PROYECTO FINAL DE INGENIERÍA

APLICACIÓN DE SALUD PARA PROYECTO DATADONORS

Banzas, Alejandro Nahuel – LU131608

Ingeniería en Informática

Tutor/es:

Rossi, Bibiana Delmira, UADE

Prada, Federico, UADE

Gargiulo, Juan, The Wikilife Foundation

Octubre 6, 2016

UNIVERSIDAD ARGENTINA DE LA EMPRESA

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS

Equipo de Trabajo

Alumno

Alejandro Nahuel BANZAS, DNI: 35.067.563, LU: 131608, Ingreso 2008

Tutores de Proyecto Final

Mg. Ing. Bibiana Delmira ROSSI
Dr. Federico PRADA
MSc. Juan GARGIULO

Agradecimientos

A mi familia, por su apoyo durante toda la carrera.

A Juan Gargiulo, por ayudarme a contribuir al proyecto de la Fundación Wikilife.

A Bibiana y Federico por su ayuda y guía para la realización del presente Proyecto Final de Ingeniería.

A Microsoft por proveer su software de forma gratuita a estudiantes, gracias a lo cual fue posible realizar este Proyecto Final de Carrera.

Índice General

Índice de Figuras

Resumen

Este proyecto final **"Aplicación de salud para el proyecto DataDonors"** tiene como objetivo el desarrollo de una aplicación móvil que recuerde al usuario mediante notificaciones la toma de medicamentos y drogas; registro de las tomas, y control de stock de la/s droga/s a tomar, para informar la necesidad de reponer. Además, puede generar reportes de consumo y enviarlos por correo electrónico.

La información recolectada sobre las dosis consumidas por los usuarios es enviada de forma anónima a la base de datos del proyecto DataDonors de la fundación Wikilife que opera en Estados Unidos, mercado foco de esta aplicación.

La aplicación está desarrollada para el sistema operativo Windows Phone. La elección del sistema operativo, entre otras cosas, se basó en un análisis de mercado y aplicaciones existentes en los Stores de las plataformas iOS, Android, BlackBerry y Windows Phone.

Se logró completar una iteración del ciclo de desarrollo, dando como resultado un producto con todos los requerimientos funcionales implementados, según se detallan en este trabajo.

Para que DataDonors obtenga datos para su base de datos son necesarias aplicaciones como la que se desarrolló para este proyecto final, o integraciones con otros sistemas, como las que se plantean en la sección "futuras líneas de desarrollo". Para interactuar con esos datos, DataDonors posee una API REST.

La implementación de casos de prueba es de gran importancia porque permiten documentar el código y asegurarse que futuros cambios no afecten a la funcionalidad. Algunos de estos casos de prueba pueden ser usados independientemente de la aplicación que se esté desarrollando en el futuro.

Abstract

This final project **"Health application for DataDonors project"** aims to develop a mobile application to remind the user through notifications taking medication and drugs; log this information to control medication stock. In addition to that, the app allows users to send by email a report of drug usage.

The information collected on the dose consumed by users is sent anonymously to the database DataDonors project of Wikilife foundation that operates in the United States, market focus of this application.

The application is developed for the Windows Phone operating system. The choice of operating system, among other things, was based on a market analysis on iOS, Android, BlackBerry and Windows Phone application stores.

One development cycle iteration was completed, resulting in a product with all the functional requirements implemented, as detailed in this paper.

To get data, DataDonors needs mobile applications, as it was developed for this final project; or integration with other systems, such as those raised in the "future lines of development" section. To interact with that data, DataDonors has a REST API.

The implementation of test cases is of great importance because they allow documenting the code and ensure that future changes do not affect functionality. Some of these test cases can be used regardless of the application being developed in the future.

Contenido

El siguiente trabajo final tiene como fin el desarrollo de una aplicación cuya funcionalidad principal es la administración de dosis de medicamentos. Para obtener información de consumo anónimo que luego será compartido con un sistema de análisis (DataDonors), desarrollado por la fundación Wikilife.

Wikilife es una fundación estadounidense sin fines de lucro (ONG) que tiene como objetivo construir una base de datos mundial sobre salud y estilo de vida, que luego es compartida de forma abierta a través de APIs. DataDonors ya cuenta con una aplicación para la plataforma iOS (Apple), y una aplicación Web, para la captura de la información. Adicionalmente obtiene datos de las siguientes fuentes y aplicaciones (Nofal, 2014):

- Twitter
- Facebook
- LinkedIn
- Gmail
- Foursquare
- UP (Jowbone)
- Firbit
- Myfitness
- Runkeeper
- Nike+
- Dailymile
- iHealth
- Body Media
- Withing
- Fatsecret
- MoodPanda
- 23andMe

El presente documento tiene la siguiente estructura:

- Capítulo I: se presenta el planteamiento del problema, los objetivos que se buscan cumplir con el trabajo, la justificación, así como también el alcance y las limitaciones del sistema desarrollado.
- Capítulo II: se da una introducción a los Procesos de Desarrollo de Software y se describe la metodología de desarrollo sobre la cual se basó el presente Proyecto.
- Capítulo III: detalla el Modelo de Requerimientos del Proyecto.
- Capítulo IV: detalla la solución propuesta.
- Capítulo V: se presenta el producto final obtenido.
- Finalmente se presentan futuras líneas de desarrollo, conclusiones y la bibliografía consultada.

1. Capítulo I – Problema y objetivos

1.1. Objetivos

- Crear una aplicación que funcione en teléfonos con sistema operativo Windows Phone que permita al usuario administrar y recordar la toma de medicamentos, almacenar las tomas realizadas, poder generar y compartir reportes sobre consumos.

- Enviar los consumos generados por los usuarios, de forma anónima a una base de datos del proyecto DataDonors de la fundación Wikilife. (Nofal, 2014)

1.2. Alcance

DataDonors se nutre de diferentes orígenes de datos, entre ellos aplicaciones móviles que ayuden a los usuarios con diferentes casos de uso. Adicionalmente los datos en DataDonors están categorizados de la siguiente forma (Nofal, 2014):

- Social
- Actividad Física
- Nutrición
- Salud
- Genoma

Actualmente, DataDonors no cuenta con una aplicación que ayude a obtener datos en la categoría "Genoma", que está relacionada al consumo y reacción a drogas, entre otras. Por ello, este trabajo se orientó a esa rama.

La aplicación desarrollada funciona solo en la versión 8 o superior del sistema operativo Windows Phone. Esto no permite la ejecución de la aplicación en Windows Phone 7.x.

Adicionalmente, se desarrolló en XAML y C#, permitiendo, con poco esfuerzo, la reutilización del código para la replicación de la aplicación en otras plataformas, como iOS y Android.

La aplicación fue desarrollada para funcionar en Windows Phone versión 8 en adelante. Se excluyeron las versiones 7.x, ya que es una versión antigua y no cuenta con las APIs necesarias para la realización del trabajo.

La aplicación puede ser utilizada por cualquier usuario de Windows Phone en cualquier país. Wikilife solo cuenta con la base de datos de medicamentos de Estados Unidos, por lo que, usuarios de otros países no podrán utilizar la base de medicamentos, y la aplicación no podrá reportar los consumos a DataDonors.

La aplicación desarrollada se encuentra disponible en idioma inglés, aunque soporta múltiples idiomas, solo será necesario traducir la interfaz de usuario.

1.3. Acerca de DataDonors

DataDonors es una plataforma que recolecta y distribuye información sobre la salud durante todos los momentos de la vida, proveyendo a los individuos y comunidad científica, acceso sin precedentes a registros detallados de actividad en salud y estilo de vida. (Nofal, 2014)

La información recolectada incluye, salud y estilo de vida sobre nutrición, actividad física, estado de ánimo, sueño, diagnóstico de salud y sus tratamientos, análisis de laboratorio, medicina por imágenes, exámenes psicológicos, exámenes de coeficiente intelectual, y hasta información de ADN.

DataDonors provee acceso ilimitado a toda esta información para uso en investigaciones y desarrollo de áreas como la medicina, psicología, educación, nutrición, física, y comunidad científica en general.

La información es ingresada por usuarios anónimos, y queda asociada a un nombre de fantasía (nombre de usuario), para brindar un histórico al usuario. Nunca se identifica al usuario con sus datos personales.

DataDonors cuenta con una aplicación móvil para iOS, en este trabajo se realizó el desarrollo de una aplicación para Windows Phone, que agrega otro canal de ingreso de información para la base de datos del proyecto DataDonors. (Nofal, 2014)

1.4. Dispositivos Móviles

El mercado de teléfonos inteligentes (SmartPhones) se divide en 4 grandes competidores, Google con Android, Apple con iOS, Microsoft con Windows Phone, y RIM con BlackBerry. Estos dos últimos se encuentran con un porcentaje de mercado menor que Google y Apple. Windows Phone está en constante crecimiento, mientras que BlackBerry se encuentra en desaceleración, decreciendo su porción de mercado, posicionándolo como cuarto en la lista, como se puede observar en el siguiente gráfico.

