor

Nombre
Profesor

Apellido
Profesor

Email
profesor@serpahacer.com

Rol
Profesor ▼

Figura 40 Pantalla de Edición de Usuarios

En esta pantalla se puede editar un usuario. Debido a la seguridad del sistema no se puede cambiar la contraseña de un usuario, pero el botón de “Reseteo Contraseña” asigna un valor de “12345” a la contraseña de un usuario, para que este pueda acceder al sistema y cambiarla posteriormente en su perfil. Se puede cambiar el rol de un usuario seleccionándolo de la lista de roles que se despliega cuando se selecciona la opción de rol. El visto en la esquina superior derecha guarda el usuario.

Administración Legal.

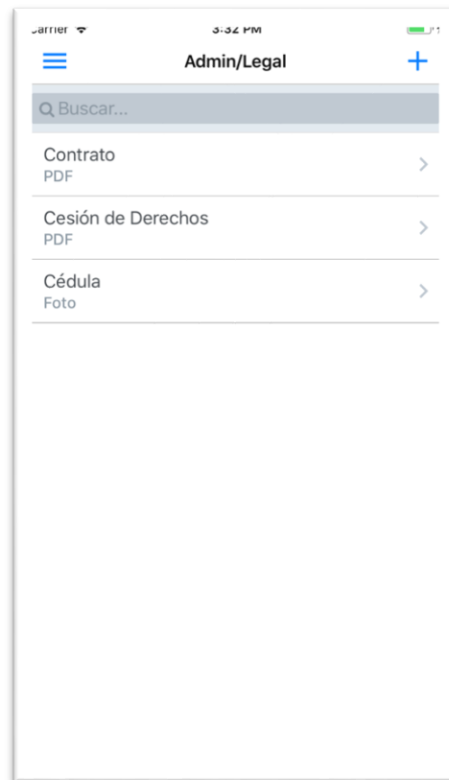
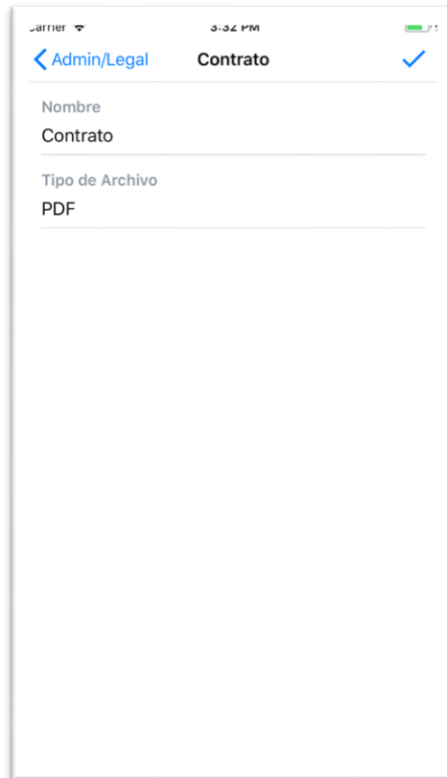


Figura 41 Pantalla de Administración Legal

Aquí se despliegan los documentos legales que pueden ser subidos para cada usuario. En negro se encuentra el nombre del documento, y en gris el formato recomendado para el documento. El botón “+” de la esquina superior derecha permite el ingreso de un nuevo documento. Seleccionar un documento permite la edición del mismo.



Carrier 3:54 PM

< Admin/Legal Contrato ✓

Nombre
Contrato

Tipo de Archivo
PDF

Figura 42 Pantalla de Edición y Creación de Documentos Legales

Aquí se pueden agregar el nombre y el tipo de archivo recomendados para el documento legal. El visto en la esquina superior derecha guarda el documento.

Administración de Logros.

