

# **Universidad de Los Andes**

Muu: Hagamos una vaca

## **Documento de Arquitectura del Sistema (SAD)**

**Nombre de los Integrantes:**

*Laura Ávila la.avila30@uniandes.edu.co*

*Sergio Hernández sd.hernandez204@uniandes.edu.co*

# Tabla de Contenido

## Contenido

### [Tabla de Contenido](#)

#### [Sección 1. Descripción del Documento](#)

- [1.1 Propósito y Audiencia](#)
- [1.2 Organización del Documento](#)
- [1.3 Terminología y Definiciones](#)
- [1.4 Documentos Relevantes](#)

#### [Sección 2. Generalidades del Proyecto](#)

- [2.1 Problema a Resolver](#)
- [2.2 Descripción General del Sistema a Desarrollar](#)
- [2.3 Objetivos](#)
- [2.4 Stakeholders](#)

#### [Sección 3. Motivadores Arquitecturales](#)

- [3.1 Motivadores de Negocio](#)
- [3.2 Restricciones de Tecnología](#)
- [3.2 Restricciones de Negocio](#)
- [3.3 Atributos de Calidad](#)
  - [3.3.1 Árbol de Utilidad](#)
  - [3.3.2 Escenarios de Calidad](#)

#### [Sección 4. Contexto](#)

- [4.1 Escenarios Operacionales](#)
  - [4.1.1 Escenario 1](#)
  - [4.1.2 Escenario 2](#)
  - [4.1.4 Escenario 3](#)
  - [4.1.5 Escenario 4](#)
- [4.2 Casos de Uso](#)

## Sección 5. Sensores y Análisis - Iteración 3

### 5.1 Sensor táctil (Pantalla)

### 5.2 Sensor de movimiento (Acelerómetro)

### 5.3 Sensor de Posición (GPS) - Análisis de Clustering usando K-Means

## Sección 6. Puntos de Vista y Modelos Arquitecturales

### 6.1 Contexto de la aplicación móvil

#### 6.2 Punto de Vista Funcional

##### 6.2.1 Modelo de Componentes

##### 6.2.2 Modelo de Desarrollo

##### 6.2.3 Modelo de Información

##### 6.2.4 Modelo de Despliegue

## Sección 7. Relaciones entre los Puntos de Vista

## Sección 8. Evaluación de Arquitectura

## Sección 1. Descripción del Documento

### 1.1 Propósito y Audiencia

Este documento muestra el diseño y la arquitectura de una aplicación que asiste la realización de las vacas o fondos de dinero, aquí se describen detalladamente las decisiones de diseño, de funciones y motivadores que tiene su desarrollo. Va dirigido a los interesados en conocer la arquitectura de la aplicación y su justificación, en particular, grupos de diseño y desarrollo de software que pueden tener interés en continuar esta iniciativa a partir de la documentación siguiente.

La aplicación que se describe en este documento parte del proyecto de la materia “*Construcción de Aplicaciones Móviles- ISIS3510*”, que se construirá de manera incremental en un lapso de 3 iteraciones, si bien cada una de ellas es una aplicación completamente funcional. Cada una de estas iteraciones están diferenciadas por los dispositivos para los que se diseña la aplicación, iniciando por dispositivos de Gama Baja, aquellos con conectividad nula; luego Gama Media, con conectividad y sensores limitados; y finalmente dispositivos Gama Alta. Si bien es una única aplicación, su arquitectura global se adapta a cada uno de los dispositivos y sus restricciones.

### 1.2 Organización del Documento

Inicialmente se describen las generalidades del proyecto y el problema que resuelve la aplicación, luego se habla sobre los motivadores arquitecturales que soportan el desarrollo de la aplicación en las diferentes iteraciones, a continuación se habla del contexto de la aplicación y finalmente de los puntos de vista y modelos arquitecturales propiamente dichos.

### 1.3 Terminología y Definiciones

- **Vaca:** Fondo de dinero que se reúne con un fin especificado, en este contexto entre personas conocidas. Recolección de dinero (una especie de crowdfunding a baja escala) entre amigos, con el fin común de comprar un servicio o pagar un producto.
- **Usuario Móvil/Administrador:** Aquel que hace uso de su dispositivo móvil, incluidas sus restricciones, y de los servicios de la aplicación para llevar el control de la vaca.
- **Participantes:** Todos aquellos que dan dinero para la vaca.
- **Registro histórico:** Registro de las vacas en las que ha participado un administrador o participante.

### 1.4 Documentos Relevantes

- Wiki del curso: <https://sistemas.uniandes.edu.co/~isis3510/dokuwiki/doku.php?id=inicio>
- Venmo - Can I use Venmo outside of the U.S.? - Consultado el 27 de Enero de 2016 <https://help.venmo.com/customer/portal/articles/1333177-can-i-use-venmo-outside-of-the-u-s->
- Venmo - Share Payments - Consultado el 27 de Enero de 2016 <https://venmo.com/>

## Sección 2. Generalidades del Proyecto

### 2.1 Problema a Resolver

Las vacas son fondos de dinero que se realizan tradicionalmente entre conocidos en diferentes lugares y eventos, pero tienen dificultades para su realización debido a que se hacen de forma informal y en situaciones donde es difícil llevar un control sobre las transacciones, por esto regularmente se olvida quién aportó, quien no y cuánto dinero puso cada persona, lo que resulta en que no sean siempre equitativas. Es común que las vacas se realicen entre las mismas personas en diferentes fechas por lo tanto el problema del control puede hacerse recurrente porque, adicionalmente, falta un historial entre las diferentes fechas. Esta falta de control tiene como consecuencias deudas recurrentes entre amigos que, algunas veces son difíciles de cobrar debido a la relación personal. Finalmente, estas vacas tienen como finalidad reunir más dinero del que puede tener una sola persona por lo tanto es trascendental la reunión de personas conocidas, lo cual se dificulta en eventos masivos.

Es por esto que nuestra aplicación surge con el fin de permitir un control puntual a cada vaca realizada y un control histórico a las vacas realizadas por el mismo conjunto de personas. Así mismo, es una forma de facilitar la reunión de conocidos y notificar entre ellos cuando hay una deuda.

### 2.2 Descripción General del Sistema a Desarrollar

Mu es una aplicación que se desarrolla de manera incremental y limitada por las restricciones de las gamas de dispositivos que se usarán en cada iteración. Permite al usuario (administrador) administrar la vaca mediante la división del costo total entre los participantes, registrar los montos que paga cada uno, sus vueltas y notificar si hay deudas entre los participantes. Así mismo, permite a los participantes buscar otros posibles participantes en recintos y eventos con el fin de que contribuyan en la vaca.

#### Iteración 1

Para la primera iteración MU brinda al administrador la posibilidad de gestionar localmente el registro histórico de sus vacas, ordenados según fecha y utilizando la agenda del dispositivo para facilitar el almacenamiento. Para cada una de las vacas permite adjuntar los participantes desde sus contactos, poner el valor que se pretende reunir, llevar las cuentas de los montos que pone cada participante y las vueltas según el mismo. Puede adjuntar fotos a cada registro de las vacas con el fin de facilitar el recuerdo de los gastos discriminados en las facturas. Finalmente como resultado del registro de una vaca se mandan mensajes SMS con el saldo que adeudan los participantes, en caso de que no se haya contribuido de manera equitativa en la vaca.

#### Iteración 2

Para la segunda iteración se hará uso de la conectividad para interoperar con las redes sociales de forma que se puedan enviar mensajes a través de las mismas. Así como tomar información de los eventos y sus integrantes con el fin de ubicar nuevos posibles participantes para la vaca. También, se hará uso de la geolocalización con el fin de encontrar otros participantes que estén en una ubicación cercana para que puedan encontrarse y hacer una vaca. El administrador puede notificar a los participantes por medio de la aplicación de sus correspondientes deudas.

#### Iteración 3

Para la última iteración, se hará uso de sensores. Se hará uso del sensor de ruido para que la aplicación pueda inducir que su propietario se encuentra en un evento masivo y así proponerle de empezar una vaca. También se hará uso del lector de huella digital como medio de firma como participante de la vaca.

### 2.3 Objetivos

La aplicación implementada sobre esta arquitectura, debe cumplir los siguientes objetivos:

- Aprovechar los recursos bien sean internos del dispositivo o externos con el fin de facilitar las interacciones del usuario.
- Facilitar la reunión de dinero con el fin de la creación de fondos entre conocidos.
- Ofrecer una experiencia al usuario intuitiva y simple para facilitar el uso de las funciones que brinda la aplicación.

#### Iteración 1: Dispositivos de Gama Baja

- Ofrecer un sistema sencillo que permita controlar la recolección de dinero para la creación de la vaca con una alta usabilidad.
- Persistir información que permita llevar un control histórico de las vacas anteriores.
- Aprovechar los recursos internos del celular para comunicar los resultados y adeudados de la vaca.

#### Iteración 2: Dispositivos de Gama Media

- Usar los canales de comunicación del dispositivo para interactuar con los participantes.
- Aprovechar la ubicación del usuario para permitir la recepción y envío de información importante.
- Distribuir la información de la vaca a los otros participantes.

#### Iteración 3: Dispositivos de Gama Alta

- Aprovechar los sensores propios del dispositivo para mejorar la experiencia del usuario y los atributos de calidad priorizados.

## 2.4 Stakeholders

A continuación se presentan los stakeholders del proyecto, su descripción y sus expectativas específicas.

Stakeholder	Descripción
Administrador (Usuario)	Usuario final. Cliente individual de la aplicación. Portador del dispositivo.
Participantes (Usuarios)	Usuario intermedio de la aplicación, con su propio dispositivo móvil. Puede tener la aplicación instalada o no.
Ingenieros de software	Desarrolladores a cargo de la construcción y mantenimiento de la aplicación descrita.
Administrador del negocio	Es el administrador general del negocio.

Tabla 1: Listado de Stakeholders

Stakeholder	Expectativas
Administrador (Usuario)	Facilidad en el control de la vaca y los montos aportados por cada participante. Así como la búsqueda de participantes y la externalización de las notificaciones a sus propios adeudados.
Participantes (Usuarios)	Control sobre el histórico de sus vacas, externalización de las notificaciones a sus propios adeudados. .
Ingenieros de software	Simplicidad en el desarrollo pero la completitud de los atributos de calidad.
Administrador del negocio	Espera sacarle ganancia a su negocio. Espera la correcta adaptación al mercado local.