La cantidad de teléfonos inteligentes vendidos en todo el mundo durante 2014 supera el billón de unidades (1,244 millones de unidades), un 28,4% más respecto a los 969 millones vendidos en 2013. Se estima que este crecimiento aumentará año a año. (Rivera, 2014)

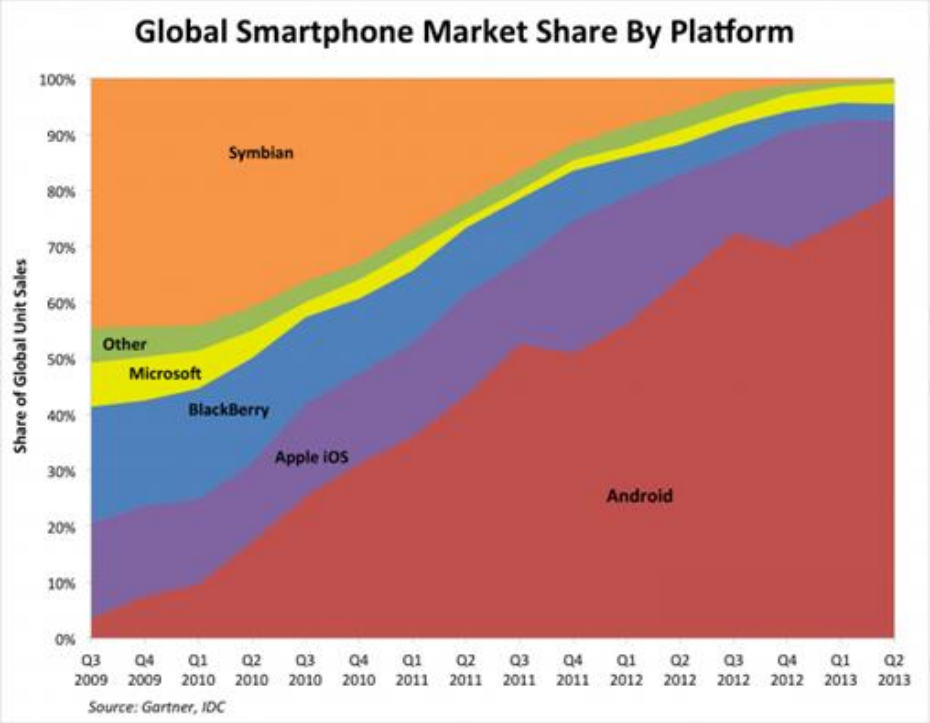

Figura I-4: Sistemas Operativos vs participación de mercado (segundo trimestre de 2013)

Hay que tener en cuenta que un teléfono inteligente es la combinación de un dispositivo y un Sistema Operativo. Existen dos casos particulares, donde el proveedor del hardware es el mismo que provee el software, BlackBerry y Apple. La dupla Samsung – Android es la actual líder de mercado. Le sigue Apple – iOS, con crecimiento desacelerado y Nokia – Windows Phone, que está ganando mercado (Llamas, 2013).

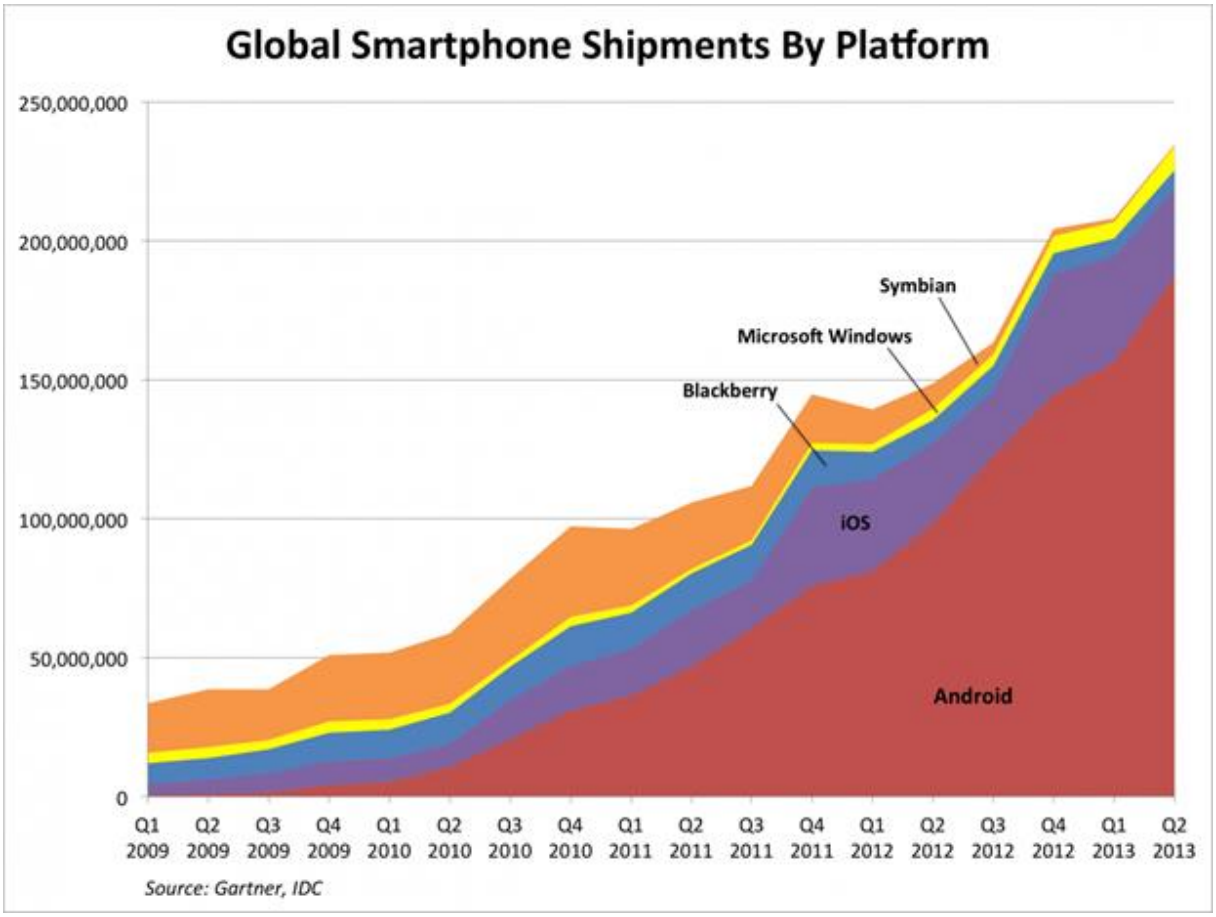

Figura I-5: Fabricantes de teléfonos vs. sistema operativo (segundo trimestre de 2013) en millones de unidades

Las plataformas móviles que dominan son Android (Danova, 2013), iOS (Llamas, 2013), Windows Phone y BlackBerry. Son analizadas individualmente en el anexo B. Windows Phone, la elegida para desarrollar este trabajo, se detalla a continuación:

1.4.1. Windows Phone

En 2008 Microsoft re-organizó su equipo de Windows Mobile y comenzó a trabajar en la nueva versión del Sistema Operativo, que se lanzó en octubre de 2010 con el nombre de Windows Phone 7, dejando atrás a Windows Mobile 6.5, ya que ninguna aplicación sería compatible con el nuevo Sistema Operativo, haciendo a Windows Phone una plataforma nueva (Petzold, 2010).

A diferencia de Windows Mobile orientado al mercado empresarial, Windows Phone está orientado al consumidor final, con una interfaz diferente (optimizada para pantallas táctiles), y con integración a servicios propios, como OneDrive, Skype, Xbox Live, directa desde el Sistema Operativo. Además de integración directa con redes sociales, contactos y cuentas de correo electrónico. Fue lanzado por primera vez en octubre de 2010, y su última versión en enero de 2015. Actualmente se está desarrollando la versión móvil de Windows 10, que reemplazará a Windows Phone 8.1. Todos los dispositivos que hoy lo tienen podrán actualizar. Todavía no hay fecha de lanzamiento (Fortin, 2016).

Windows Phone tiene una pantalla de inicio compuesta por íconos de aplicaciones elegidas por el usuario, llamados mosaicos vivos (live tiles), ya que éstos muestran información actualizada de las aplicaciones, interactuando con el usuario sin necesidad de accederlas.