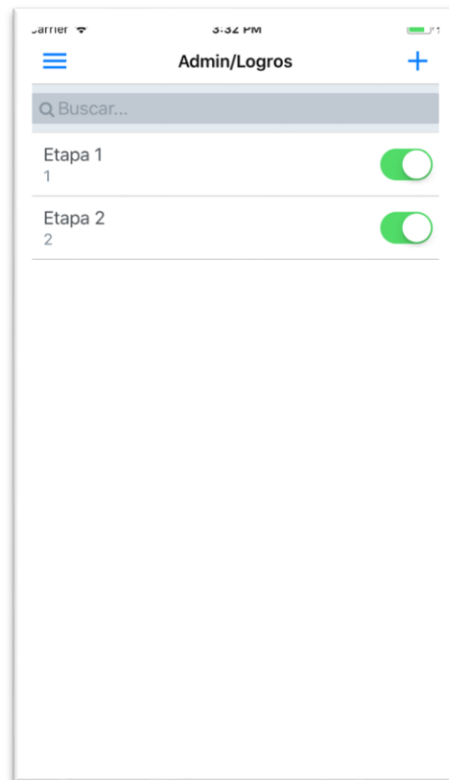
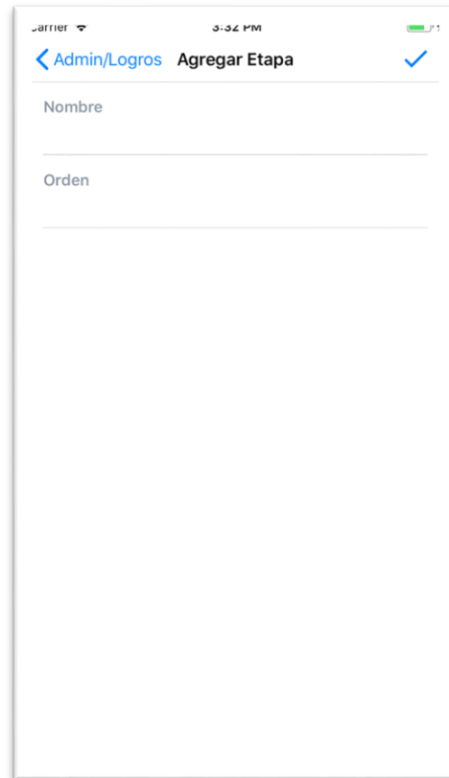


Figura 43 Pantalla de Administración de Logros

Aquí se pueden visualizar las etapas de logros del sistema. En negro se ve el nombre de la etapa, y en gris el orden de la etapa. Las etapas se activan para una persona en el orden en el que se encuentra, el orden no necesariamente tiene que ser números continuos. El botón de la derecha de cada etapa permite activarla o desactivarla, una etapa desactivada no se genera para un nuevo usuario. El botón “+” en la esquina superior derecha permite la creación de una nueva etapa. Seleccionar una etapa permite la visualización de la etapa y sus logros.



Carrier 3:52 PM

< Admin/Logros Agregar Etapa ✓

Nombre

Orden

Figura 44 Pantalla de Creación de Etapas

En esta pantalla se pueden crear etapas de logros. Se ingresan el nombre y el orden, el orden no tienen que ser continuos necesariamente. El visto en la esquina superior derecha guarda la etapa.

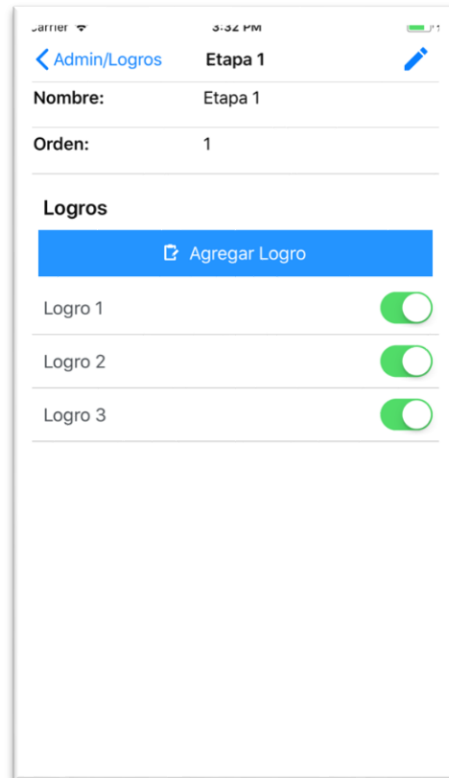


Figura 45 Pantalla de Visualización de Etapas

Quando una etapa es seleccionada en la administración de logros, la aplicación redirige al usuario a esta pantalla. Aquí se visualiza el nombre y el orden de la etapa. El botón de “Agregar Logro” permite agregar un nuevo logro a la etapa. Debajo de este botón se despliega una lista de los logros actuales de la etapa. Los logros se pueden activar o desactivar, un logro desactivado no se visualiza en los logros de un usuario y no es necesario para completar una etapa.