Tabla 2: Stakeholders y Expectativas

## Sección 3. Motivadores Arquitecturales

### 3.1 Motivadores de Negocio

Nombre del Motivador de Negocio	Descripción del Motivador de Negocio	
Disminuir la inequidad en las vacas	Disminuir la inequidad en las vacas en un 80% mediante el registro de los montos aportados por cada participante	
Medida del Impacto		
Diferencia en el porcentaje de dinero que aporta cada participante		
Rangos	Cota Mínima	Cota Máxima
Ninguno	100	50
Bajo	50	20
Moderado	20	10
Fuerte	10	2
Muy Fuerte	2	0
Asociación del Motivador con el Negocio	Definido Por:	Ingenieros de Software
	Ejecutado Por:	Ingenieros de Software
	Ubicación en el Portafolio del negocio	

Nombre del Motivador de Negocio	Descripción del Motivador de Negocio	
Facilitar la transacción de la vaca	Facilitar la transacción de la vaca en un 80% mediante el ingreso de los montos que da cada participante y las vueltas necesarias para cada uno de ellos.	
Medida del Impacto		
Tiempo que tarda la transacción de la vaca		
Rangos	Cota Mínima	Cota Máxima
Ninguno	en adelante	30 min
Bajo	30 min	20 min
Moderado	20 min	15 min
Fuerte	15 min	10 min
Muy Fuerte	5 min	menos
Asociación del Motivador con el Negocio	Definido Por:	Ingenieros de Software
	Ejecutado Por:	Ingenieros de Software
	Ubicación en el Portafolio del negocio	

Nombre del Motivador de Negocio	Descripción del Motivador de Negocio
Aumentar el registro histórico de las vacas	Aumentar en un 90% el registro histórico de las vacas mediante el almacenamiento de la aplicación.
Medida del Impacto	
Número de vacas históricas registradas	

Rangos	Cota Mínima	Cota Máxima
Ninguno	0	1
Bajo	1	5
Moderado	5	10
Fuerte	10	20
Muy Fuerte	20	en adelante
Asociación del Motivador con el Negocio	Definido Por:	Ingenieros de Software
	Ejecutado Por:	Ingenieros de Software
	Ubicación en el Portafolio del negocio	

Nombre del Motivador de Negocio	Descripción del Motivador de Negocio	
Aumentar la cantidad de participantes que se encuentran en un evento masivo	Aumentar la cantidad de participantes que se encuentran en un evento masivo en un 70% mediante la comunicación de la ubicación de los mismos.	
Medida del Impacto		
Número de participantes que se encuentran en un evento masivo.		
Rangos	Cota Mínima	Cota Máxima
Ninguno	0	1
Bajo	1	2
Moderado	2	4
Fuerte	4	7
Muy Fuerte	7	En adelante
Asociación del Motivador con el Negocio	Definido Por:	Ingenieros de Software
	Ejecutado Por:	Ingenieros de Software
	Ubicación en el Portafolio del negocio	

Nombre del Motivador de Negocio	Descripción del Motivador de Negocio	
Disminuir las notificaciones personales de deuda	Disminuir las notificaciones personales de deuda en un 90% mediante las notificaciones automáticas de la aplicación.	
Medida del Impacto		
Número de notificaciones personales entre participantes.		
Rangos	Cota Mínima	Cota Máxima
Ninguno	en adelante.	5
Bajo	5	3
Moderado	3	2
Fuerte	2	1
Muy Fuerte	0	0
Asociación del Motivador con el Negocio	Definido Por:	Ingenieros de Software
	Ejecutado Por:	Ingenieros de Software
	Ubicación en el Portafolio del negocio	



### 3.2 Restricciones de Tecnología

ID Restricción: RT1	Tipo: Tecnología ( X ) Negocio ( )	Nombre: Dispositivos
Descripción:	La aplicación debe ejecutarse en dispositivos de distintas gamas.	
Establecida por:	Ingenieros de Software	
Alternativas:	Ninguna	
Observaciones:	Para llegar a una mayor cantidad de dispositivos y usuarios es necesario generar la aplicación en diferentes gamas de móviles.	

ID Restricción: RT2	Tipo: Tecnología ( X ) Negocio ( )	Nombre: Acceso al dispositivo
Descripción:	La aplicación debe ser capaz de responder cada funcionalidad en un tiempo razonable, que deberá ir disminuyendo a medida que aumenta la gama del dispositivo. (Ej: Gama baja: 4 segundos, gama media: 2 segundos, gama alta: 1 segundo)	
Establecida por:	Desarrolladores	
Alternativas:	Mejorar el diseño al máximo para “distraer” el usuario mientras se ejecuta una funcionalidad.	
Observaciones:	Se espera que los usuarios no sean muy pacientes.	

ID Restricción: RT3	Tipo: Tecnología ( X ) Negocio ( )	Nombre: Capacidad del Almacenamiento
Descripción:	La aplicación debe ser capaz de almacenar el registro histórico sin copar la capacidad del dispositivo. Deberá pesar menos del límite establecido para la gama del dispositivo.	
Establecida por:	Desarrolladores	
Alternativas:	Usar dispositivos de almacenamiento externos	
Observaciones:	Se debe notificar el uso de capacidad de almacenamiento	

### 3.2 Restricciones de Negocio

ID Restricción: RNe1	Tipo: Tecnología ( ) Negocio ( X )	Nombre: Mercado Colombiano - Dispositivos
Descripción:	La aplicación debe poderse instalar en los principales dispositivos usados en el mercado urbano colombiano. Debe poder funcionar en el 80% de los dispositivos.	
Establecida por:	Administrador del proyecto	
Alternativas:	Irse adaptando a la franja que más consume la aplicación.	
Observaciones:	N.A	

ID Restricción: RNe2	Tipo: Tecnología ( )	Nombre: Legislación: Manejo de Dinero
----------------------	-------------------------	---------------------------------------

	Negocio (X)	
<b>Descripción:</b>	La aplicación aún no puede manejar transacciones debido a la legislación del país. (En países como Estados Unidos es mucho más simple).	
<b>Establecida por:</b>	Administrador del proyecto	
<b>Alternativas:</b>	No manejar transacciones con dinero real sino manejar las cuentas y sumarle a estas cuentas el valor de los mensajes de texto.	
<b>Observaciones:</b>	N.A	

### 3.3 Atributos de Calidad

#### 3.3.1 Árbol de Utilidad

<b>Atributo de Calidad:</b>		Eficiencia		
<b>Tiempo</b>	<b>ID</b>	<b>Descripción</b>	<b>Prioridad</b>	
Tiempo de respuesta a peticiones	01	El sistema debe responder a las peticiones de los usuarios en menos de 1 segundo	Alta	
Tiempo de respuesta de ubicación	02	El sistema de ubicación debe mostrar la información con un retardo no mayor a 10 segundos	Media	
<b>Recursos</b>				
Almacenamiento del registro	03	La información debe persistir de manera eficiente sin superar un 15% de la capacidad de almacenamiento	Alta	
Envío de información	04	La información enviada debe ser lo más compacta posible y solo para los participantes elegidos.	Media	

<b>Atributo de Calidad:</b>		Fiabilidad		
<b>Tolerancia a Fallas</b>	<b>ID</b>	<b>Descripción</b>	<b>Prioridad</b>	
Fallas de ubicación	05	Errores de cálculo de ubicación deben ser menores a 1%	Media	
<b>Recuperabilidad</b>				
Fallas en el dispositivo o la aplicación	06	En caso de fallo de la aplicación o el dispositivo, al reiniciar debe recuperar el 100% de la aplicación	Media. Alta	
<b>Disponibilidad</b>				
Disponibilidad de la aplicación	07	La disponibilidad de la aplicación debe ser del 90%	Media	

<b>Atributo de Calidad:</b>		Mantenimiento		
<b>Modificación</b>	<b>ID</b>	<b>Descripción</b>	<b>Prioridad</b>	
Modificación del código	08	Al modificar un requerimiento menos del 15% de los métodos serán modificados.	Baja	

<b>Atributo de Calidad:</b>		Seguridad		
-----------------------------	--	-----------	--	--

Confidencialidad			
Confidencialidad en las redes sociales	09	La información y el acceso a las redes sociales se realizará con la aprobación del usuario	Alta
Confidencialidad de la ubicación	10	La información de la ubicación de los participantes debe ser confidencial a menos que ellos la quieran compartir.	Alta

Atributo de Calidad:		Facilidad de uso	
Interfaz	ID	Descripción	Prioridad
Sistema intuitivo	11	El usuario tiene una curva de aprendizaje corta	Alta

### 3.3.2 Escenarios de Calidad

Escenario de Calidad #	01	Stakeholder:	Administrador- Participante
Atributo de Calidad	Eficiencia tiempo		
Justificación	Al ser cálculos sencillos la aplicación no debe presentar congelamientos de la interfaz		
Fuente	Usuario		
Estímulo	Petición		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	La interfaz no se demora más de un segundo en responder		
Medida de la Respuesta	Tiempo de respuesta de la interfaz		

Escenario de Calidad #	02	Stakeholder	Administrador-Participante
Atributo de Calidad	Eficiencia tiempo		
Justificación	Al mostrar la ubicación el usuario no quiere esperar		
Fuente	Usuario		
Estímulo	Petición de ubicación de participantes		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	La interfaz no se demora más de diez segundos en responder		
Medida de la Respuesta	Tiempo de respuesta de la interfaz		

Escenario de Calidad #	03	Stakeholder:	Administrador-Participante
Atributo de Calidad	Eficiencia recursos		
Justificación	El usuario no quiere copar su capacidad con la información de la aplicación		
Fuente	Usuario		

Estímulo	Almacenamiento registro histórico
Artefacto	Dispositivo Móvil
Ambiente	Ejecución normal
Respuesta	La información se almacena sin superar el 15% de la capacidad
Medida de la Respuesta	% de la capacidad usado en almacenamiento

Escenario de Calidad #	04	Stakeholder:	Administrador
Atributo de Calidad	Eficiencia recursos		
Justificación	El administrador no quiere enviar información a todo el mundo ni gastarse la totalidad de su saldo		
Fuente	Usuario		
Estímulo	Enviar información participantes		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	La información es compacta y se envía solo a los elegidos		
Medida de la Respuesta	Tamaño de la información		

Escenario de Calidad #	05	Stakeholder:	Administrador-Participantes
Atributo de Calidad	Fiabilidad tolerancia a fallas		
Justificación	El usuario puede soportar algunas fallas en la ubicación pequeñas		
Fuente	Usuario		
Estímulo	Envío o recepción de ubicación participantes		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	Los errores de cálculo de ubicación son menores al 1%		
Medida de la Respuesta	Porcentaje de error		

Escenario de Calidad #	06	Stakeholder:	Administrador-Participantes
Atributo de Calidad	Fiabilidad recuperabilidad		
Justificación	El usuario quiere recuperar su información aun en caso de fallos		
Fuente	Usuario		
Estímulo	Reinicia el sistema		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución anormal		
Respuesta	En caso de fallas el sistema al reiniciar muestra el 100% de la información		

Medida de la Respuesta	% de información recuperada
------------------------	-----------------------------

Escenario de Calidad #	07	Stakeholder:	Administrador- Participantes
Atributo de Calidad	Fiabilidad disponibilidad		
Justificación	La aplicación debe estar disponible para cuando quiera usarla el usuario		
Fuente	Usuario		
Estímulo	Petición		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	La disponibilidad de la aplicación debe ser del 90%		
Medida de la Respuesta	% de tiempo que está disponible		

Escenario de Calidad #	08	Stakeholder:	Desarrollador
Atributo de Calidad	Mantenimiento		
Justificación	El desarrollador quiere modificar de manera sencilla y ágil		
Fuente	Desarrollador		
Estímulo	Modificación de un requerimiento		
Artefacto	Computador		
Ambiente	Ejecución normal		
Respuesta	Menos del 15 % de los métodos son modificados.		
Medida de la Respuesta	% métodos modificados		

Escenario de Calidad #	09	Stakeholder:	Administrador- Participante
Atributo de Calidad	Seguridad Confidencialidad		
Justificación	Los usuarios quieren tener información personal de manera privada		
Fuente	Usuario		
Estímulo	Extraer información de las redes sociales.		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	El usuario debe dar su aprobación de acceso a las redes sociales.		
Medida de la Respuesta	No se consulta ninguna red social sin el consentimiento del usuario		

Escenario de Calidad #	10	Stakeholder	Participante
Atributo de Calidad	Seguridad confidencialidad		

Justificación	No todos los participantes quieren mostrar su ubicación en las diferentes situaciones
Fuente	Usuario
Estímulo	Petición ubicación
Artefacto	Dispositivo Móvil
Ambiente	Ejecución normal
Respuesta	No se muestra la ubicación a menos que el participante lo autorice
Medida de la Respuesta	No hay ningún participante que muestre su ubicación sin autorización