Figura I-6: generaciones de equipos, requerimientos, versión del Sistema Operativo

Generación	Versión SO	Requerimientos Mínimos
Primera	7.0 (10/2010)	1 GHz de CPU, 512 MB de RAM, Cámara de 5 MP, Memoria interna de 8 GB
Segunda	7.5 (09/2011) 7.8 (01/2013)	800 MHz de CPU, 256 MB de RAM, Cámara de 5 MP, Memoria interna de 4 GB
Tercera	8.0 (10/2012) 8.1 (04/2014)	1 GHz de CPU, 512 MB de RAM, Cámara de 5 MP, Memoria interna de 8 GB

Los desarrolladores pueden distribuir sus aplicaciones a través del Windows Phone Store, una tienda de aplicaciones, disponible en 191 países, donde se pueden descargar y enviar comentarios. Para publicar una aplicación es necesario registrarse en el Store de Windows Phone. Estudiantes pueden acceder de manera gratuita, empresas e individuos deben pagar por única vez.

Para desarrollar una aplicación es necesario Visual Studio 2013 o superior e instalar el kit de desarrollo (SDK) de Windows Phone, el cual contiene varios emuladores y

herramientas adicionales para generar perfiles y probar la aplicación de Windows Phone en condiciones reales, simulando GPS, acelerómetro, velocidad de conexión a internet, entre otras.

La plataforma de desarrollo permite la utilización de diferentes combinaciones de lenguajes; JavaScript/HTML, C#, Visual Basic, C++, XAML. Desde la versión 8 del Sistema Operativo, este comparte las mismas APIs que Windows 8 para PCs, dando la posibilidad de reutilizar código en el desarrollo para ambas plataformas (Wastell, 2014).

1.5. Aplicaciones actuales

Se realizó un análisis de las aplicaciones existentes en diferentes plataformas y también en la elegida para la realización de este trabajo, Windows Phone. Se eligieron dentro del universo de aplicaciones las que tienen mayor calidad según los usuarios, basado en las de mayor cantidad de comentarios positivos y descargas. Se realiza este análisis de las aplicaciones en todas las plataformas, para conocer lo que el mercado está consumiendo, y utilizar ideas para lograr adopción de usuarios en esta nueva aplicación. El análisis detallado se encuentra en el anexo A de este trabajo. (Apple, 2014; Google Inc., 2014; Microsoft Corporation, 2014)

Como se puede apreciar en el siguiente cuadro, la aplicación Pill Reminder (Windows Phone), que es la más descargada, y competencia directa de la aplicación propuesta en este trabajo, carece de funciones que, si tienen otras plataformas y que serán incluidas en este trabajo, como la “Base de datos de drogas”, “información de medicamentos” y “lista de drogas”, entre otras.

Figura I-4: cuadro comparativo de aplicaciones analizadas

	MedCoach Medication Reminder (iOS)	Pill Reminder by Drugs.com (iOS)	AnyTimer Pill Reminder (Android)	Pill Reminder (Windows Phone)
Alerta de toma de medicamento	SI	SI	SI	SI
Por intervalos	SI	SI	SI	NO

Por tiempo fijo	SI	SI	SI	SI
Lista de alertas	SI	SI	SI	NO
Recarga de dosis	SI	SI	NO	NO
Base de drogas	SI	SI	NO	NO
Información de medicamentos	SI	NO	NO	NO
Notificaciones	SI	SI	SI	SI
Otras	Doctores y farmacias	-	-	-

2. Capítulo II - Proceso de desarrollo de software

2.1. Fases

Para el desarrollo de este trabajo final se utilizó el modelo iterativo e incremental, siguiendo en cada iteración las siguientes etapas:

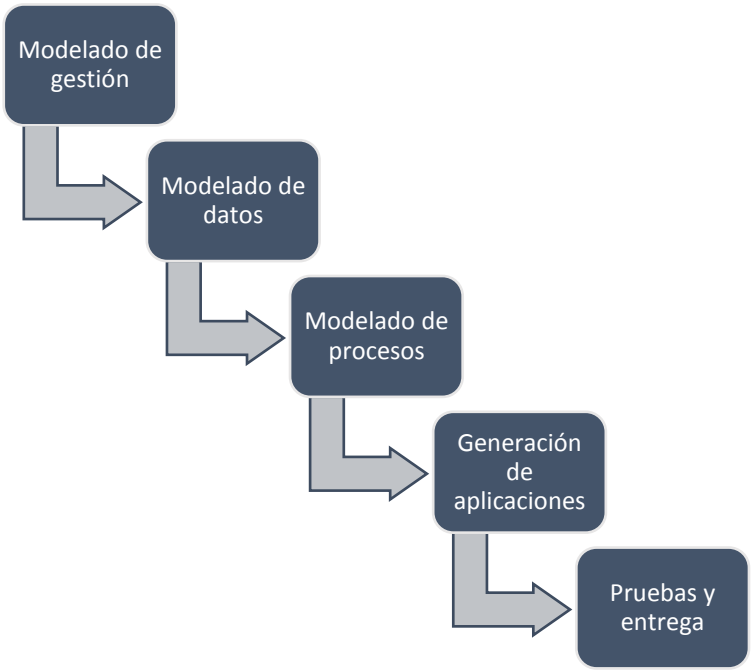

Figura II-1: etapas del proceso de desarrollo

2.2. Información de la plataforma Windows

Cualesquiera de los siguientes lenguajes de programación pueden utilizarse para crear aplicaciones que se ejecuten en Windows o Windows Phone, o en ambos, con el modelo de aplicaciones universales de Windows (compartiendo el mismo código para Windows y Windows Phone):

- C# o C++/CX y XAML
- C++/CX y Microsoft DirectX
- JavaScript y HTML5

Cada lenguaje de programación tiene su correspondiente modelo de aplicaciones. El modelo de aplicaciones es un conjunto de archivos y patrones de diseño que expresa la arquitectura de una implementación. Los modelos de aplicaciones se caracterizan por una serie de funciones, como el punto de entrada de la aplicación en el código ("entry point"), el diseño del archivo en la solución y la tecnología de presentación. Este trabajo está desarrollado en C# y XAML.

La tecnología de presentación del modelo de aplicaciones define la apariencia de la aplicación. Pueden utilizarse tres tecnologías distintas para crear aplicaciones de Windows en tiempo de ejecución: XAML, HTML5 y DirectX. En el siguiente grafico se muestran los distintos lenguajes de programación y visualización, tanto para aplicaciones (a la izquierda, "Windows Store Apps") como para programas de escritorio en Windows (a la derecha, "Desktop Apps") (Microsoft MSDN, 2015).

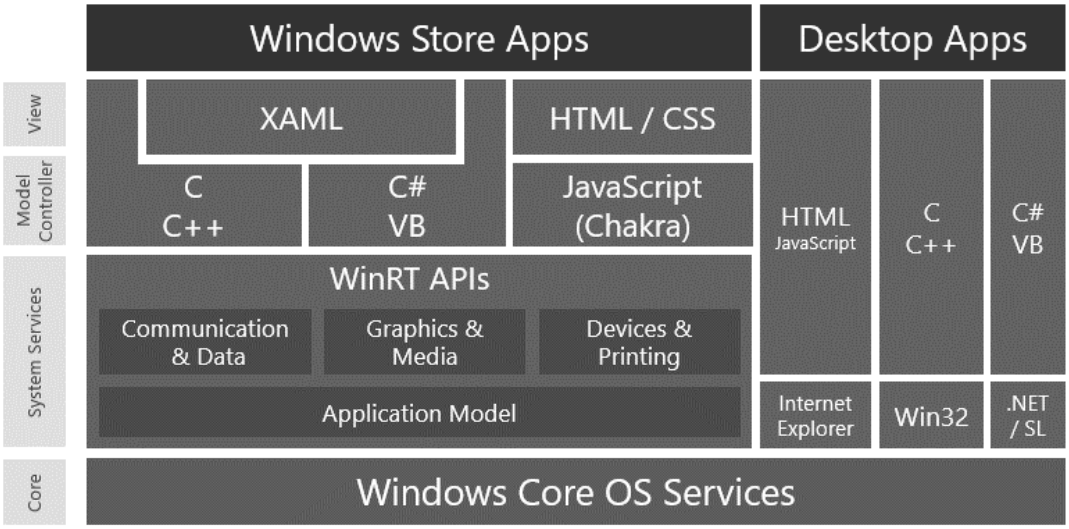

Figura II-2: modelos de desarrollo de Windows

2.3. Arquitectura MVVM

El patrón MVVM (Model View View-Model) es un patrón de arquitectura de software creado por Microsoft como una especialización del patrón PM (Presentation Model).

En gran parte está basado en el patrón MVC (Model View Controller); MVVM es una implementación pensada para plataformas de interfaz de usuario que soportan programación orientada a eventos, como WPF (Windows Presentation Foundation) y Silverlight en la plataforma Microsoft .NET, utilizada en este trabajo, y compuesto por tres elementos detallados a continuación. (Wildermuth, 2009).