El botón de lápiz en la esquina superior derecha permite la edición de la etapa y redirige al usuario a la pantalla de creación de etapa, con los valores de esta etapa ingresados. Seleccionar un logro permite la edición de un logro.

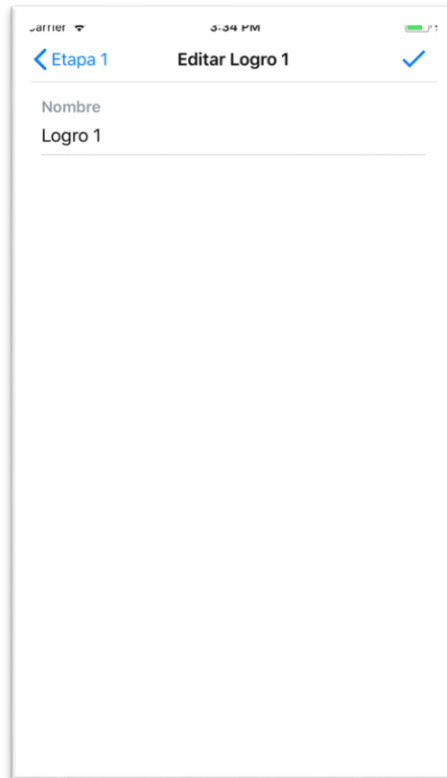


Figura 46 Pantalla de Creación y Edición de Logros

En esta pantalla se puede crear o editar un logro de una etapa. El único campo es el nombre del logro. El visto en la esquina superior derecha guarda el logro.

Administración de Programas.

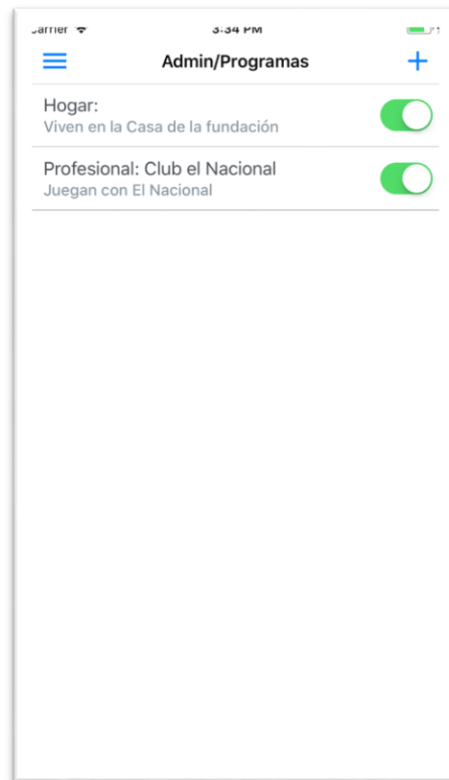
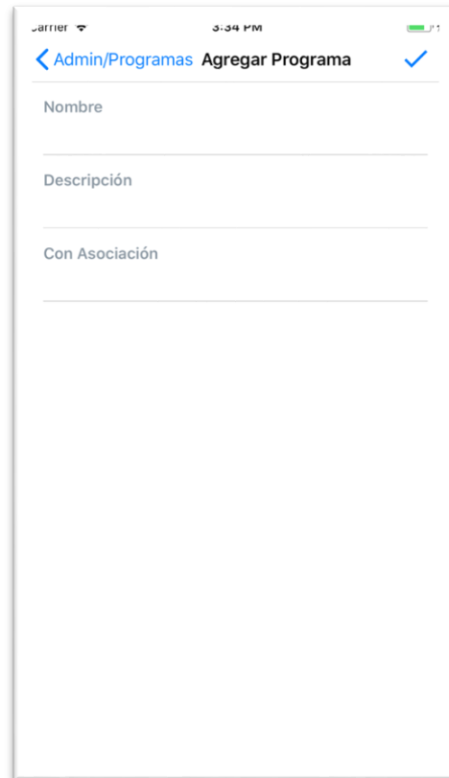


Figura 47 Administración de Programas

En esta pantalla se pueden visualizar los programas de la fundación. En negro se visualiza el nombre del programa y la entidad con la cual está asociado el programa, mientras que en gris se visualiza la descripción del programa. El botón a la derecha de cada programa permite activar o desactivar el programa, un programa desactivado no puede ser seleccionado para un jugador, pero si se puede usar como filtro. El botón “+” en la esquina superior derecha permite agregar un nuevo programa. Si se selecciona un programa este se puede editar.