Escenario de Calidad #	11	Stakeholder:	Administrador- Participante
Atributo de Calidad	Facilidad de uso en la interfaz		
Justificación	El funcionamiento es sencillo y por lo tanto el usuario no debe tardar aprendiéndolo		
Fuente	Usuario		
Estímulo	Inicio del uso de la aplicación		
Artefacto	Dispositivo Móvil		
Ambiente	Ejecución normal		
Respuesta	El usuario tiene una curva de aprendizaje corta		
Medida de la Respuesta	Tiempo requerido para aprender a usar la aplicación		

## Sección 4. Contexto

### 4.1 Escenarios Operacionales

#### Iteración 1

##### 4.1.1 Escenario 1

Título del Escenario Operacional			
Escenario 1			
Stakeholder Asociado	Usuario Final	ID	Eo-01
Consideración Operacional	Respuesta del Stakeholder		
Descripción general de la funcionalidad	El usuario ingresa a la aplicación. Realiza la vaca anotando quienes le aportan y cuánto aportan. La aplicación realiza los cálculos necesarios para redistribuir vueltas, si estas existen. La aplicación realiza el registro una vez hecho. La aplicación le envía un mensaje SMS a los deudores luego de una cierta cantidad de tiempo predefinida por el usuario.		
Describe lo que el Stakeholder hace ahora o le gustaría poder hacer	El Stakeholder puede realizar la funcionalidad primaria de la aplicación. Realiza el registro de los aportadores, anotando cuánto van aportando. Luego registra la compra, y le entrega el dinero de cambio a las personas correspondientes. La aplicación luego envía SMS a los deudores luego de cierto tiempo. El Stakeholder podrá revisar el historial cuando lo desee.		

Describe cualquier entrada provista o disponible al momento del inicio	El usuario debe ingresar cada participante de la vaca con su aporte respectivo. Una vez se complete dicha recolección, debe registrar la compra.
Describe el contexto de la operación	Un usuario que se encuentre en una fiesta (o cena) y que desee recolectar dinero entre sus amigos que se encuentren en el evento para poder comprar algún producto (ej: plato, botella de licor) costoso que sólo sea accesible si todos se cotizan.
Describe cómo el sistema debe responder	El sistema debe registrar la información, realizar los cálculos para el cambio. Debe guardar el historial. La información debe ser persistida de manera sencilla y eficiente. El sistema debe enviar un SMS periódico a cada contacto que deba dinero, indicando la suma que debe y a quien debe pagarse mientras este no haya pagado y su pago no haya sido registrado en el sistema.
Describe las salidas que el sistema produce como resultado de la acción	El sistema tendrá almacenado el registro de la recolección, habrá hecho los cálculos para el cambio. Luego, de cierta periodicidad definida por el usuario, enviará SMS a cada contacto que deba dinero.
Describe quién o qué usa la salida y para que es utilizada	Los contactos que reciben los SMS recordarán su deuda. La probabilidad de que contacten a la persona a la que le deben dinero y le paguen su deuda aumentarán.

## Iteración 2

### 4.1.2 Escenario 2

Título del Escenario Operacional			
Escenario 2			
Stakeholder Asociado	Usuario Final	ID	Eo-02
Consideración Operacional	Respuesta del Stakeholder		
Descripción general de la funcionalidad	El usuario ingresa a la aplicación. Realiza la vaca anotando quienes le aportan y cuánto aportan. La aplicación realiza los cálculos necesarios para redistribuir vueltas, si estas existen. La aplicación realiza el registro una vez hecho. La aplicación interopera con las cuentas de redes sociales y aplicaciones de comunicación (ej: Facebook Messenger, Facebook, Whatsapp, etc..) para notificar a los miembros de la recolección el estado de sus cuentas. Adicionalmente, el sistema aprovechará la información de eventos en dichas redes sociales para proponerle a amigos cercanos del usuario de acceder a una vaca.		
Describe lo que el Stakeholder hace ahora o le gustaría poder hacer	El Stakeholder puede realizar la funcionalidad primaria de la aplicación. El sistema le propone un primer grupo para empezar la recolección si el evento donde se encuentra el usuario está registrado en una red social. Realiza el registro de los aportadores, anotando cuánto van aportando. Luego registra la compra, y le entrega el dinero de cambio a las personas correspondientes. El sistema se conecta a las cuentas de redes sociales y aplicaciones de comunicación (ej: Facebook Messenger, Facebook, Whatsapp, etc..) para notificar a los miembros de la recolección el estado de sus cuenta. El Stakeholder podrá revisar el historial cuando lo desee.		
Describe cualquier entrada provista o disponible al momento del inicio	El usuario le permite a la aplicación extraer información de su cuenta de red social. El usuario debe ingresar cada participante de la vaca con su aporte respectivo. Una vez se complete dicha recolección, debe registrar la compra.		
Describe el contexto de la operación	Un usuario que se encuentre en una fiesta (o cena) y que desee recolectar dinero entre sus amigos que se encuentren en el evento para poder comprar algún producto (ej: plato, botella de licor) costoso que sólo sea accesible si todos se cotizan.		
Describe cómo el sistema debe responder	El sistema debe registrar la información, realizar los cálculos para el cambio. Debe guardar el historial. La información debe ser persistida de manera sencilla y eficiente. El sistema debe enviar mensajes a través de la red social apropiada a cada contacto que deba dinero, indicando la suma que debe y a quien debe pagarse mientras este no haya pagado y su pago no haya sido registrado en el sistema. Si el usuario no tiene cuenta en red social, se debe realizar la comunicación por SMS.		

Describe las salidas que el sistema produce como resultado de la acción	El sistema tendrá almacenado el registro de la recolección, habrá hecho los cálculos para el cambio. Luego, de cierta periodicidad definida por el usuario, enviará una notificación a través de red social a cada contacto que deba dinero. Si dicho contacto no tiene cuenta en la red social, se enviará un SMS con dicha información.
Describe quién o qué usa la salida y para que es utilizada	Los contactos que reciben los mensajes/notificaciones (o SMS) recordarán su deuda. La probabilidad de que contacten a la persona a la que le deben dinero y le paguen su deuda aumentarán.

#### 4.1.4 Escenario 3

Título del Escenario Operacional			
Escenario 3			
<b>Stakeholder Asociado</b>	Usuario Final	<b>ID</b>	Eo-003
<b>Consideración Operacional</b>	<b>Respuesta del Stakeholder</b>		
Descripción general de la funcionalidad	El usuario ingresa a la aplicación. El sistema aprovechará la geolocalización de los aparatos, además de la información de eventos en redes sociales para proponerle a amigos cercanos del usuario de realizar una <i>vaca</i> . El usuario va anotando quienes le aportan y cuánto aportan. La aplicación realiza los cálculos necesarios para redistribuir vueltas, si estas existen. La aplicación realiza el registro una vez hecho. La aplicación interopera con las cuentas de redes sociales y aplicaciones de comunicación (ej: Facebook Messenger, Facebook, Whatsapp, etc..) para notificar a los miembros de la recolección el estado de sus cuentas.		
Describe lo que el Stakeholder hace ahora o le gustaría poder hacer	El Stakeholder puede realizar la funcionalidad primaria de la aplicación. El sistema le propone un primer grupo cercano a él para empezar la recolección (o unirse a una en curso). Realiza el registro de los aportadores, anotando cuánto van aportando. Luego registra la compra, y le entrega el dinero de cambio a las personas correspondientes. El sistema se conecta a las cuentas de redes sociales y aplicaciones de comunicación (ej: Facebook Messenger, Facebook, Whatsapp, etc..) para notificar a los miembros de la recolección el estado de sus cuenta. El Stakeholder podrá revisar el historial cuando lo desee.		
Describe cualquier entrada provista o disponible al momento del inicio	El usuario le permite a la aplicación extraer información de su cuenta de red social. El usuario debe ingresar cada participante de la vaca con su aporte respectivo. Una vez se complete dicha recolección, debe registrar la compra.		
Describe el contexto de la operación	Un usuario que se encuentre en una fiesta (o cena) y que desee recolectar dinero entre sus amigos que se encuentren en el evento para poder comprar algún producto (ej: plato, botella de licor) costoso que sólo sea accesible si todos se cotizan.		
Describe cómo el sistema debe responder	El sistema debe registrar la información, realizar los cálculos para el cambio. El sistema debe ser capaz de detectar los dispositivos en una localización cercana. Debe guardar el historial. La información debe ser persistida de manera sencilla y eficiente. El sistema debe enviar mensajes a través de la red social apropiada a cada contacto que deba dinero, indicando la suma que debe y a quien debe pagarse mientras este no haya pagado y su pago no haya sido registrado en el sistema. Si el usuario no tiene cuenta en red social, se debe realizar la comunicación por SMS.		
Describe las salidas que el sistema produce como resultado de la acción	El sistema tendrá almacenado el registro de la recolección, habrá hecho los cálculos para el cambio. Luego, de cierta periodicidad definida por el usuario, enviará una notificación a través de red social a cada contacto que deba dinero. Si dicho contacto no tiene cuenta en la red social, se enviará un SMS con dicha información.		
Describe quién o qué usa la salida y para que es utilizada	Los contactos que reciben los mensajes/notificaciones (o SMS) recordarán su deuda. La probabilidad de que contacten a la persona a la que le deben dinero y le paguen su deuda aumentarán.		

#### Iteración 3

#### 4.1.5 Escenario 4

Título del Escenario Operacional
Escenario 4



Stakeholder Asociado	Usuario Final	ID	Eo-004
Consideración Operacional	Respuesta del Stakeholder		
Descripción general de la funcionalidad	El usuario ingresa a la aplicación. El sistema aprovechará el sensor de sonido para saber si el participante se encuentra en un evento masivo. Luego, gracias la geolocalización de los aparatos, además de la información de eventos en redes sociales el sistema le propone a amigos cercanos del usuario realizar una <i>vaca</i> . El usuario va anotando quienes le aportan y cuánto aportan, y por medio del lector de huella digital, los usuarios <i>firman</i> la vaca. La aplicación realiza los cálculos necesarios para redistribuir vueltas, si estas existen. La aplicación realiza el registro una vez hecho. La aplicación interopera con las cuentas de redes sociales y aplicaciones de comunicación (ej: Facebook Messenger, Facebook, Whatsapp, etc..) para notificar a los miembros de la recolección el estado de sus cuentas.		
Describe lo que el Stakeholder hace ahora o le gustaría poder hacer	El Stakeholder puede realizar la funcionalidad primaria de la aplicación. El sistema le propone un primer grupo cercano a él para empezar la recolección (o unirse a una en curso). Realiza el registro de los aportadores, anotando cuánto van aportando. Luego registra la compra, y le entrega el dinero de cambio a las personas correspondientes. Los participantes se comprometen a la vaca con sus huellas. El sistema se conecta a las cuentas de redes sociales y aplicaciones de comunicación (ej: Facebook Messenger, Facebook, Whatsapp, etc..) para notificar a los miembros de la recolección el estado de sus cuenta. El Stakeholder podrá revisar el historial cuando lo desee.		
Describe cualquier entrada provista o disponible al momento del inicio	El usuario le permite a la aplicación extraer información de su cuenta de red social. El usuario debe ingresar cada participante de la vaca con su aporte respectivo. Una vez se complete dicha recolección, debe registrar la compra.		
Describe el contexto de la operación	Un usuario que se encuentre en una fiesta (o cena) y que desee recolectar dinero entre sus amigos que se encuentren en el evento para poder comprar algún producto (ej: plato, botella de licor) costoso que sólo sea accesible si todos se cotizan.		
Describe cómo el sistema debe responder	El sistema debe registrar la información, realizar los cálculos para el cambio. Es capaz de reconocer si el usuario se encuentra en un evento masivo usando el sensor de sonido. El sistema debe ser capaz de detectar los dispositivos en una localización cercana. Debe guardar el historial. La información debe ser persistida de manera sencilla y eficiente. El sistema debe registrar las <i>firmas</i> de los participantes (lector de huella digital). El sistema debe enviar mensajes a través de la red social apropiada a cada contacto que deba dinero, indicando la suma que debe y a quien debe pagarse mientras este no haya pagado y su pago no haya sido registrado en el sistema. Si el usuario no tiene cuenta en red social, se debe realizar la comunicación por SMS.		
Describe las salidas que el sistema produce como resultado de la acción	El sistema tendrá almacenado el registro de la recolección, habrá hecho los cálculos para el cambio. Luego, de cierta periodicidad definida por el usuario, enviará una notificación a través de red social a cada contacto que deba dinero. Si dicho contacto no tiene cuenta en la red social, se enviará un SMS con dicha información.		
Describe quién o qué usa la salida y para que es utilizada	Los contactos que reciben los mensajes/notificaciones (o SMS) recordarán su deuda. La probabilidad de que contacten a la persona a la que le deben dinero y le paguen su deuda aumentarán.		