Modelo: es una implementación del modelo de dominio de la aplicación, puede incluir modelo de datos y lógica de validación. En algunos casos, encapsula la capa de acceso a datos, que, en Windows Phone, puede incluir llamadas a servicios web o almacenamiento local.

Vista: está encargada de la estructura, diseño y apariencia de lo que el usuario ve en la pantalla. En Windows Phone, las vistas se escriben en lenguaje XAML.

Modelo de Vista: es intermediario entre el modelo y la vista, encargado de manejar la lógica de la vista. El Modelo de Vista provee datos que son fáciles de utilizar por la vista. Es encargado de obtener la información del modelo, y disponibilizarla a la vista, y puede darle formato simple para que la vista la pueda manejarla.

En el Modelo de Vista se implementan comandos que el usuario ejecuta desde la vista, también maneja la lógica de cambios de estado en la vista, por ejemplo, cuando la vista está aguardando una operación pendiente.

Además de entender las responsabilidades de cada elemento que compone MVVM, es importante conocer como estos componentes interactúan entre sí, como se muestra en la siguiente figura.

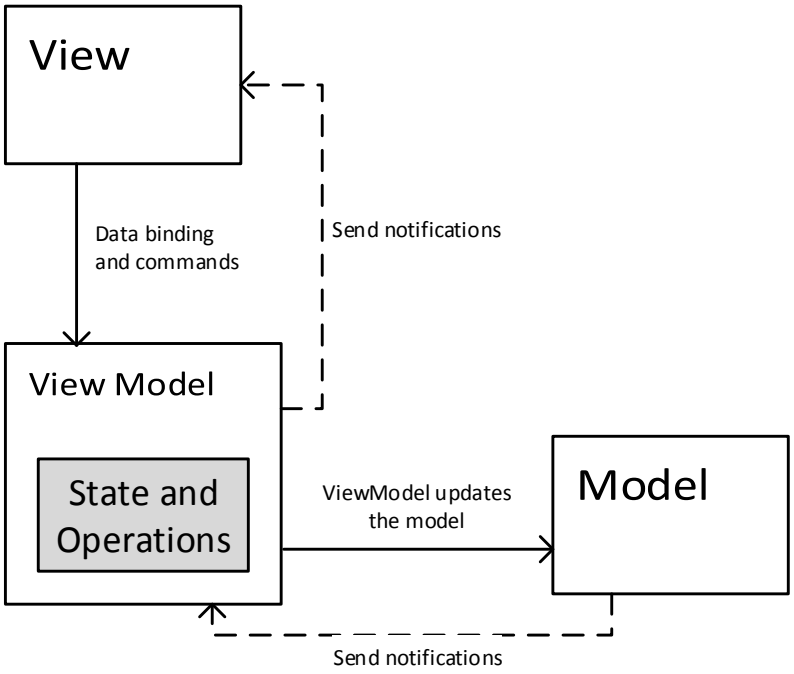

Figura II-3: implementación de patrón MVVM

MVVM aprovecha las capacidades de enlace de datos (data-binding) de Silverlight para gestionar la relación entre el Modelo de Vista y la vista, así como los comportamientos y triggers de eventos. Estas capacidades limitan la necesidad de poner la lógica de negocio en el "code-behind" de la vista.

En general, el Modelo de Vista interactúa con el modelo invocando métodos de las clases del modelo. El modelo también debe ser capaz de informar errores al modelo de vista mediante el uso de un conjunto estándar de eventos a los cuales el modelo de vista suscribe (el modelo no conoce el modelo de vista). También, el modelo tiene que ser capaz de informar de cambios en los datos al modelo de vista, se logra hacer esto utilizando eventos.

Durante el proceso de desarrollo, los desarrolladores y diseñadores pueden trabajar de forma más independiente en simultáneo, cada uno con sus correspondientes componentes.

Se pueden desarrollar tests unitarios al modelo, y al modelo de vista, sin utilizar ni necesitar la vista. Los tests al modelo de vista se pueden realizar simulando acciones y funcionalidad utilizada por y desde la vista.

Es fácil rediseñar la interfaz de usuario de la aplicación sin tener que modificar el código, ya que la vista es implementada completamente en XAML. Una nueva versión de la Vista funcionará sin modificar el Modelo de Vista (Britch, 2012).