The screenshot shows a mobile application interface for adding a program. At the top, the status bar displays 'Carrier', '3:34 PM', and a battery icon. Below the status bar, there is a navigation bar with a back arrow, the text 'Admin/Programas', and a checkmark icon. The main content area contains three text input fields with labels: 'Nombre', 'Descripción', and 'Con Asociación'. Each field is followed by a horizontal line indicating the input area.

Figura 48 Pantalla de Edición y Creación de Programas

En esta pantalla se pueden crear o editar los programas de la fundación. El visto en la esquina superior derecha guarda el programa.

Administración de Cursos.

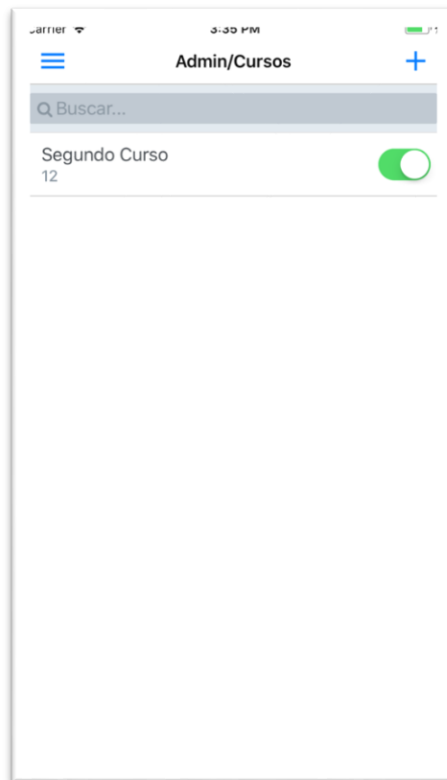
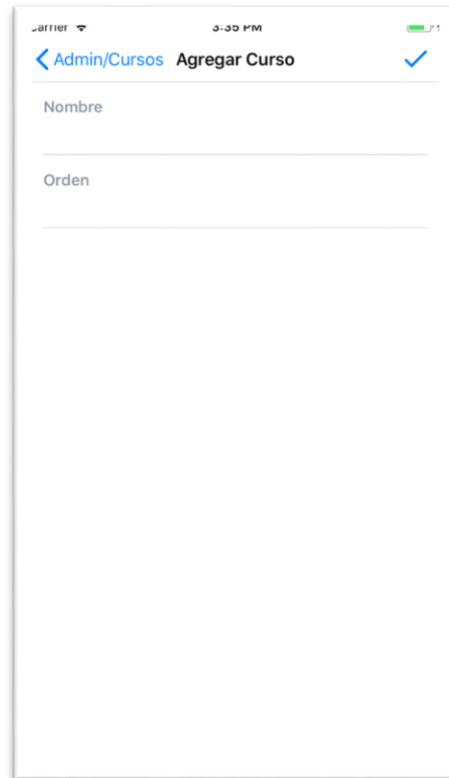


Figura 49 Administración de Cursos

En esta pantalla se despliega una lista de los cursos que dicta la fundación. En negro el nombre del curso y en gris el orden del curso, el orden no necesariamente tiene que ser de números continuos. El botón a la derecha de cada curso permite desactivar o activar un curso, un curso desactivado no puede ser asignado a un beneficiario. El botón “+” de la esquina superior derecha permite la creación de un nuevo curso. Seleccionar un curso permite la visualización del curso y las materias que se dictan en el curso.



The image shows a mobile application interface for adding a course. At the top, there is a status bar with the carrier name 'Carrier', the time '3:39 PM', and a battery icon. Below the status bar is a navigation bar with a blue back arrow, the text 'Admin/Cursos', and 'Agregar Curso' with a blue checkmark icon. The main content area contains two input fields: the first is labeled 'Nombre' and the second is labeled 'Orden'. Both fields are currently empty.

Figura 50 Pantalla de Creación de Cursos

En esta pantalla se puede agregar un nuevo curso. El nombre del curso es para identificarlo. El orden del curso es para ordenarlos. Los cursos están ordenados de mayor a menor, no necesariamente tienen que ser números continuos. El visto en la esquina superior derecha guarda el curso.