## 4.2 Casos de Uso

### Iteración 1

Título del Caso de Uso	Registro de vaca	ID del Caso de Uso	Cu-1-1
Descripción General del Caso de Uso			

El usuario registra a los que van aportando a la vaca. El sistema va almacenando la información. Luego el usuario registra la compra y el sistema calcula el cambio a devolver. Luego el sistema le envía SMS a los contactos que deban dinero.		
<b>Entidades Involucradas</b>		
Usuario		
<b>Precondiciones</b>		
El usuario está inscrito en el sistema		
<b>Flujo normal de Eventos</b>		
	<b>Usuario</b>	<b>Sistema</b>
1	Abre la aplicación y va ingresando la información de la vaca (Nombre del contacto + cantidad de dinero)	-Almacena la información a medida que el usuario la ingresa. -Estando conectado con los contactos en el celular, le ofrece autocompletar el nombre del contacto.
2	Finaliza la recolección	-Almacena la información
3	Registra la compra	-Hace el cálculo del cambio a dar.
4	Finaliza la vaca	-Persiste la información -Tiene el registro de los deudores
5	Define periodicidad del SMS a los deudores.	-Envía un SMS recordando su deuda a los deudores con la periodicidad configurada.
<b>Pos condiciones principales del caso de uso</b>		
La vaca se registra. El usuario y sus compañeros disfrutan del producto adquirido. Si quedan cuentas por hacer, el sistema tiene la información registrada y le enviará un SMS automático a los deudores.		
<b>Caminos de Excepción</b>		
Se interrumpe la vaca. Se podrá reanudar más adelante ya que la información es persistida.		

<b>Título del Caso de Uso</b>	Actualización de deuda	<b>ID del Caso de Uso</b>	Cu-2-1
<b>Descripción General del Caso de Uso</b>			
El usuario registra una deuda como pagada. El sistema registra la información, y deja de enviarle SMS a dicha persona.			
<b>Entidades Involucradas</b>			
Usuario			
<b>Precondiciones</b>			
El usuario está inscrito en el sistema			
<b>Flujo normal de Eventos</b>			
	<b>Usuario</b>	<b>Sistema</b>	
1	Abre la aplicación, ingresa a una vaca histórica.	-Despliega la información de una vaca histórica.	
2	Selecciona el deudor que realizó su pago y registra su pago.	-Actualiza la información del deudor.	
3	Cierra	-El sistema ya no le enviará SMS al deudor.	
<b>Pos condiciones principales del caso de uso</b>			
Se registra el pago. No se le enviarán más SMS al deudor.			
<b>Caminos de Excepción</b>			
Se interrumpe la actualización. En dicho caso se puede realizar más adelante, ya que la información es persistida.			

## Iteración 2

<b>Título del Caso de Uso</b>	Registro de vaca	<b>ID del Caso de Uso</b>	Cu-1-2
<b>Descripción General del Caso de Uso</b>			

El usuario registra a los que van aportando a la vaca. El sistema detecta la geolocalización del usuario. Le propone asociarse con sus amigos que se encuentren en el mismo lugar. El sistema va almacenando la información. Luego el usuario registra la compra y el sistema calcula el cambio a devolver. Luego el sistema le envía mensajes a través de redes sociales a los contactos que deban dinero. En caso en que no tengan cuenta de red social, se les envía un SMS.		
<b>Entidades Involucradas</b>		
<b>Usuario</b>		
<b>Precondiciones</b>		
El usuario está inscrito en el sistema		
<b>Flujo normal de Eventos</b>		
	<b>Usuario</b>	<b>Sistema</b>
1	Abre la aplicación y va ingresando la información de la vaca (Nombre del contacto + cantidad de dinero)	-Almacena la información a medida que el usuario la ingresa. -Detecta la geolocalización del usuario. Le ofrece un listado de sus amigos que se encuentran en el mismo lugar para realizar la vaca entre todos. -Detecta si el usuario está en un evento registrado en una red social. Si si, le ofrece un listado de sus amigos que están en el mismo evento. -Estando conectado con los contactos en el celular y las redes sociales del usuario, le ofrece autocompletar el nombre del contacto.
2	Finaliza la recolección	-Almacena la información
3	Registra la compra	-Hace el cálculo del cambio a dar.
4	Finaliza la vaca	-Persiste la información -Tiene el registro de los deudores
5	Define periodicidad del SMS a los deudores.	-Envía un mensajes a través de redes sociales recordando su deuda a los deudores con la periodicidad configurada. Si no tienen cuenta en las redes sociales, les envía un SMS.
<b>Pos condiciones principales del caso de uso</b>		
La <i>vaca</i> se registra. El usuario y sus compañeros disfrutan del producto adquirido. Si quedan cuentas por hacer, el sistema tiene la información registrada y le enviará un mensajes a través de redes sociales recordando su deuda a los deudores con la periodicidad configurada. Si no tienen cuenta en las redes sociales, les envía un SMS.		
<b>Caminos de Excepción</b>		
Se interrumpe la vaca. Se podrá reanudar más adelante ya que la información es persistida.		

<b>Título del Caso de Uso</b>	Actualización de deuda	<b>ID del Caso de Uso</b>	Cu-2-2
<b>Descripción General del Caso de Uso</b>			
El usuario registra una deuda como pagada. El sistema registra la información, y deja de enviarle mensajes a dicha persona.			
<b>Entidades Involucradas</b>			
<b>Usuario al que le pagan</b>			
<b>Precondiciones</b>			
El usuario está inscrito en el sistema			
<b>Flujo normal de Eventos</b>			
	<b>Usuario al que le pagan</b>	<b>Sistema</b>	

1	Abre la aplicación, ingresa a una vaca histórica.	-Despliega la información de una vaca histórica.
2	Selecciona el deudor que realizó su pago y registra su pago.	-Actualiza la información del deudor en los celulares que hicieron parte de la recolección.
3	Cierra	-El sistema ya no le enviará mensajes al deudor.
<b>Pos condiciones principales del caso de uso</b>		
Se registra el pago. No se le enviarán más mensajes al deudor.		
<b>Caminos de Excepción</b>		
Se interrumpe la actualización. En dicho caso se puede realizar más adelante, ya que la información es persistida.		

### Iteración 3

Título del Caso de Uso	Registro de vaca	ID del Caso de Uso	Cu-1-3
<b>Descripción General del Caso de Uso</b>			
El sistema detecta si el usuario está en un evento masivo usando el sensor de sonido y le propone iniciar una vaca. El sistema detecta la geolocalización del usuario. Le propone asociarse con sus amigos que se encuentren en el mismo lugar. El usuario registra a los que van aportando a la vaca. Los participantes firman en el sistema usando el lector de huellas digital. El sistema va almacenando la información. Luego el usuario registra la compra y el sistema calcula el cambio a devolver. Luego el sistema le envía mensajes a través de redes sociales a los contactos que deban dinero. En caso en que no tengan cuenta de red social, se les envía un SMS.			
<b>Entidades Involucradas</b>			
Usuario			
<b>Precondiciones</b>			
El usuario está inscrito en el sistema			
<b>Flujo normal de Eventos</b>			
	Usuario	Sistema	
1	Abre la aplicación y va ingresando la información de la vaca (Nombre del contacto + cantidad de dinero)	-Detecta si el usuario está en un evento masivo gracias al sensor de sonido. -Almacena la información a medida que el usuario la ingresa. -Detecta la geolocalización del usuario. Le ofrece un listado de sus amigos que se encuentran en el mismo lugar para realizar la vaca entre todos. -Detecta si el usuario está en un evento registrado en una red social. Si si, le ofrece un listado de sus amigos que están en el mismo evento. -Estando conectado con los contactos en el celular y las redes sociales del usuario, le ofrece autocompletar el nombre del contacto. -Ofrece la posibilidad de firmar la vaca usando la huella digital	
2	Finaliza la recolección	-Almacena la información	
3	Registra la compra	-Hace el cálculo del cambio a dar.	
4	Finaliza la vaca	-Persiste la información -Tiene el registro de los deudores	
5	Define periodicidad del SMS a los deudores.	-Envía un mensajes a través de redes sociales recordando su deuda a los deudores con la	

		periodicidad configurada. Si no tienen cuenta en las redes sociales, les envía un SMS.
<b>Pos condiciones principales del caso de uso</b>		
La <i>vaca</i> se registra. El usuario y sus compañeros disfrutan del producto adquirido. Si quedan cuentas por hacer, el sistema tiene la información registrada y le enviará un SMS automático a los deudores.		
<b>Caminos de Excepción</b>		
Se interrumpe la <i>vaca</i> . Se podrá reanudar más adelante ya que la información es persistida.		

Título del Caso de Uso	Actualización de deuda	ID del Caso de Uso	Cu-2-3
Descripción General del Caso de Uso			
El usuario registra una deuda como pagada. El sistema registra la información, y deja de enviarle SMS a dicha persona.			
Entidades Involucradas			
Usuario			
Precondiciones			
El usuario está inscrito en el sistema			
Flujo normal de Eventos			
	Usuario	Sistema	
1	Abre la aplicación, ingresa a una vaca histórica.	-Despliega la información de una vaca histórica.	
2	Selecciona el deudor que realizó su pago y registra su pago.	-Actualiza la información del deudor en los celulares que hicieron parte de la recolección.	
3	Cierra	-El sistema ya no le enviará mensajes al deudor.	
Pos condiciones principales del caso de uso			
Se registra el pago. No se le enviarán más SMS al deudor.			
Caminos de Excepción			
Se interrumpe la actualización. En dicho caso se puede realizar más adelante, ya que la información es persistida.			

## Sección 5. Sensores y Análisis - Iteración 3

Antes de proceder a la exposición de los diferentes puntos de vista, describimos los diferentes sensores y análisis de datos de estos realizados para la tercera iteración.