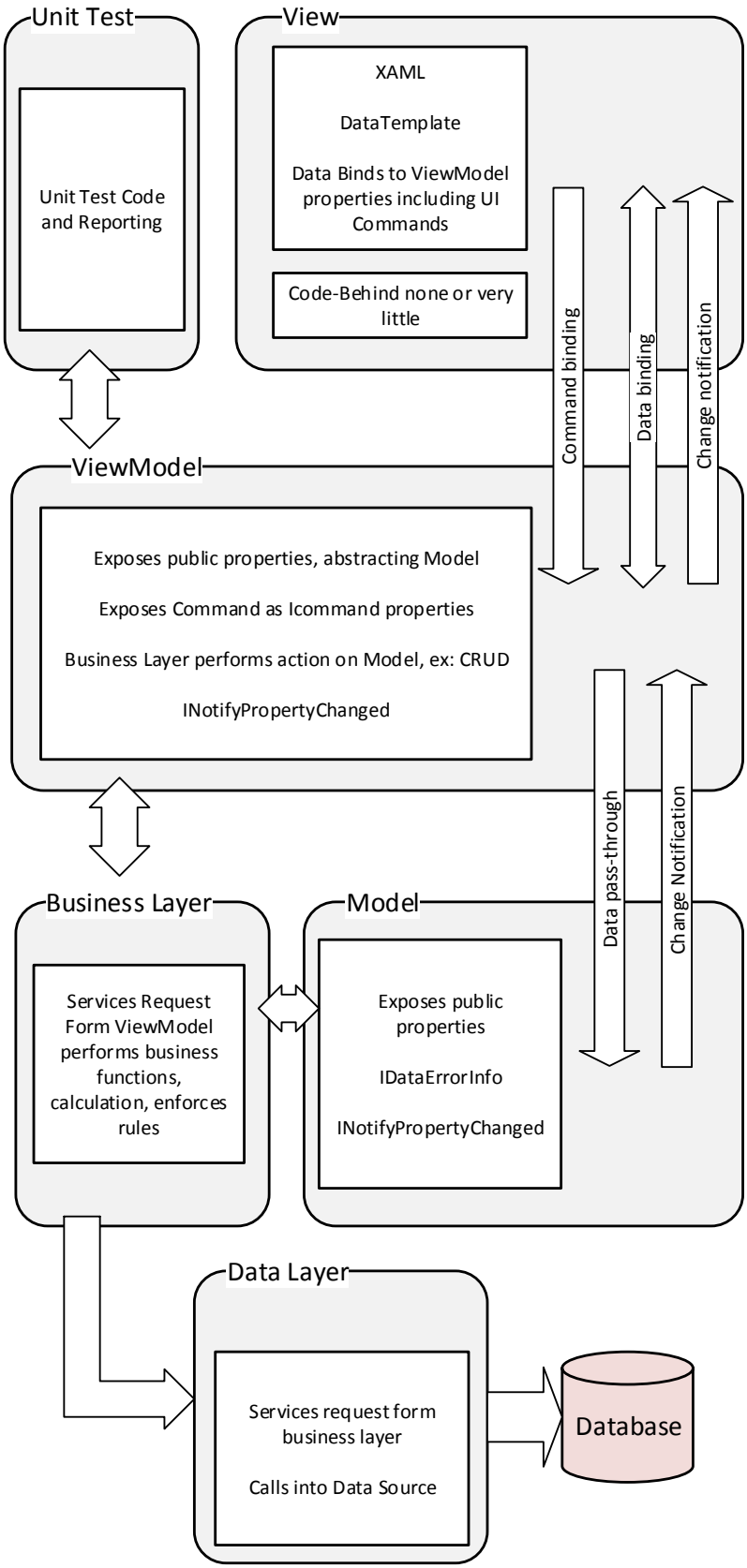

Figura II-4: Ejemplo de aplicación multicapa implementando el patrón MVVM

3. Capitulo III – Modelo de requerimientos

3.1. Diagrama de Casos de Uso del Sistema

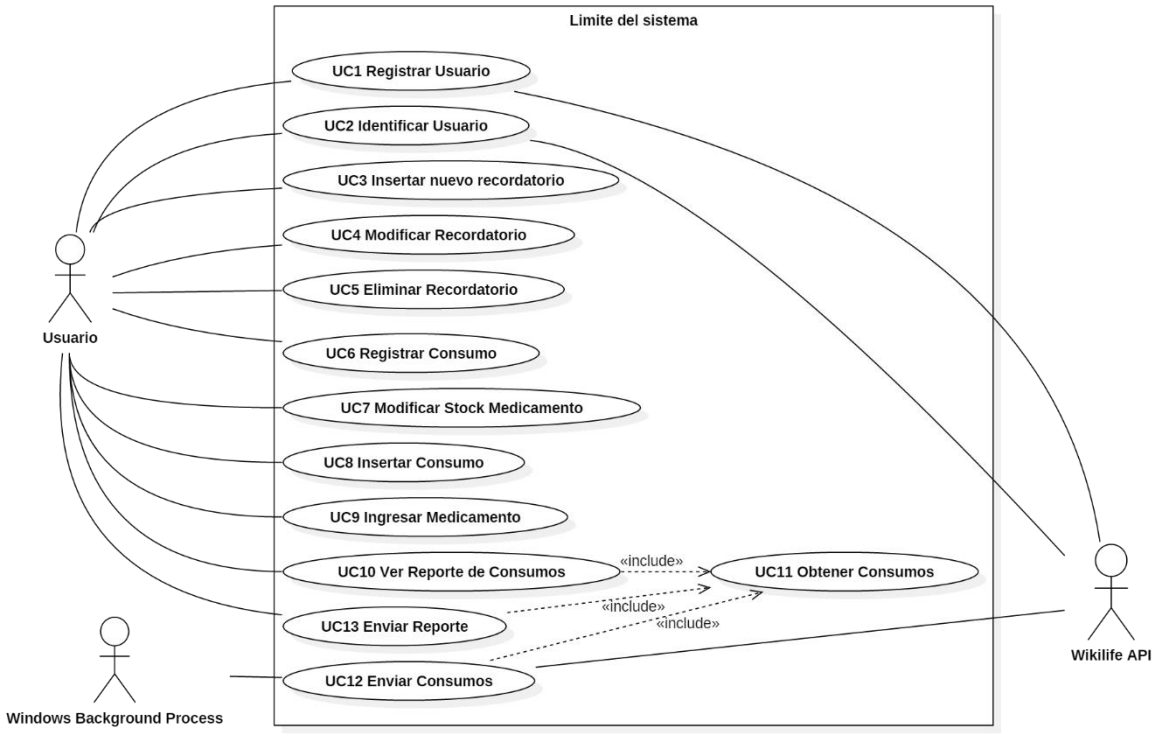

Figura III-1: Diagrama de casos de uso del sistema

3.2. Casos de uso detallados

UC1 - Registrar Usuario

Escenario principal

[1] El Usuario selecciona la opción Registrarse.

[2] El Sistema solicita el ingreso de:

-Nombre de usuario

-PIN (4 dígitos numéricos)

-Sexo

-Fecha de nacimiento

-Altura

-Peso

-Región

-Ciudad

[3] El Usuario ingresa los datos solicitados, y presiona el botón “Register”.

[4] El Sistema persiste los datos.

Caminos Alternativos:

[3a] No se ingresa un valor de peso y/o altura:

[3a 1] El Usuario no ingresa un valor en peso y/o altura.

[3a 2] El Sistema alerta al usuario que falta un valor en peso y/o altura.

[3a 3] Se une al escenario principal en el paso [4]

Excepciones

[*] Se cancela la operación:

[* 1] El Usuario cancela la operación.

[* 2] No se persisten los cambios.

[3b] Ingreso nombre de usuario existente:

[3b 1] El Usuario ingresa un nombre de usuario ya registrado.

[3b 2] El Sistema muestra un mensaje de error.

[3c] Ingreso numero negativo en peso y/o altura:

[3c 1] El Usuario ingresa un valor negativo en peso y/o altura.

[3c 2] El Sistema muestra un mensaje de error.

UC2 - Identificar Usuario

Escenario principal

[1] El Usuario selecciona la opción Identificarse.

[2] El Sistema solicita el ingreso de:

-Nombre de usuario

-PIN (4 dígitos numéricos)

[3] El Usuario ingresa los datos solicitados, y presiona el botón “Enter”.

[4] El Sistema valida los datos enviándolos a la API de DataDonors.

[5] El Sistema envía al Usuario a la pantalla principal.

Excepciones

[*] Se cancela la operación:

*1 El Usuario cancela la operación.

*2 No se persisten los cambios.

3b Ingreso nombre de usuario inexistente:

3b 1 El Usuario ingresa un nombre de usuario no registrado.

3b 2 El Sistema muestra un mensaje de error.

3b Ingreso password incorrecto:

3b 1 El Usuario ingresa un password invalido.

3b 2 El Sistema muestra un mensaje de error.

4a Error en el servidor de la API de DataDonors:

4a 1 El servidor de la API de DataDonors devuelve un código de error.

4a 2 El Sistema muestra un mensaje de error

UC3- Insertar Nuevo Recordatorio

Escenario principal

1 El Usuario selecciona la opción Nuevo Recordatorio.

2 El Sistema solicita el ingreso de:

-Medicamentos a tomar

-Horas del día

-Fecha de inicio

-Fecha de fin

-Frecuencia de uso

3 El Usuario ingresa los datos solicitados, y presiona el botón "Add".

4 El Sistema persiste los datos.

5 El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

3a No se ingresa ningún medicamento:

3a 1 El Usuario no ingresa medicamentos a consumir.

3a 2 El Sistema muestra un mensaje de error

3a 3 Se une al escenario principal en el paso 2

3b No se ingresa ninguna hora:

3b 1 El Usuario no ingresa horas de toma.

3b 2 El Sistema muestra un mensaje de error

3b 3 Se une al escenario principal en el paso 2

3c No se ingresa fecha de inicio:

3c 1 El Usuario no ingresa fecha de inicio.

3c 2 El Sistema utiliza la fecha actual como fecha de inicio del recordatorio.

3c 3 Se une al escenario principal en el paso 4

3d No se ingresa fecha de fin:

3d 1 El Usuario no ingresa fecha de fin.

3d 2 El Sistema marca el recordatorio como crónico.

3d 3 Se une al escenario principal en el paso 2

Excepciones

* Se cancela la operación:

* 1 El Usuario cancela la operación.

* 2 No se persisten los cambios.

3b Ingreso un medicamento que existe en otro recordatorio:

3b 1 El Usuario ingresa un medicamento ya registrado.

3b 2 El Sistema muestra un mensaje alertando al usuario.

3b 3 Se une al escenario principal en el paso 4

UC4 - Modificar Recordatorio

Escenario principal

1 El Usuario selecciona la opción Editar Recordatorio.

2 El Sistema solicita al Usuario que elija el recordatorio a modificar.

3 El Usuario selecciona el recordatorio, y presiona el botón “Edit”.

4 El Sistema obtiene la información del recordatorio y la muestra en un formulario para editar.

5 El Sistema solicita el ingreso de:

-Medicamentos a tomar

-Horas del día

-Fecha de inicio

-Fecha de fin

-Frecuencia de uso

6 El Usuario ingresa los datos solicitados, y presiona el botón “Edit”.

7 El Sistema persiste los datos.

8 El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

6a No se ingresa ningún medicamento:

6a 1 El Usuario no ingresa medicamentos a consumir.

6a 2 El Sistema muestra un mensaje de error

6a 3 Se une al escenario principal en el paso 5

6b No se ingresa ninguna hora:

6b 1 El Usuario no ingresa horas de toma.

6b 2 El Sistema muestra un mensaje de error

6b 3 Se une al escenario principal en el paso 5

6c No se ingresa fecha de inicio:

6c 1 El Usuario no ingresa fecha de inicio.

6c 2 El Sistema utiliza la fecha actual como fecha de inicio del recordatorio.

6c 3 Se une al escenario principal en el paso 5

6d No se ingresa fecha de fin:

6d 1 El Usuario no ingresa fecha de fin.

6d 2 El Sistema marca el recordatorio como crónico.

6d 3 Se une al escenario principal en el paso 5

Excepciones

* Se cancela la operación:

* 1 El Usuario cancela la operación.

* 2 No se persisten los cambios.

6a Ingreso un medicamento que existe en otro recordatorio:

6a 1 El Usuario ingresa un medicamento ya registrado.

6a 2 El Sistema muestra un mensaje alertando al usuario.

6a 3 Se une al escenario principal en el paso 7

UC5 - Eliminar Recordatorio

Escenario principal

[1] El Usuario selecciona la opción Eliminar Recordatorio.

[2] El Sistema alerta al usuario y solicita que confirme la operación.

[3] El Usuario confirma la operación.

[4] El Sistema elimina el recordatorio.

[5] El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

[3a] No se confirma la operación:

[3a 1] El Usuario no confirma la operación.

[3a 2] El Sistema informa al usuario que la operación fue cancelada.

Excepciones

[*] Se cancela la operación:

[* 1] El Usuario cancela la operación.

[* 2] No se eliminan los cambios.

UC6 - Registrar Consumo

Escenario principal

[1] El Sistema lanza un recordatorio de consumo.

[2] El Usuario acepta el recordatorio.

[3] El Sistema reduce el stock del medicamento consumido.