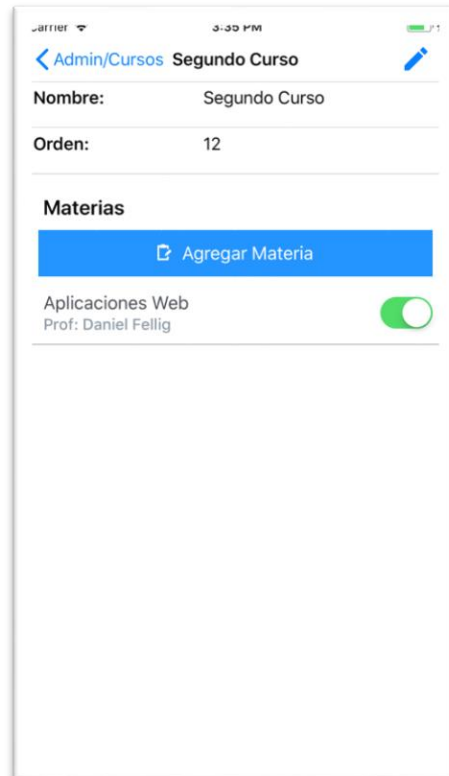
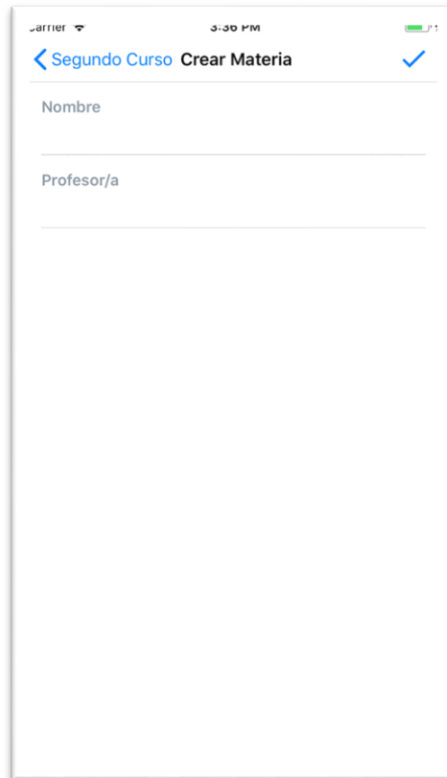


Figura 51 Pantalla de Visualización de Curso

En esta pantalla se visualiza la información de un curso. El botón de “Agregar Materia” permite agregar una materia relacionada al curso. El botón de lápiz en la esquina superior derecha permite editar un curso, esta opción redirige a la pantalla de creación de cursos, con la información del curso actual. Debajo del botón de “Agregar Materia” se enlistan todas las materias asignadas al curso, en negro se ve el nombre de la materia y en gris el profesor de la materia. El botón a la derecha de cada materia permite activarla o desactivarla, si una materia esta desactivada un beneficiario no puede ser inscrito en ella. Cuando se selecciona una materia redirige a la pantalla de edición de la materia.



The screenshot shows a mobile application interface for creating a subject. At the top, the status bar displays 'Carrier', '3:30 PM', and a battery icon. Below the status bar, there is a navigation bar with a back arrow, the text 'Segundo Curso', and 'Crear Materia', followed by a checkmark icon. The main content area contains two text input fields: the first is labeled 'Nombre' and the second is labeled 'Profesor/a'. Both fields are currently empty.

Figura 52 Pantalla de Creación y Edición de Materias

En esta pantalla se ingresan los datos de la materia que van a ser agregados al curso seleccionado previamente. El visto en la esquina superior derecha guarda la materia.