### 5.1 Sensor táctil (Pantalla Sensible al tacto)

Muu quiere priorizar la usabilidad de la aplicación debido a que quiere adaptarse a muchas situaciones y diferentes públicos, por lo tanto le da prioridad a la sencillez y a la interacción. Por esta razón se eligió hacer un uso de los gestos, una manera innovadora para que el usuario interactúe con la pantalla y las funcionalidades de la aplicación. Para implementar la interacción se usó el *GestureDetector* de Android para detectar los diferentes gestos, en particular: double tap, swipe, long press. Para facilitar el uso de este sensor se pusieron anotaciones sobre los gestos en las pantallas donde se pueden utilizar:



imagen: Captura de pantalla del main de la aplicación

Como se observa en la parte inferior del main se pueden ver los diferentes tipos de gestos para abrir las pantallas correspondientes. Se implementó también su uso en otras pantallas como:

- Abonar: En esta pantalla el administrador puede editar los saldos de los participantes, por ejemplo si un participante abono dinero a la vaca, o si por el contrario le fue pagado algún dinero de la vaca a un participante. En este caso se hizo el uso del gesto swipe, si se hace hacia afuera del celular (parte superior) significa que está saliendo plata de la vaca para ser entregada a algún participante, por el contrario, si se hace swipe entrando al celular (hacia abajo) significa que algún participante ha abonado dinero a la vaca. Como se puede ver, esta es una interacción sencilla y entendible el uso del gesto según su significado. Al igual que en el main hay indicaciones en la parte inferior de la pantalla.
- Envío de mensajes a los participantes: la aplicación tiene la opción de enviar mensajes por facebook y SMS por lo tanto el usuario puede seleccionar por cual de estos medios quiere enviar la información con tan solo hacer un gesto en la pantalla.

En conclusión, los gestos se implementaron con el fin de incrementar la usabilidad y la interacción con el usuario dándoles significados en el contexto de la solución.

## 5.2 Sensor de movimiento (Acelerómetro)

El acelerómetro es un sensor que permite medir la aceleración del movimiento del dispositivo, se implementa una clase especial que permite controlar la fuerza que se tomará como válida para el movimiento y el tiempo durante el cual debía ser ejercida dicha fuerza. Lo principal que se quiere resaltar con la aplicación en general es la usabilidad, por esto se buscaron interacciones simples e intuitivas y pantallas secuenciales fáciles de seguir.

Por esta razón el uso del acelerómetro busca apoyar este mismo atributo de calidad, inicialmente una de las funcionalidades de la aplicación es comunicar los saldos de las vacas a los participantes. Se pensó que no todo el tiempo los usuarios quieren enviar mensajes a todos los participantes ya sea por ahorro de minutos o datos, o por cualquier otro caso, por esto los usuarios pueden elegir a qué contactos enviarles mensajes. El acelerómetro busca aumentar la facilidad para el caso en que se quiera comunicar los saldos a todos los participantes, con solo agitar el celular automáticamente se mandan mensajes a todos los participantes de la vaca, sin necesidad de seleccionar uno por uno.

En particular la implementación de este sensor se hace en una pantalla denominada comunicación, una vez se han registrado los montos para todos los participantes y se finaliza la vaca, se visualizan los montos totales de cada participante y se pueden seleccionar algunos para comunicarse por SMS o Facebook, en esta pantalla sin necesidad de seleccionar los participantes se puede agitar el celular y enviar SMS a todos los participantes con su monto individual.

### 5.3 Sensor de posición (GPS) - Análisis de Clustering usando K-Means

La antena del Gps, la antena de Wi-fi o la antena de red del dispositivo puede proveer la ubicación en coordenadas de Longitud - Latitud del usuario del dispositivo.

Se usaron estos datos para hacer un análisis de clustering basado en K-Means para obtener el número de clusters que se encuentran a una distancia razonable. Se desarrollo en Python haciendo uso de las librerías Scikit-Learn, Pandas, Numpy y Flask.

Pandas es una librería en Python para el tratamiento de datos, ofrece funcionalidades para limpiar los datos y prepararlos para un análisis posterior. Scikit-learn es una librería que ofrece funciones de Machine Learning de manera sencilla. Es perfecta para la introducción de estos conceptos. En nuestro caso usamos K-Means<sup>1</sup>. Numpy es una librería básica en python para hacer uso de funciones matemáticas. Finalmente, Flask es una librería de Python para el despliegue de una aplicación como servidor REST.

K-Means tiene como objetivo particionar un set de n observaciones (datos) en k clusters (o grupos). Cada observación pertenece al cluster con la media más cercana (mean). La clave de este método es encontrar el número k de clusters óptimo para un cierto criterio. Este criterio puede ser la dispersión de los datos, como es en nuestro caso. El resultado son k cúmulos (clusters) cada uno con su media (mean) y su dispersión.

Este análisis es valioso en nuestro contexto. Le permite a la aplicación identificar conglomeraciones de usuarios. Así, la aplicación puede reconocer eventos en donde numerosos usuarios de la aplicación se conglomeran sin hacer uso de antenas más avanzadas (NFC, Bluetooth) que la geolocalización de los usuarios (GPS, Wifi, Red móvil). Esto permite ahorrar energía ya que no son necesarias todas las antenas.

Resumiendo, seguimos los siguientes pasos:

1. Cada dispositivo con la aplicación provee su geolocalización a un servidor REST mediante una petición HTTP (JSON) de manera periódica.
2. El servidor implementado en Python y desplegado en Flask realiza periódicamente (no necesariamente el mismo periodo)
  - a. Carga los datos usando Pandas
  - b. Realiza el análisis de K-Means
  - c. Obtiene los k-clusters
  - d. Pone a disposición esta información

---

<sup>1</sup> Scikit-Learn - K-Means documentation [online] consultado el 15 de Mayo de 2016 <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

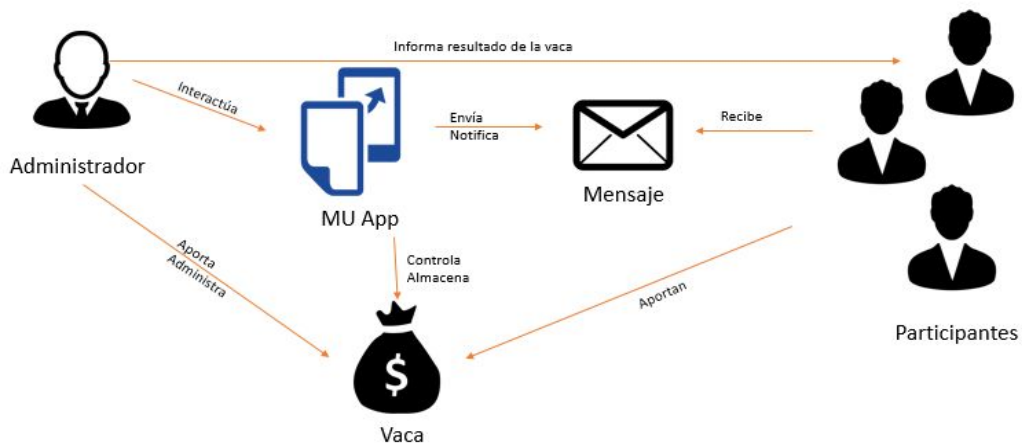
3. El dispositivo (periódicamente) le solicita la información de los k-clusters al servidor mediante una petición HTTP.
4. El dispositivo despliega esta información.

## Sección 6. Puntos de Vista y Modelos Arquitecturales

### 6.1 Contexto de la aplicación móvil

A continuación se presenta el contexto de la aplicación que si bien tiene los mismos perfiles para cada iteración la forma de interactuar varía según el dispositivo, la conectividad y los sensores utilizados.

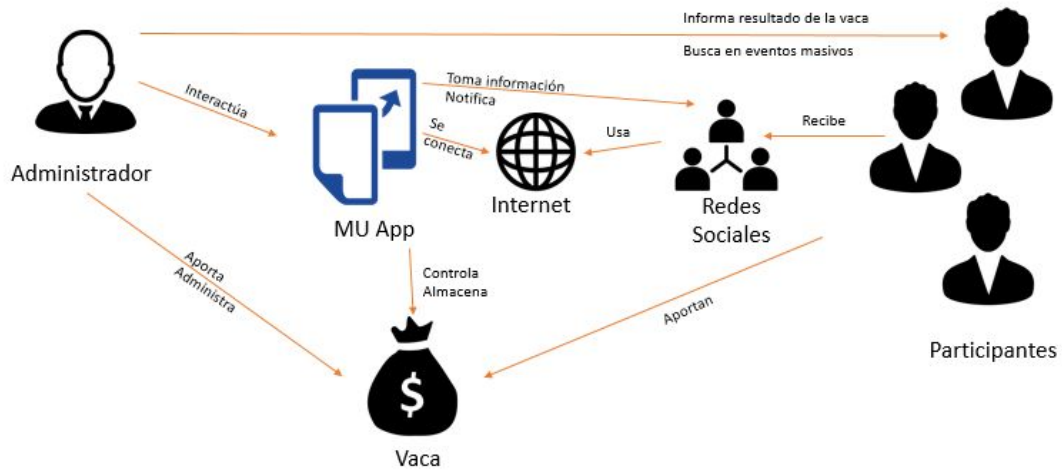
#### Iteración 1



Durante esta primera iteración el usuario (Administrador) podrá usar una interfaz gráfica de nuestra aplicación para administrar y controlar la vaca que realiza con los otros participantes y que almacenará en su registro histórico. Se hará uso de los recursos del dispositivo de Gama Baja para tener acceso a los archivos a las listas de contacto a la agenda y al calendario con el fin de comunicar a los participantes los resultados de la vaca y sus deudas en caso de que existan.



## Iteración 2

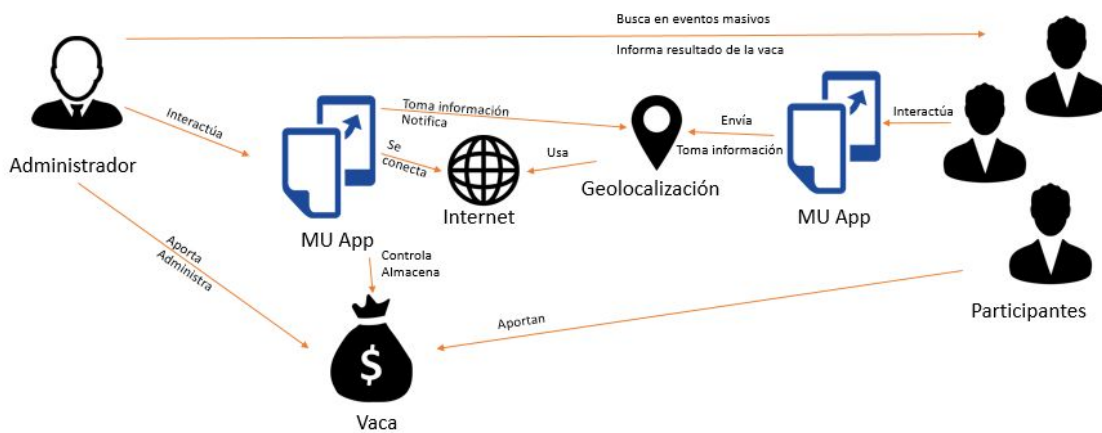


En la segunda iteración se complementa el desarrollo de la primera iteración al aprovechar la conectividad del dispositivo Gama Media permitiendo el acceso a la información de las redes sociales para verificar la existencia de eventos y los posibles participantes en estos. Además puede usar estas redes sociales para notificar los resultados de la vaca.