[4] El Sistema registra el consumo en persistencia local.

[5] El Sistema envía los datos de consumo a la API de DataDonors.

[6] El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

[2a] Ignorar recordatorio:

[2a 1] El Usuario ignora el recordatorio.

2a 2 Se une al escenario principal en el paso 6

2b Posponer recordatorio:

2b 1 El Usuario pospone el recordatorio.

2b 2 El Sistema modifica la fecha de la instancia de recordatorio.

2b 3 Se une al escenario principal en el paso 6

3a Stock en cero:

3a 1 El Sistema detecta falta de stock.

3a 2 El Sistema muestra un mensaje informativo.

Excepciones

5a Error en el servidor de la API de DataDonors:

5a 1 El servidor de la API de DataDonors devuelve un código de error.

5a 2 El Sistema marca el consumo en estado "no sincronizado"

UC7 - Modificar Stock Medicamento

Escenario principal

1 El Usuario selecciona la opción Modificar Medicamento.

2 El Sistema solicita al Usuario que elija el medicamento a modificar.

3 El Usuario selecciona el medicamento.

4 El Sistema solicita al Usuario que ingrese la cantidad que posee.

5 El Usuario ingresa el valor de la cantidad del medicamento seleccionado.

6 El Sistema persiste el dato.

7 El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

5a El valor ingresado es cero:

5a 1 El Usuario ingresa cero.

5a 2 El Sistema alerta al usuario que ingresa falta de stock.

5a 3 Se une al escenario principal en el paso 6

Excepciones

* Se cancela la operación:

* 1 El Usuario cancela la operación.

* 2 No se persisten los cambios.

5b Ingreso de número negativo:

5b 1 El Usuario ingresa un número negativo.

5b 2 El Sistema muestra un mensaje de error.

UC8 - Insertar Consumo

Escenario principal

1 El Usuario selecciona la opción Ingresar Consumo.

2 El Sistema busca los recordatorios pasados que no registraron consumo.

3 El Sistema lista los consumos y solicita al usuario que seleccione cuales fueron consumidos.

4 El Usuario selecciona los consumos que realizo.

5 El Sistema guarda los consumos seleccionados.

6 El Sistema modifica el stock de los medicamentos.

7 El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

4a No se selecciona ningún consumo:

4a 1 El Usuario no selecciona ningún consumo.

4a 2 Se une al escenario principal en el paso 3

6a Stock en cero:

6a 1 El Sistema detecta falta de stock.

6a 2 El Sistema muestra un mensaje informativo.

Excepciones

* Se cancela la operación:

* 1 El Usuario cancela la operación.

* 2 No se persisten los cambios.

UC9 - Ingresar Medicamento

Escenario principal

[1] El Usuario selecciona la opción Nuevo Medicamento.

[2] El Sistema solicita el ingreso de:

-Nombre

-Tipo de dosis

-Tamaño de dosis

-Cantidad de dosis por caja

-Cantidad de cajas que posee

[3] El Usuario ingresa los datos solicitados.

[4] El Sistema persiste los datos.

[5] El Sistema envía al Usuario a la pantalla principal.

Caminos Alternativos:

[3a] No se ingresa cantidad de dosis por caja:

[3a 1] El Usuario no ingresa la cantidad de dosis por caja.

[3a 2] El Sistema muestra un mensaje alertando que no será informado del stock que posee.

[3a 3] Se une al escenario principal en el paso [4]

[3b] No se ingresa cantidad de cajas que posee:

[3b 1] El Usuario no ingresa cantidad de cajas que posee.

[3b 2] El Sistema muestra un mensaje alertando que no será informado del stock que posee.

[3b 3] Se une al escenario principal en el paso [4]

Excepciones

[*] Se cancela la operación:

[* 1] El Usuario cancela la operación.

[* 2] No se persisten los cambios.

[3b] Ingreso un medicamento, tipo de dosis, y tamaño de dosis existente:

[3b 1] El Usuario ingresa un medicamento ya registrado.

[3b 2] El Sistema muestra un mensaje alertando al usuario.

UC10 - Ver Reporte de Consumos

Escenario principal

[1] El Usuario selecciona la opción Reporte de Consumos.

[2] El Sistema solicita el ingreso de:

-Fecha inicio

-Fecha fin

[3] El Usuario ingresa los datos solicitados.

[4] <include> UC11 - Obtener Consumos.

[5] El Sistema filtra los consumos realizados entre la fecha inicio y fecha fin.

[6] El Sistema muestra los datos.

Caminos Alternativos:

[3a] No se ingresa un valor de fecha inicio y/o fecha fin:

[3a 1] El Usuario no ingresa un valor en peso y/o altura.

[3a 2] <include> UC11 - Obtener Consumos

[3a 3] Se une al escenario principal en el paso [6]

Excepciones

[3b] Ingreso de rango de fechas incoherente:

[3b 1] El Usuario ingresa una fecha inicio mayor a fecha fin.

[3b 2] El Sistema muestra un mensaje de error.

UC12 - Enviar Consumos

Escenario principal

[1] El Proceso en segundo plano de Windows lanza la función de envío de consumos.

[2] <include> UC11 - Obtener Consumos.

[3] El Sistema filtra los consumos obteniendo solamente los no enviados con anterioridad.

[4] El Sistema envía los consumos a la API de DataDonors.

[5] El Sistema actualiza los consumos enviados

[6] El Sistema devuelve el resultado de operación al Proceso en segundo plano de Windows.

Caminos Alternativos:

[3a] No hay consumos para reportar:

[3a 1] El sistema no encuentra nuevos consumos para enviar.

[3a 2] Se une al escenario principal en el paso [6]

Excepciones

[*] Se cancela la operación:

[*1] El Usuario cancela la operación.

[*2] No se persisten los cambios.

[4a] Error en el servidor de la API de DataDonors:

[4a 1] El servidor de la API de DataDonors devuelve un código de error.

[4a 2] Se une al escenario principal en el paso [6]

UC13 – Enviar Reporte

Escenario principal

[1] El Usuario selecciona la opción Enviar Reporte.

[2] El Sistema solicita el ingreso de:

-Fecha inicio

-Fecha fin

[3] El Usuario ingresa los datos solicitados.

[4] <include> UC11 - Obtener Consumos.

[5] El Sistema filtra los consumos realizados entre la fecha inicio y fecha fin.

[6] El Sistema muestra los canales para compartir la información (email, redes sociales).

Caminos Alternativos:

[3a] No se ingresa un valor de fecha inicio y/o fecha fin:

[3a 1] El Usuario no ingresa un valor en peso y/o altura.

[3a 2] <include> UC11 - Obtener Consumos

[3a 3] Se une al escenario principal en el paso [6]

Excepciones

[3b] Ingreso de rango de fechas incoherente:

[3b 1] El Usuario ingresa una fecha inicio mayor a fecha fin.

[3b 2] El Sistema muestra un mensaje de error.

4. Capítulo IV - Solución propuesta

4.1. Modelo de objetos

4.1.1. Arquitectura de Capas de la Aplicación

En la figura se pueden observar las capas en las que está dividida la aplicación.

Cada capa está representada por un color diferente:

- Model en azul.
- ViewModel en amarillo.
- View en naranja.
- Utilidades en verde.

Cada cuadro representa un NameSpace dentro del programa, y el número la cantidad de objetos (clases, interfaces, etc.) en cada uno.

El límite del Sistema está representado por el recuadro de color azul. Por fuera del mismo se encuentra la API de DataDonors, con la que interactúa el Sistema.