Perfil

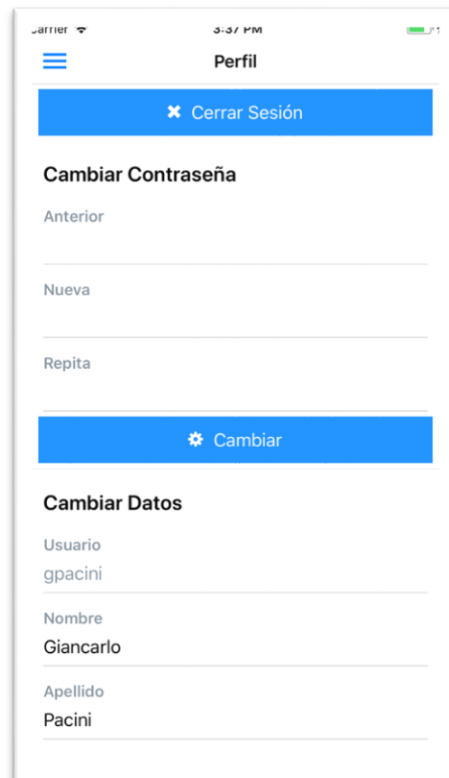


Figura 53 Vista superior de la pantalla de Perfil

En la parte superior de la pantalla de perfil está el botón de “Cerrar Sesión”, este botón cierra la sesión del usuario en el dispositivo móvil actual. La segunda sección permite cambiar la contraseña del usuario, para esto el usuario debe conocer su contraseña actual. Una vez ingresada la contraseña actual y la nueva contraseña el botón “Cambiar” cambia la contraseña del usuario.



Figura 54 Vista inferior de la pantalla de Perfil

La tercera sección de la pantalla de perfil permite cambiar los datos de un usuario. Aquí el usuario puede cambiar su nombre, apellido y su correo electrónico si desease. El nombre de usuario no puede ser cambiado por el usuario, solo por el administrador del sistema. El botón “Actualizar” cambia los datos del usuario.

ANEXO #6 MANUAL ADMINISTRATIVO

MANUAL ADMINISTRATIVO: APLICACIÓN “SER PA’ HACER”

Manual Administrativo: Aplicación “Ser Pa’ Hacer”	96
Requerimientos del Sistema	97
Requerimientos Mínimos de Hardware	97
Recomendaciones de Hardware	97
Requerimientos de Software	97
Recomendaciones de Software.....	97
Introducción	98
Preparación del Sistema	98
Instalación de Dependencias	99
MySQL.	99
Java.....	99
Wildfly	100
Configuración	101
Configuración de MySQL.....	101
Creación Carpeta Imágenes.	102
Configuración de Wildfly.....	102
Instalación EAR y Lanzamiento Aplicación.....	104

Requerimientos del Sistema

Requerimientos Mínimos de Hardware

- Procesador: 1 core.
- Memoria: 1 GB.
- Almacenamiento: 20 GB.

Recomendaciones de Hardware

- Procesador: 2 cores.
- Memoria: 4 GB.
- Almacenamiento: 50 GB.

Requerimientos de Software

- Sistema Operativo: Windows, Linux o MacOS.
- Motor Base de Datos: Cualquiera con driver Java.

Recomendaciones de Software

- Sistema Operativo: CentOS
- Motor de Base de Datos: MySQL

Introducción

El siguiente documento es una guía para la instalación del servidor “Ser Pa’ Hacer”. Esta guía está basada en las recomendaciones de software a usar, pero no necesariamente significa que se deban instalar las mismas dependencias. El administrador del sistema es libre de instalar el software que desee, mientras sea compatible con la aplicación EAR.

El manual está diseñado para instalar el software en un servidor CentOS, una base de datos MySQL, y un servidor de aplicaciones Wildfly.

Preparación del Sistema

El instalador de paquetes de CentOS se llama “yum”. Este permite instalar nuevos paquetes y actualizar los existentes. Para poder instalar las dependencias que se necesitan, es necesario tener actualizado yum y sus repositorios, para esto corremos las siguientes líneas en una consola del servidor, con permiso de super usuario.

```
[root@tempPrueba ~]# yum update
```

El sistema va a instalar varias actualizaciones y va a preguntar si desea continuar. Cuando aparezca lo siguiente, responde “y” y enter para continuar.

```
Is this ok [y/N]: █
```

Hay varios programas que vamos a usar en el transcurso de la aplicación, con el siguiente comando los instalamos todos.

```
[root@tempPrueba /]# yum install wget nano unzip
```

Instalación de Dependencias

MySQL.

A partir de CentOS 7, los desarrolladores de MySQL crearon una alternativa para los usuarios de CentOS. Esta alternativa es compatible con el sistema operativo, pero funciona igual que MySQL. Se llama MariaDB. Para instalar MariaDB ingresamos lo siguiente en la consola.

```
[root@tempPrueba ~]# yum install mariadb-server
```

Una vez que el sistema pregunte si deseamos continuar, presionamos “y” y enter.