## Iteración

3

3



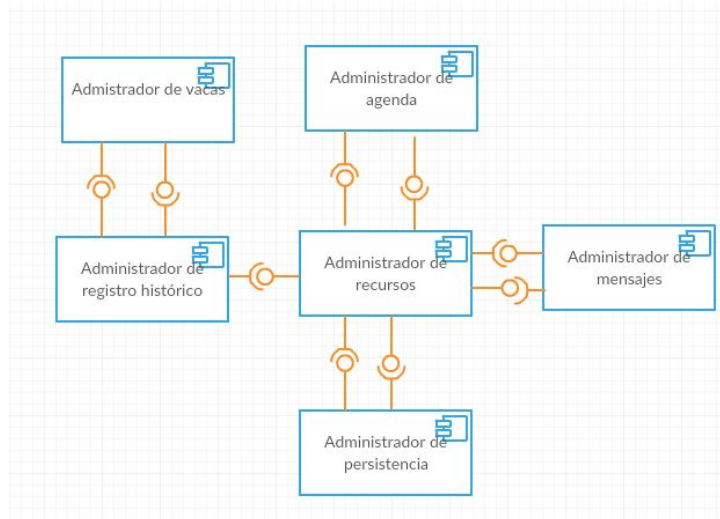
Para la tercera iteración haremos uso de la conectividad y el sensor para hacer uso de los servicios de localización (GPS) para comunicar la ubicación de los clientes con el fin de reunirlos en los eventos masivos. Adicionalmente los participantes y el administrador tendrán su aplicación para notificarse tanto la ubicación como los resultados de la vaca.

## 6.2 Punto de Vista Funcional

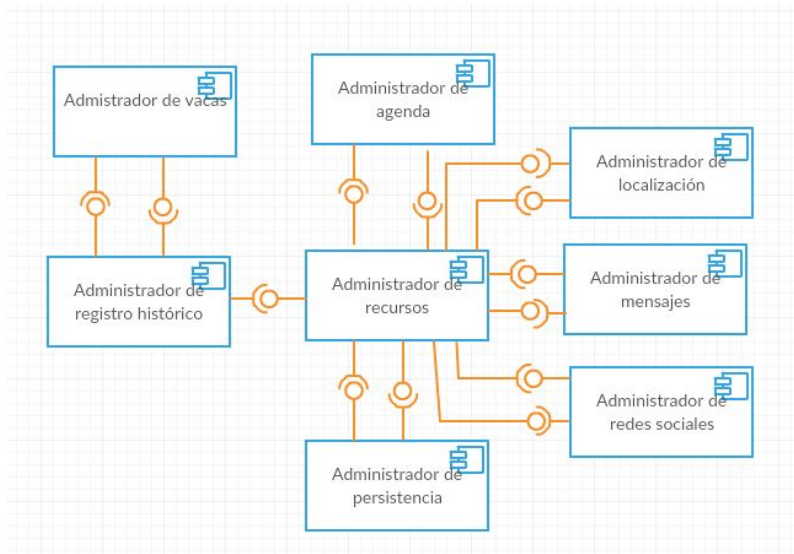
### 6.2.1 Modelo de Componentes

#### Iteración 1

La aplicación se organiza según varios componentes administrativos, que conllevan en sí funciones tanto de persistencia como de administración de recursos y tareas. A continuación se puede observar que dentro del dispositivo hay en consideración un total de 6 componentes generales:



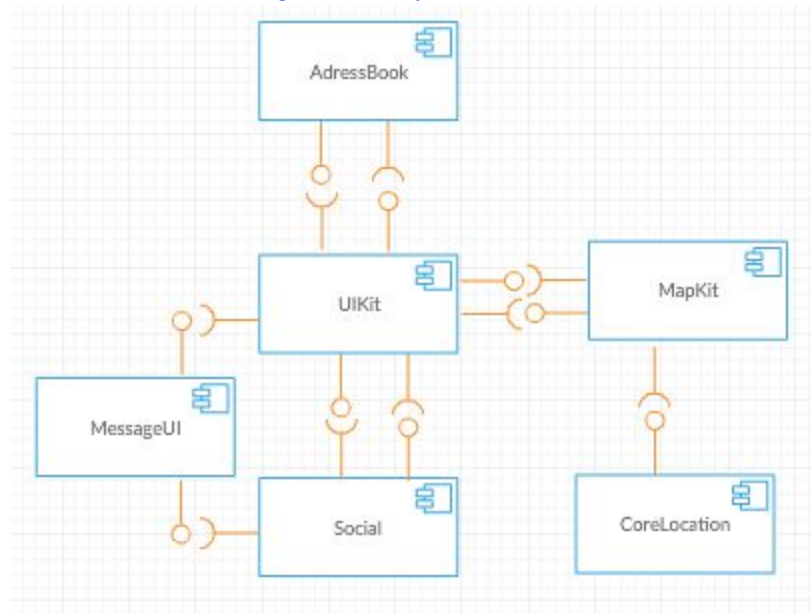
#### Iteración 2



En esta aplicación se incluye un manejador de redes sociales para la comunicación y se mantienen los diferentes componentes anteriores así como el manejador de la localización para la búsqueda de participantes.

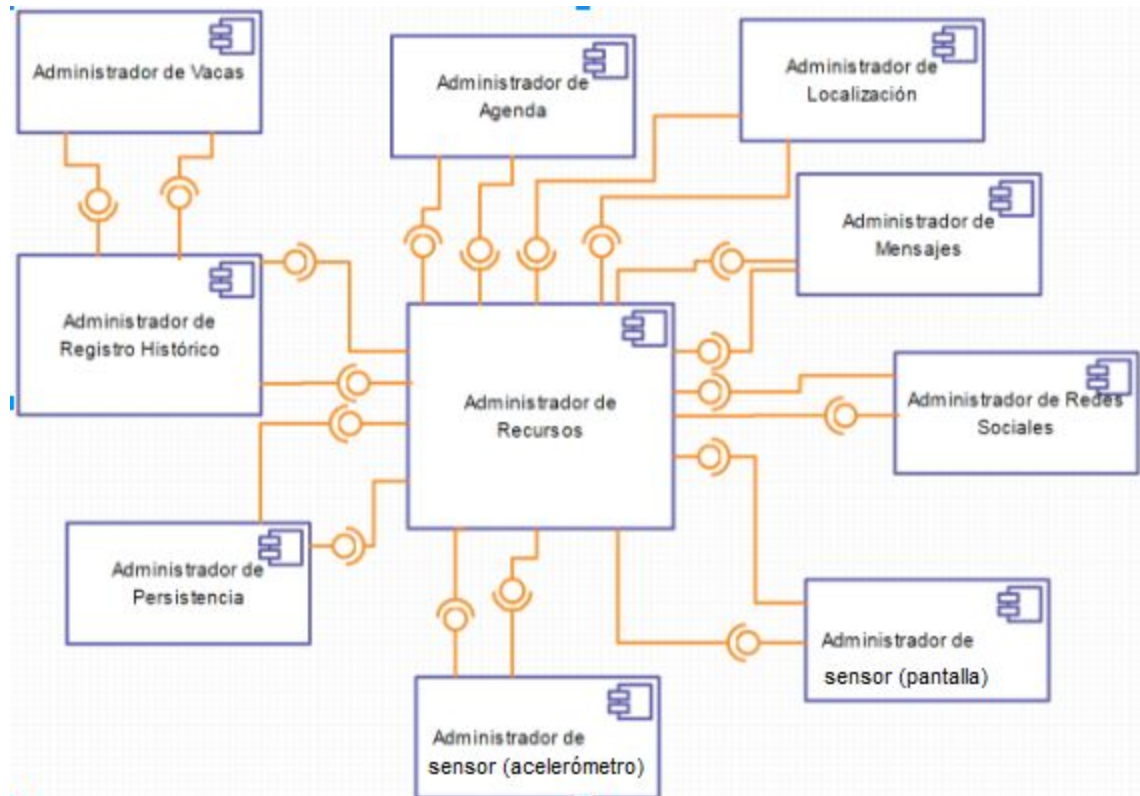
#### Segunda Plataforma

Para el desarrollo en IOs se utilizaron los siguientes componentes:



Para el administrador de recursos generales, objetos y viewControllers se usó el componente UIKit, para el administrador de localización se usaron dos componentes (MapKit y CoreLocation), para las redes sociales y la comunicación se usaron el componente Social y MessageUI, finalmente para los contactos se usó el componente de AddressBook. No se implementan los componentes para el manejo de agenda.

### Iteración 3

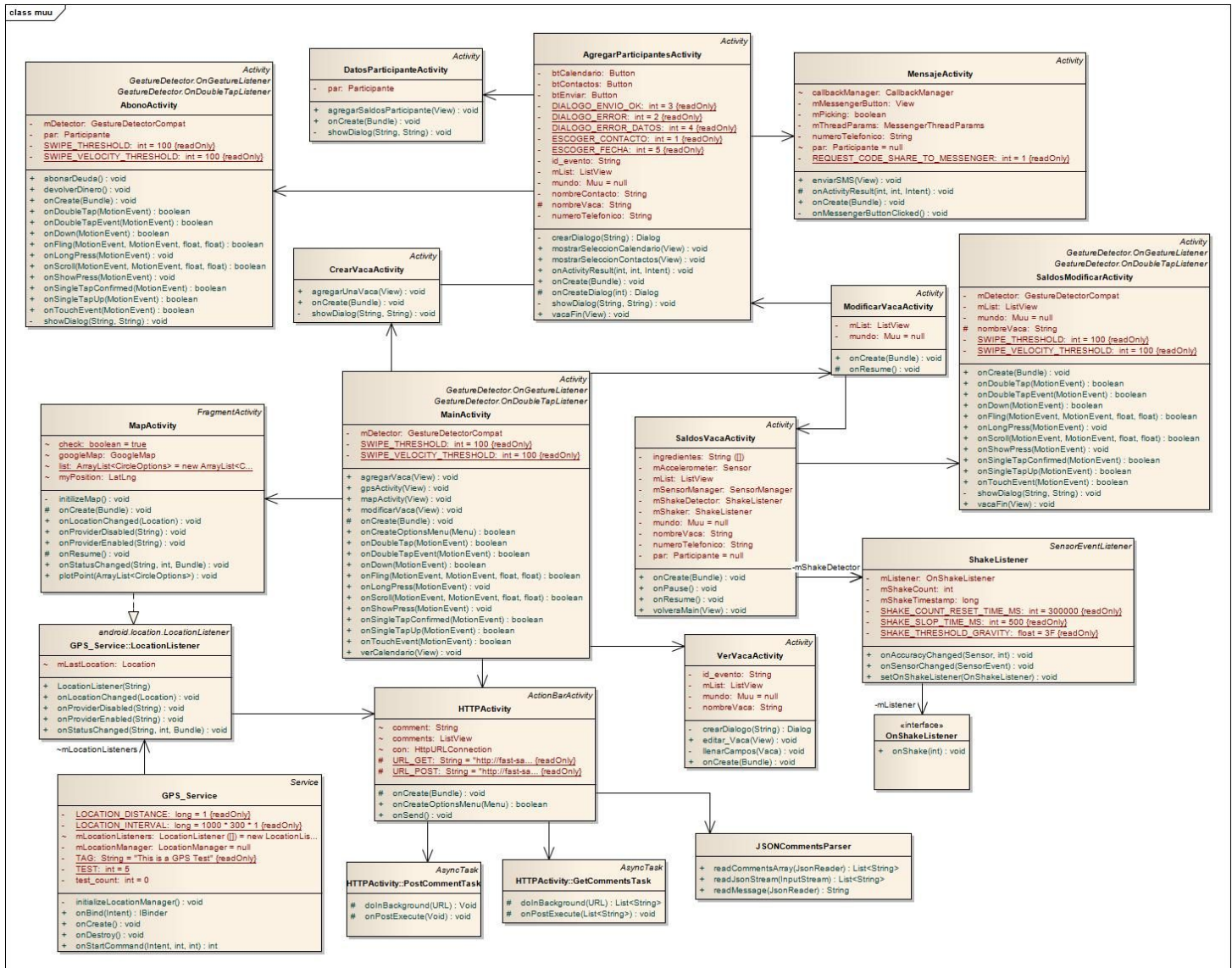


Aquí se incluye el manejador del sensor de gestos (acelerómetro) y el manejador de la pantalla (sensor táctil).