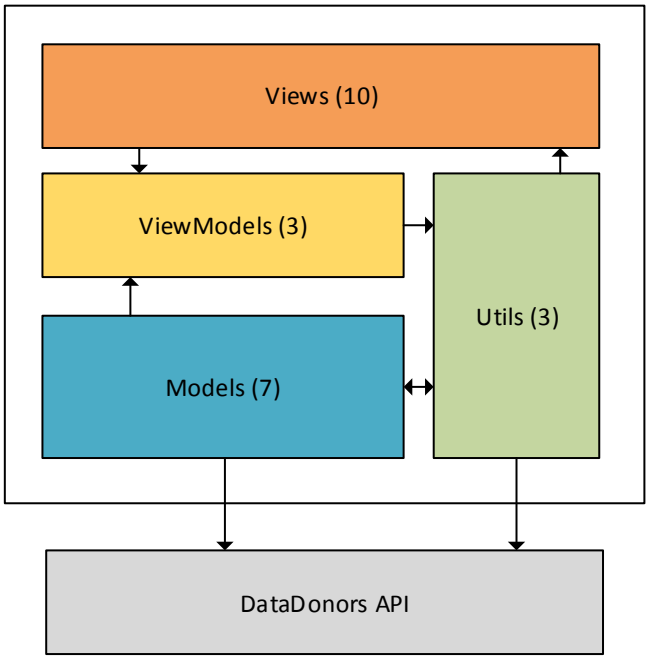

Figura IV-1: gráfico de arquitectura de las capas de la aplicación

4.1.2. Diagramas de Clases por Capa

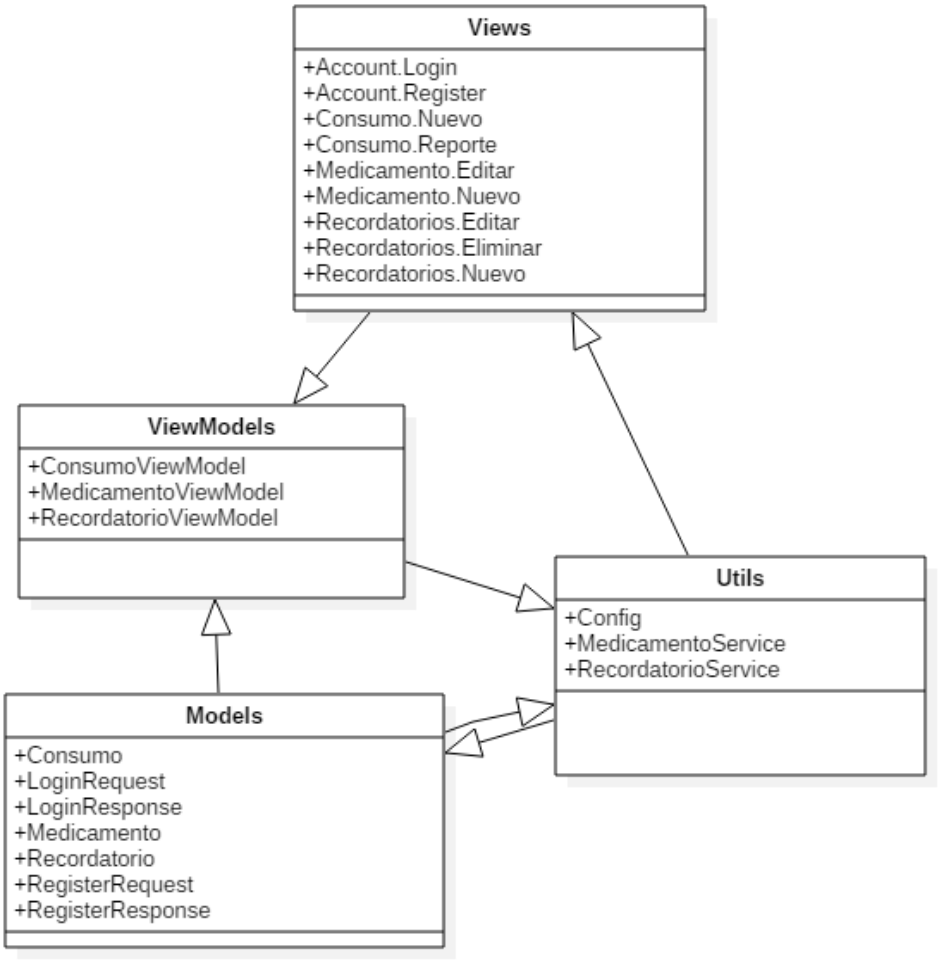

Figura IV-2: diagrama de clases por capa

4.1.3. Business Entities

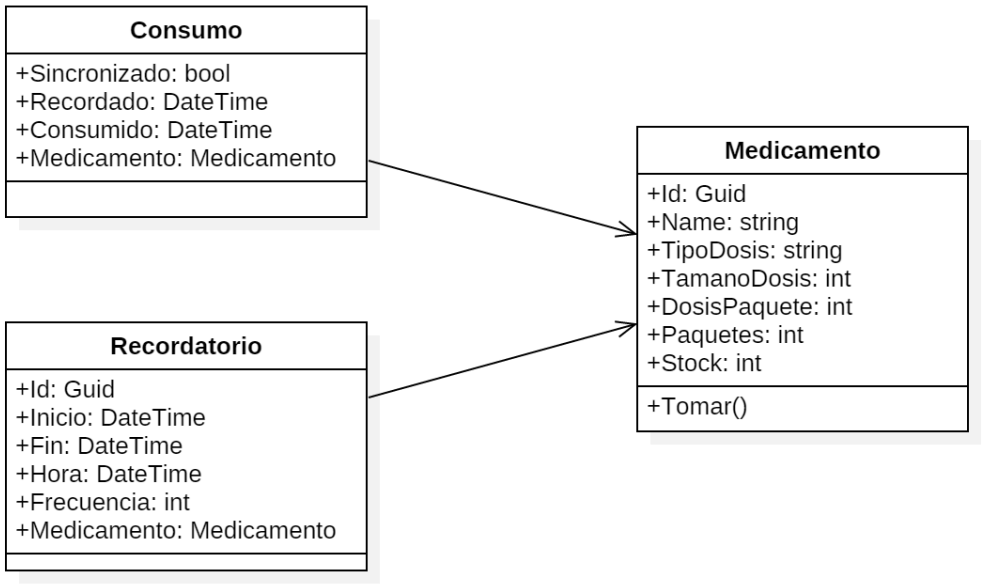

Figura IV-3: diagrama de clases de la capa “entidades de negocio”

4.1.4. Views

Figura IV-4: diagrama de clases de la capa “vistas”

4.1.5. ViewModels

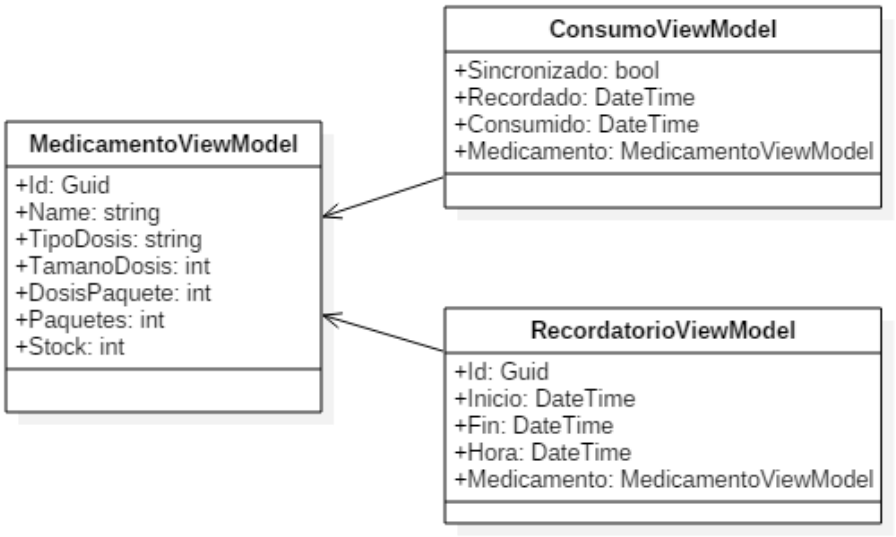

Figura IV-5: diagrama de clases de la capa “modelos de vista”

4.2. Modelo de interfaz de usuario

A continuación, se muestra el desarrollo de los prototipos de Interfaz de Usuario de la aplicación.

Menú de opciones (usuario no identificado)

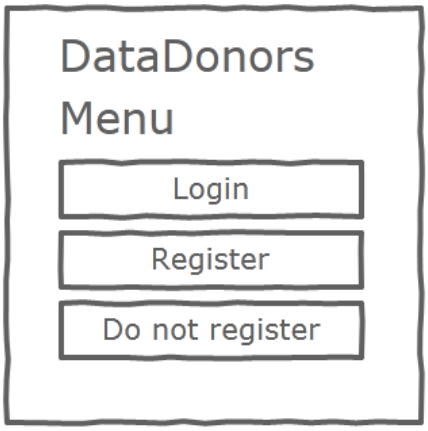

Figura IV-6: boceto de vista de menú de opciones con el usuario no identificado

Menú de opciones (usuario no identificado)