Terminada la instalación de MariaDB hay que configurarla. En la consola ingresamos los siguientes comandos para iniciar el servicio y configurar el servidor MySQL.

```
[root@tempPrueba bin]# systemctl start mariadb  
[root@tempPrueba bin]# /bin/mysql_secure_installation
```

Seguimos los pasos del proceso de configuración y asignamos una contraseña. El resto de pasos dependen del acceso que se vaya a tener a base de datos, pero se recomienda poner “Y” en todas las opciones.

Java.

El servidor de aplicaciones Wildfly depende de Java para funcionar. En este caso vamos a instalar Java 8. Existe una variación de Java 8 con compatibilidad para CentOS, que se encuentra fácilmente en los repositorios de yum, esta es Java 8 OpenSDK.

Para instalar java corremos el siguiente comando en la consola de CentOS.

```
[root@tempPrueba bin]# yum install java-8-openSDK
```

Wildfly

Para instalar wildfly tenemos que haber instalado “wget” y “unzip” previamente. Si no están instalados, corriendo las siguientes líneas se pueden instalar.

```
[root@tempPrueba bin]# yum install wget
[root@tempPrueba ~]# yum install unzip
```

Una vez instaladas las dependencias obtenemos el archivo de Wildfly.

```
[root@tempPrueba bin]# wget http://download.jboss.org/wildfly/11.0.0.Final/wildfly-11.0.0.Final.zip
```

El sistema se va a descargar el archivo zip de wildfly, y a continuación corremos los siguientes comandos.

```
[root@tempPrueba ~]# mkdir /opt/wildfly && unzip wildfly-11.0.0.Final.zip &&
cp -rf wildfly-11.0.0.Final/* /opt/wildfly/ && rm -rf wildfly-11.0.0.Final && rm wildfly-11.0.0.Final.zip
```

Esos comandos descomprimen el archivo descargado, copia el contenido en “/opt/wildfly” y elimina la carpeta y archivos zip.

A continuación, abrimos el archivo de configuración de wildfly y ajustamos las variables del sistema para que funcione. Si es que el sistema dice que no existe nano, instalamos nano con el siguiente comando.

```
[root@tempPrueba /]# yum install nano
```

Luego abrimos el archivo de configuración.

```
[root@tempPrueba /]# nano /opt/wildfly/bin/standalone.conf
```

En un espacio libre escribimos la variable del sistema: `JBOSS_HOME="/opt/wildfly"`

Usamos “ctrl-o” para guardar y “ctrl-x” para salir.

Configuración

Una vez instalado todo lo que necesitamos, pasamos al paso de configuración para que nuestro servidor funcione. Primero, hay que configurar mysql para crear la base de datos. Luego, vamos a crear la carpeta donde se van a guardar las imágenes. Por último, vamos a configurar wildfly para que funcione con nuestra aplicación. Una vez realizados estos pasos, lanzamos nuestro EAR en el servidor de aplicaciones, y todo debería funcionar a la perfección.

Configuración de MySQL.

Primero, hay que crear el usuario y la base de datos que van a acceder al servidor. Para esto ingresamos a la configuración de MySQL con el siguiente comando e ingresamos la contraseña de administrador creada en el paso anterior.

```
[root@tempPrueba bin]# mysql -u root -p
```

Dentro del servidor MySQL ingresamos los siguientes comandos para crear la base de datos, el usuario con acceso y asignarle los privilegios al usuario.

```
[MariaDB [(none)]]> create database serpahacer;
```

El siguiente comando crea el usuario. Para esto hay que reemplazar {Nombre_Usuario} por el nombre de usuario que desee el administrador del sistema, hay que hacer lo mismo con {Contraseña}. No se puede olvidar de estos dos valores, debido a que se van a usar posteriormente.

```
MariaDB [(none)]]> create user '{Nombre_Usuario}'@'localhost' identified by '{Contraseña}';
```

Una vez creado el usuario, se le asignan los permisos necesarios para que acceda únicamente a la base de datos “serpahacer”.

```
MaríaDB [(none)]> GRANT ALL PRIVILEGES ON serpahacer.* TO '{Nombre_Usuario}'@'localhost';
```

La base de datos está lista para ser usada por la aplicación.

Creación Carpeta Imágenes.

La aplicación usa por defecto una carpeta del sistema para guardar las imágenes subidas a la aplicación. Para que funcione correctamente, hay que crear esta carpeta y darle los permisos necesarios.

Corremos el siguiente comando para crear la carpeta y asignarle los permisos necesarios.

```
[root@tempPrueba /]# mkdir -p /opt/serpahacer/fotos && chmod 775 /opt/serpahacer/fotos
```

Configuración de Wildfly.