### 6.2.2 Modelo de Desarrollo

## Primera Plataforma

En Android, el diagrama completo es el siguiente.



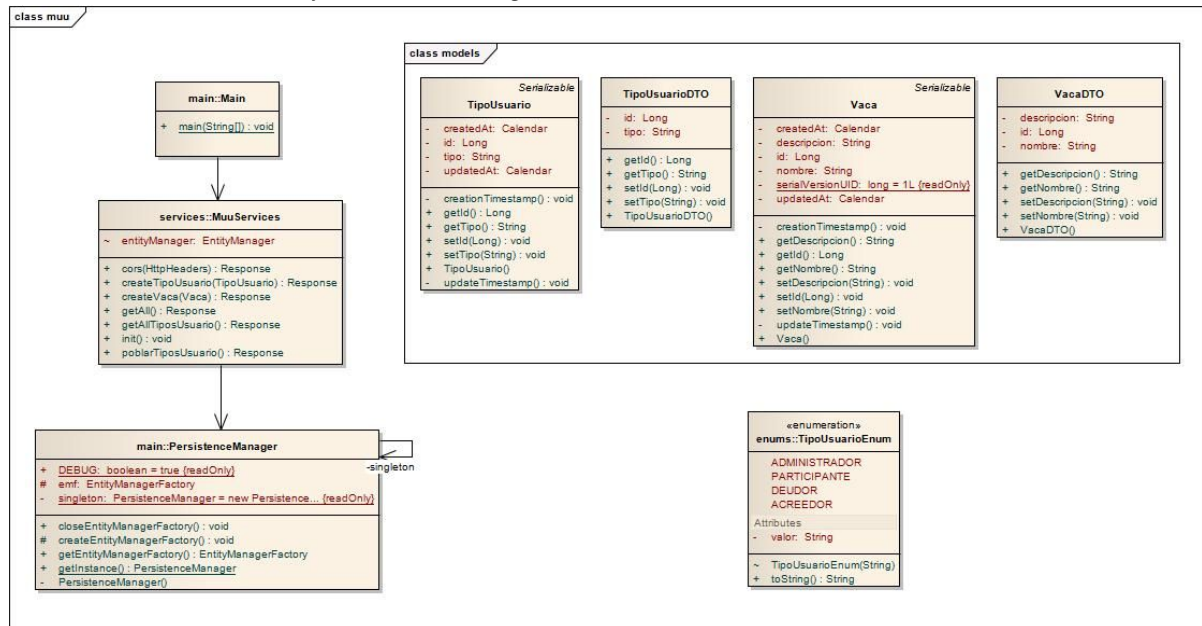
Se pueden observar las diferentes actividades (Clases) que se encuentran para el funcionamiento de la aplicación, partiendo de MainActivity, el inicio de ella. También vale la pena resaltar la presencia de GPS\_Service, servicio que adquiere periódicamente la localización del usuario y la envía hacia el servidor. Mediante HTTP activity se realizan las peticiones Http (POST y GET).

Finalmente, se puede observar el Shake Listener, que se encarga de manejar el acelerómetro.

La aplicación fue desarrollada en Android. El módulo de mapas hace uso de Google-Play Services 8.4. Se inició el desarrollo del módulo relacionado con Facebook con facebook-sdk-android 4.5 .

## Servidor

En el servidor se encuentra implementado de la siguiente manera,

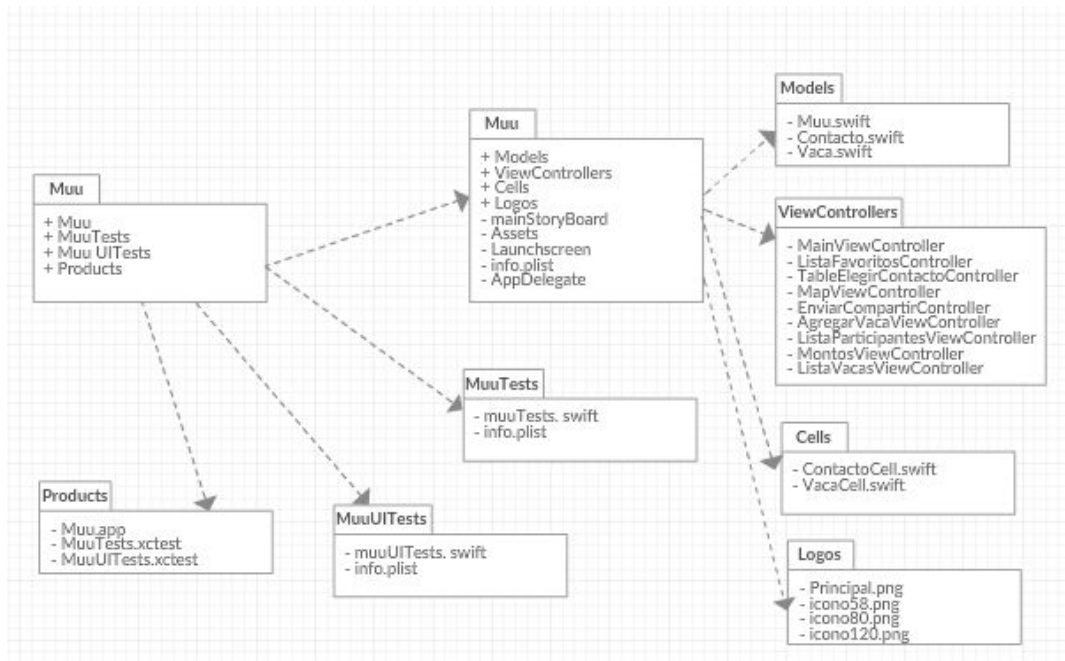


Está desarrollado en JAVA, compilado con Maven 3.3.9 y Java 1.8. Se encuentra desplegado a través de la plataforma HEROKU en la siguiente URL: <http://fast-savannah-95487.herokuapp.com> Para su desarrollo y despliegue se siguió la siguiente guía de desarrollo académica: <https://sistemasacademico.uniandes.edu.co/~isis2503/dokuwiki/doku.php?id=laboratorios:desplieguerest-cloud> Utiliza una base de datos PostGres de Heroku (como se especifica en la guía).

El servidor también cuenta con un script de Python, desplegado en Flask. Este fue desarrollado siguiendo la guía <http://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world> . A esta se le adjunto un script que se encarga del clustering usando K-means.

## Segunda Plataforma

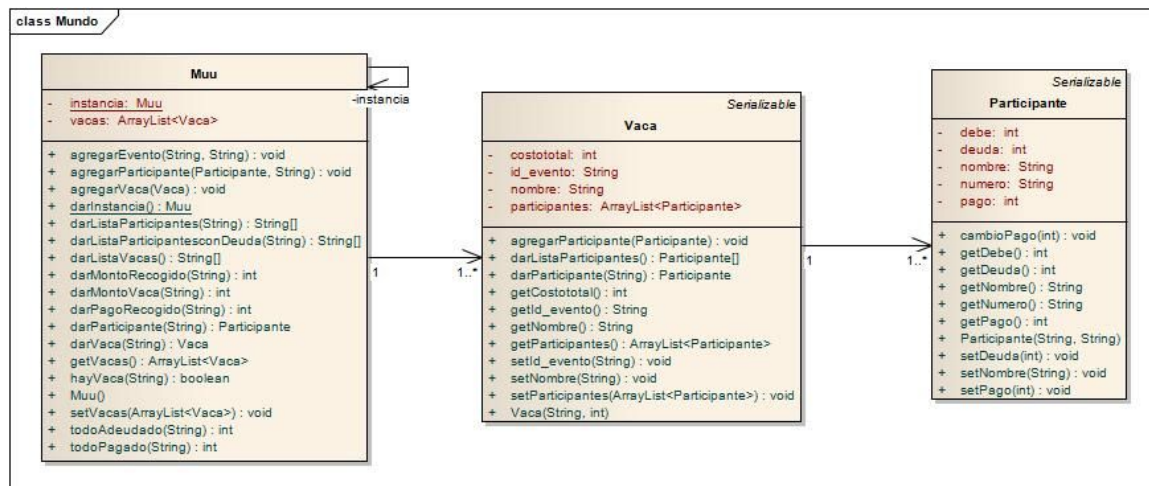




El desarrollo de esta plataforma se hizo en Xcode para IOs, el proyecto principal contiene los modelos de los objetos, los view controllers de las diferentes vistas, las celdas para la visualización en tablas y los logos que se usan en la aplicación.

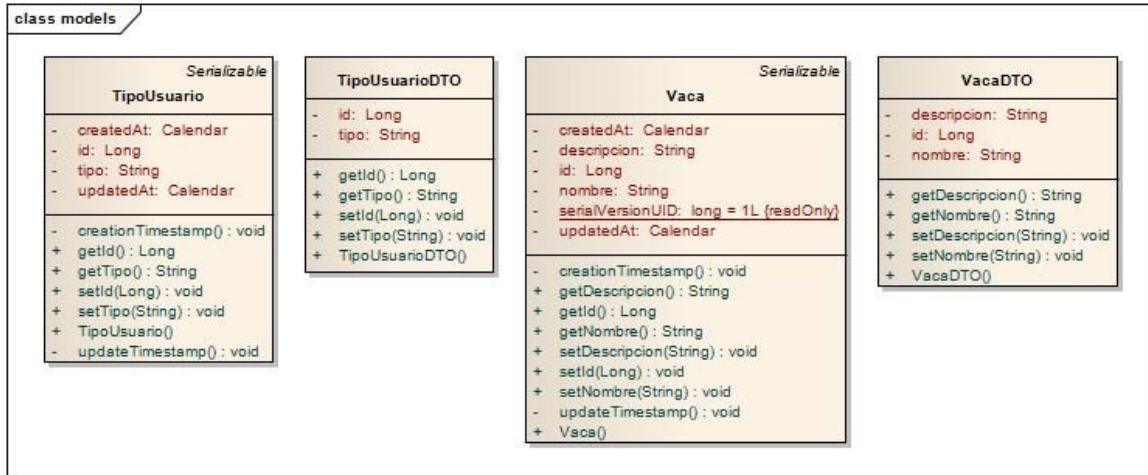
### 6.2.3 Modelo de Información

#### Primera Plataforma



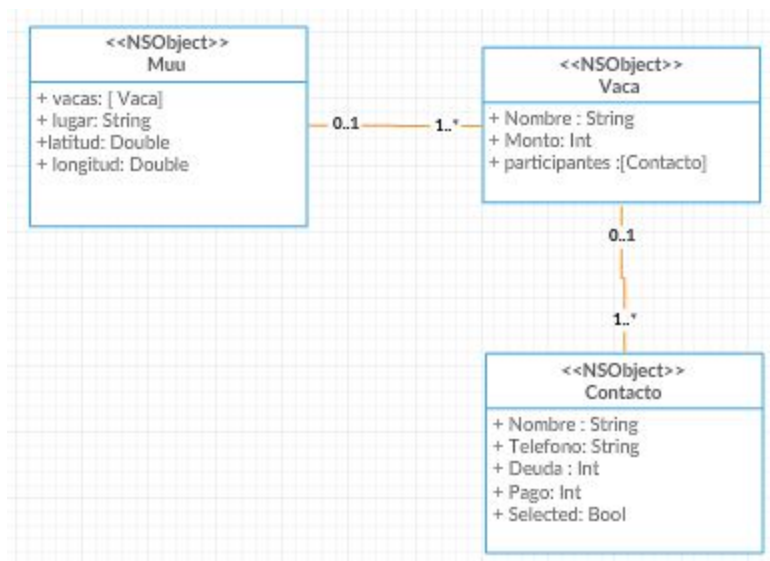
Se pueden observar los datos simples que utiliza la aplicación. Básicamente contienen la lista de vacas con sus listas respectivas de contactos, y atributos adicionales que las describen como su nombre o valores que representan el dinero.

El servidor maneja los mismos tipos de datos, salvo que en un nivel superior.



En el servidor se tienen los mismos tipos de datos. Se espera manejar por separado los diferentes tipos de usuario de la aplicación.

## Segunda Plataforma

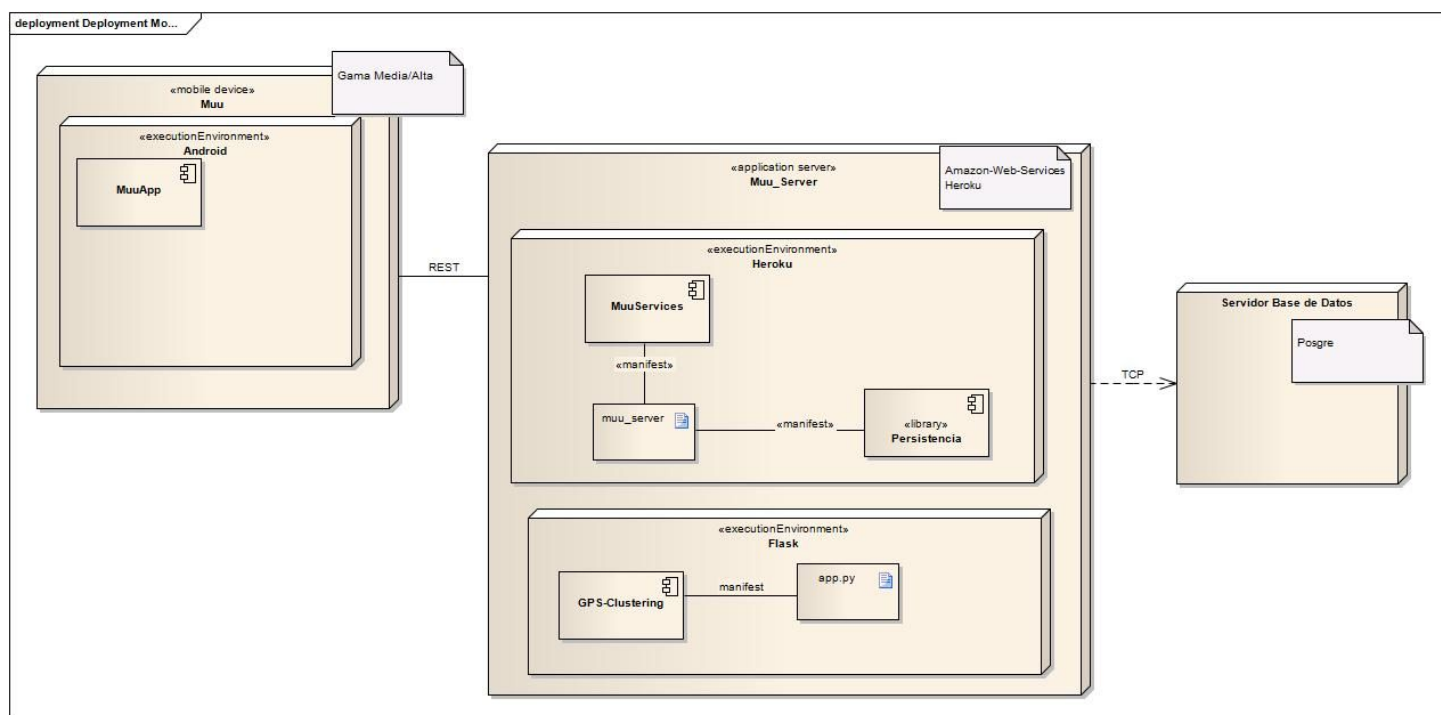


Las clases que se persisten son de tipo NSObject y básicamente contienen la lista de vacas con sus listas respectivas de contactos, y atributos adicionales que las describen como su nombre o valores que representan el dinero.

### 6.2.4 Modelo de Despliegue

Este es el modelo de despliegue actual del proyecto. La novedad que tenemos es la sección del servidor que se encuentra en Python. Esta es la que maneja el clustering de los datos GPS. Esta parte es desplegada en Flask.





## Sección 7. Relaciones entre los Puntos de Vista

Se muestra la relación entre el diagrama de contexto con el de componentes en los recursos sobre todo de comunicación: administrador de mensajes y administrador de redes sociales. Al detalle para esta iteración se pueden ver los componentes particulares para cada una de las plataformas y se relacionan en su descripción con los componentes descritos de forma general.

Finalmente se muestra la vista de desarrollo que es la que implementa los componentes, usándolos para crear las vistas y los modelos que controlan la ejecución de la aplicación. Así como una descripción detallada de los modelos que se persisten, con sus atributos.

## Sección 8. Evaluación de Arquitectura

Analice sus resultados a nivel de diseño y arquitectura con respecto a los escenarios y atributos de calidad esperados. Incluya conclusiones sobre el proceso, la tecnología y el producto logrado. Resalte los aspectos más sobresalientes del producto final, así como los aspectos no logrados y la razón para ello. Incluya los aspectos de evaluación de arquitectura y herramientas con respecto al desarrollo del producto completo en la segunda plataforma.

## Resultados de Diseño y Arquitectura

A nivel de diseño y arquitectura nuestra solución logra satisfacer los atributos de calidad. A nivel de **eficiencia**, nuestra solución satisface lo propuesto. Los cálculos complicados se hacen en el servidor, cuya capacidad de cómputo es más flexible que la de un dispositivo móvil. Los hace de manera transparente al usuario, por lo que este no tiene que realizar ninguna tarea complicada. Esto garantiza que a nivel de dispositivo la solución sea ligera y rápida. A nivel de **fiabilidad**, los cálculos obedecen a las limitaciones del dispositivo. Sin embargo, al ser cálculos simples, la fiabilidad es muy alta. El único verdadero limitante de fiabilidad es la calidad de las antenas del dispositivo. Si estas no son muy buenas, los cálculos de geolocalización se verán afectados. El servidor, al estar en la nube, garantiza una **disponibilidad** alta del servicio y una **integridad** a nivel de los datos. A nivel de **mantenimiento**, la implementación ha sido incremental y bien desarrollada. Las funcionalidades se encuentran separadas desde su implementación por lo que es simple agregar y retirar funcionalidades a la solución.. Finalmente, a nivel de **seguridad**, la solución se basa en la integridad del usuario en este primer instante

## Conclusiones

El producto final logra un uso intuitivo y avanzado de gestos táctiles y dinámicos del dispositivo. Esto facilita su uso y llama la atención de los usuarios al ser un factor diferenciador de la mayoría de aplicaciones que existen en el mercado. Además, la interfaz hace uso de un código de colores simple y atractivo para los usuarios.

La solución es en conjunto con un servidor REST. Esto permite desacoplar la persistencia de los datos globales hacia una base de datos independiente de los dispositivos, estableciendo estándares de comunicación haciendo uso de HTTP y JSON. Es fundamental el cómputo que se realiza en el servidor ya que permite hacer análisis de clustering (usando técnicas de Aprendizaje de Máquina) sobre los datos GPS de los usuarios para, de un lado, darle información a los administradores de la solución, y por otro lado hacer sugerencias a los usuarios sin que estos tengan que hacer un esfuerzo. Esta contribución *pervasive* es sin duda alguna un atractivo diferenciador de la solución.

Al respecto de aspectos no logrados de la solución está el análisis de datos a nivel de redes sociales que se tenía planteado. Se esperaba lograr tener acceso a los detalles de los eventos (en Facebook por ejemplo) a los cuales asiste cada usuario. Una vez obtenidos dichos datos, se esperaba tratarlos y asegurarse que los *amigos* del usuario se enteraran que el usuario estuviera planeando en asistir al evento y que la aplicación les propendiera a todos de crear una *vaca*. Este punto no se logró debido a la complejidad de conectar la aplicación con los datos de las redes sociales (ej: Facebook) a través de las respectivas APIs. La información a disposición no siempre era válida y había una clara variabilidad de las soluciones en la web al respecto. Sin embargo, se trabajó e invirtió tiempo en este punto por lo que es frustrante no haber logrado la funcionalidad y no saber precisamente el porqué de ello.

## Tecnologías

El desarrollo en **Android** (plataforma principal) tiene la ventaja de tener una documentación (no siempre válida y actualizada) a disposición del público. Muchas veces, el desarrollo se veía retrasar por causa de pequeños detalles sobre los cuales no se tenía conocimiento previo. Una ayuda y documentación más completa de parte del curso sería infinitamente valiosa acá.

**Android Studio** ofrece un fácil manejo de la descarga de APIs y se trata de un valioso ambiente de desarrollo y de despliegue. Sin embargo se vió una disparidad sobre las versiones que se ofrecen en los laboratorios (Turing & Waira) y las versiones necesarias para el desarrollo. Se termina entonces invirtiendo mucho tiempo en la correcta configuración del ambiente de desarrollo cuando este debería ser facilitado por los laboratorios a disposición (esta es la meta de estos).

El desarrollo en **IOS** (segunda plataforma) se vió como privilegiado a la hora de realizar una interfaz amigable y a la hora de conectarse con las redes sociales. Los módulos que se proveen fueron funcionales y amigables. Pero se

encontró una mayor dificultad para interactuar con aplicaciones propias del dispositivo, por ejemplo contactos, si bien se pudo extraer la información de dicha aplicación no era tan fácil de visualizar en la aplicación como en Android; así como el acceso a almacenar información en el dispositivo comparado con la facilidad presentado en este punto por Android. Por otro lado, no se pudo conectar la aplicación con el servidor implementado para la geolocalización, pero iOS permite una fácil conexión con su propio servicio que permite desplegar el mapa con la ubicación del usuario. Otra ventaja presentada por iOS fue el uso del storyboard que permitió una rápida curva de aprendizaje en el uso de las vistas y el diseño mismo de ellas. Finalmente, lo más limitante de esta plataforma es que solo permite su desarrollo en **Xcode** que solo puede instalarse en equipos Mac.

Para el uso de sensores se debería promover la interacción con tarjetas de tipo **Arduino** que facilitan el acceso a sensores básicos (temperatura, humedad, luz, etc.). Esto sería una ayuda importante y amplía las posibilidades de implementación de sensores de tipo *pervasive* en el marco del curso.

El uso de **Java**, **Maven** y **Heroku** fue útil para el despliegue de un servidor en la nube. La documentación de Heroku es bastante completa y amigable, factor que facilita el desarrollo.

Aunque el desarrollo en Java es transversal a lo largo del actual pregrado, fue de gran apoyo desarrollar parte del servidor en **Python** y **Flask**. Python ofrece funcionalidades avanzadas de manera sencilla y por ello es atractivo. En nuestro caso se hizo uso de funciones relativamente avanzadas de *Machine Learning* de manera rápida y clara. Esto corresponde perfectamente al marco de desarrollo del curso, lograr explorar técnicas y tecnología de manera sencilla, siendo consciente que soluciones masivas necesitan otro tipo de implementación más extensiva.