La configuración de wildfly se da principalmente en el archivo standalone, que es el que lee el servidor cuando inicia el sistema. Para modificar este archivo, corremos el siguiente comando.

```
[root@tempPrueba /]# nano /opt/wildfly/standalone/configuration/standalone.xml
```

Dentro de este archivo vamos a modificar varias cosas. Primero vamos a buscar el siguiente subsistema dentro del archivo.

```
<subsystem xmlns="urn:jboss:domain:datasources:5.0">
  <datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="tr$
      <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
      <driver>h2</driver>
      <security>
        <user-name>sa</user-name>
        <password>sa</password>
      </security>
    </datasource>
  </datasources>
  <drivers>
    <driver name="h2" module="com.h2database.h2">
      <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
    </driver>
  </drivers>
</subsystem>
```

Debajo de `</datasource>` agregamos las siguientes líneas:

```
<datasource jndi-name="java:/SerPaHacerDS" pool-name="SerPaHacerDS" enabled="true">
  <connection-url>jdbc:mysql://localhost:3306/test_serpahacer</connection-url>
  <driver>mysql-driver</driver>
  <pool>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>30</max-pool-size>
  </pool>
  <security>
    <user-name>{Nombre_Usuario}</user-name>
    <password>{Password}</password>
  </security>
</datasource>
```

Donde `{Nombre_Usuario}` y `{Password}`, fueron los que le asignamos a la base de datos cuando la configuramos. Y, dentro de drivers, agregamos las siguientes líneas:

```
<driver name="mysql-driver" module="com.mysql.driver">
  <driver-class>com.mysql.jdbc.Driver</driver-class>
</driver>
```

Luego, buscamos el siguiente subsistema:

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  <buffer-cache name="default"/>
  <server name="default-server">
    <http-listener name="default" socket-binding="http" redirect-socket="https" enable-http2="true"/>
    <https-listener name="https" socket-binding="https" security-realm="ApplicationRealm" enable-http2="true"/>
    <host name="default-host" alias="localhost">
      <location name="/" handler="welcome-content"/>
      <filter-ref name="server-header"/>
      <filter-ref name="x-powered-by-header"/>
      <http-invoker security-realm="ApplicationRealm"/>
    </host>
  </server>
  <servlet-container name="default">
    <jsp-config/>
    <websockets/>
  </servlet-container>
  <handlers>
    <file name="welcome-content" path="{jboss.home.dir}/welcome-content"/>
  </handlers>
  <filters>
    <response-header name="server-header" header-name="Server" header-value="WildFly/11"/>
    <response-header name="x-powered-by-header" header-name="X-Powered-By" header-value="Undertow/1"/>
  </filters>
</subsystem>
```

Aquí, dentro de `<handlers>` agregamos la siguiente línea.

```
<file name="images" path="/opt/serpahacer/fotos" directory-listing="true"/>
```

Esta línea nos sirve para que el sistema lea las fotos guardadas en la carpeta que creamos previamente, y las sirva en la URL `/images`.

Por último, buscamos la siguiente sección:

```
<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address:127.0.0.1}"/>
  </interface>
</interfaces>
```

Y, dentro de <interface name="public">, cambiamos toda la línea por: <any-address/>.

Debería quedar así:

```
<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management:127.0.0.1}"/>
  </interface>
  <interface name="public">
    <any-address/>
  </interface>
</interfaces>
```

El último paso necesario para que funcione el servidor, es copiar el driver de MySQL necesario para la conexión con base de datos. Copiamos el archivo "mysql-connector-java-5.1.36.jar" adjunto, a la siguiente dirección: "/opt/wildfly/modules/system/layers/base/com/mysql/drive/main/mysql-connector-java-5.1.36.jar". Con esto, nuestro servidor de aplicaciones está listo para correr.

Instalación EAR y Lanzamiento Aplicación

Todo el ambiente y las dependencias necesarias para que nuestra aplicación funcione están instalados. Ahora necesitamos correr la aplicación. Para esto, vamos a copiar el archivo "serpahacer-ear.ear" a la siguiente dirección: "/opt/wildfly/standalone/deployments/serpahacer-ear.ear". Una vez copiado el archivo, corremos el siguiente comando y nuestra aplicación esta en producción.


```
[root@tempPrueba /]# /opt/wildfly/bin/standalone.sh
```