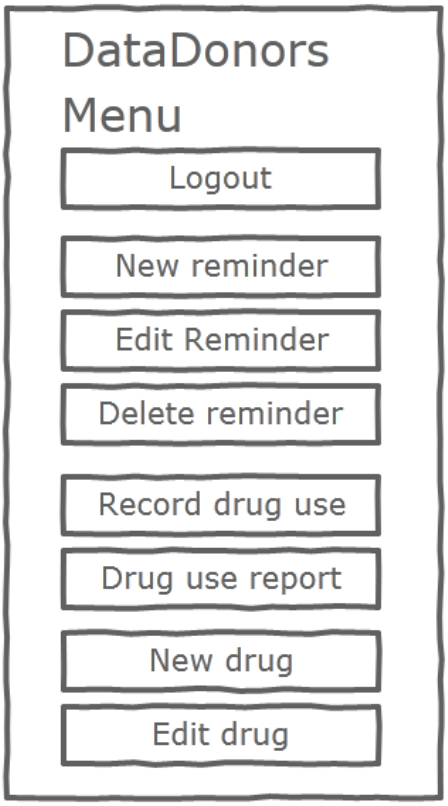

Figura IV-7: boceto de vista de menú de opciones con el usuario identificado

Registro de usuario

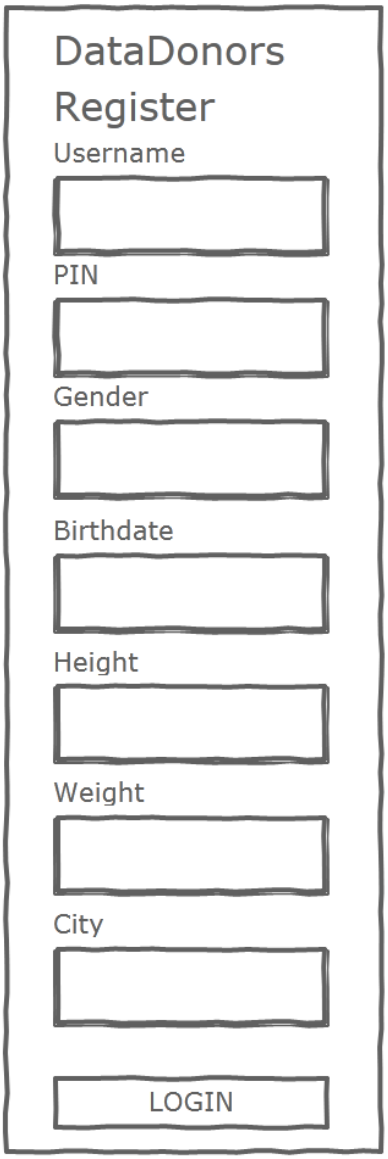

Figura IV-8: boceto de vista de registro de usuario

Identificación del usuario

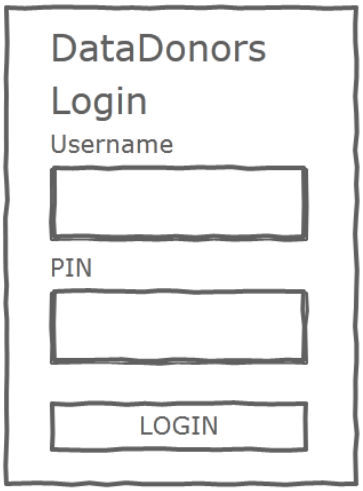

Figura IV-9: boceto de vista de identificación del usuario

Alta de medicamento

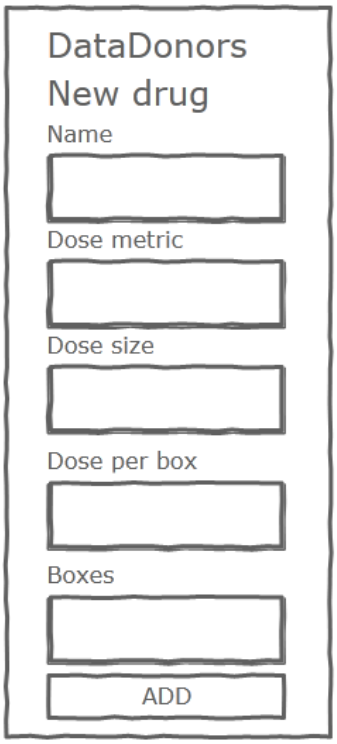

Figura IV-10: boceto de vista de alta de nueva droga

Editar medicamento

Figura IV-11: boceto de vista de edición de drogas

Nuevo recordatorio

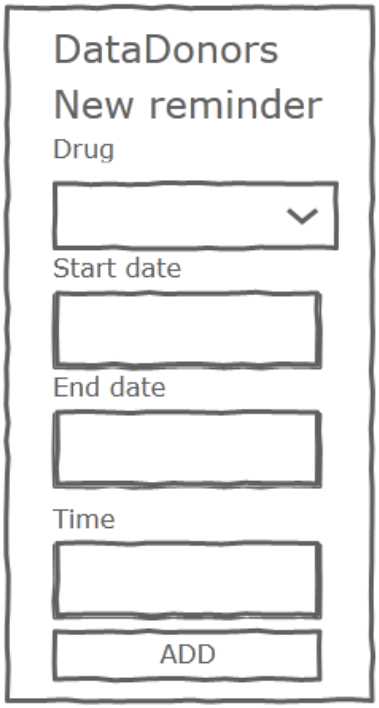

Figura IV-12: boceto de vista de nuevo recordatorio

Editar recordatorio

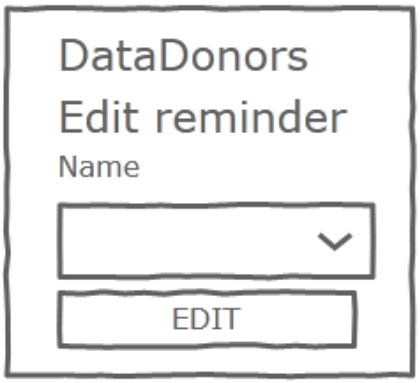

Figura IV-13: boceto de vista de edición de recordatorio

4.3. Modelo físico de los datos

Para guardar la información producida por la aplicación se decidió utilizar la serialización de los objetos.

El archivo con los datos serializados se guarda en la memoria interna del teléfono. El tamaño máximo que permite almacenar depende del modelo de teléfono. El. La clase utilizada para acceder al almacenamiento local es “IsolatedStorageSettings”.

Para evitar la falta de memoria, ese archivo estará sincronizado con el almacenamiento en la nube que el usuario tiene configurado en el teléfono. Esto, además, permitirá hacer backup automático de la información del usuario.

Como una tarea a futuro, se podría implementar el guardado de los objetos de negocio en una base de datos local utilizando SQL Lite.

5. Capítulo V - Producto obtenido

En este capítulo se muestra el producto en una primera iteración de desarrollo, con todos los requerimientos funcionales descriptos en este trabajo terminados.

Las funciones de registro de usuario, identificación, obtener base de medicamentos, métricas, y reporte de consumo están conectados a la API de DataDonors. [ver Anexo C]

5.1. Capturas de pantalla

A continuación, se muestran las capturas de pantalla de la aplicación obtenida corriendo sobre un emulador de Windows Phone 8.

Menú de opciones (usuario no identificado)

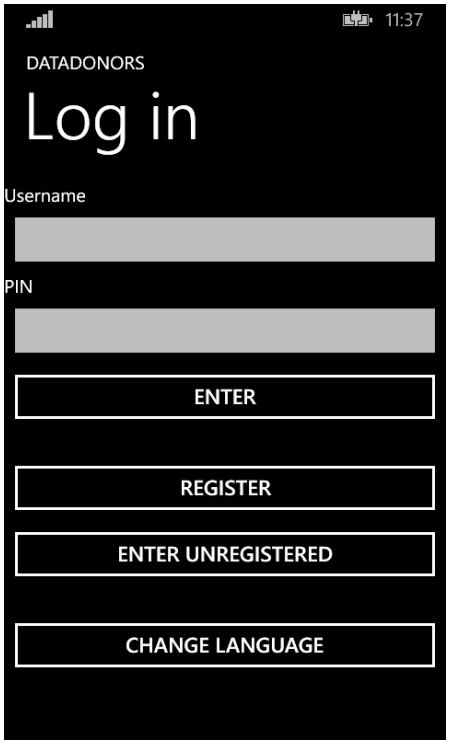

Figura V-1: menú de opciones con usuario no identificado

Menú de opciones (usuario identificado)

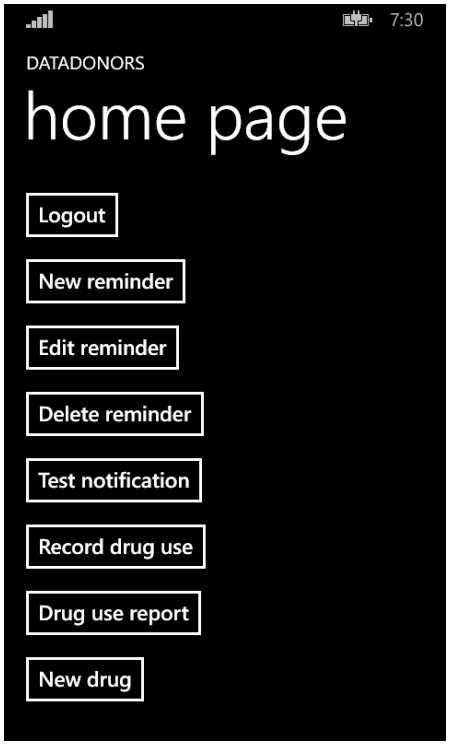

Figura V-2: menú de opciones con usuario identificado

Registro de usuario

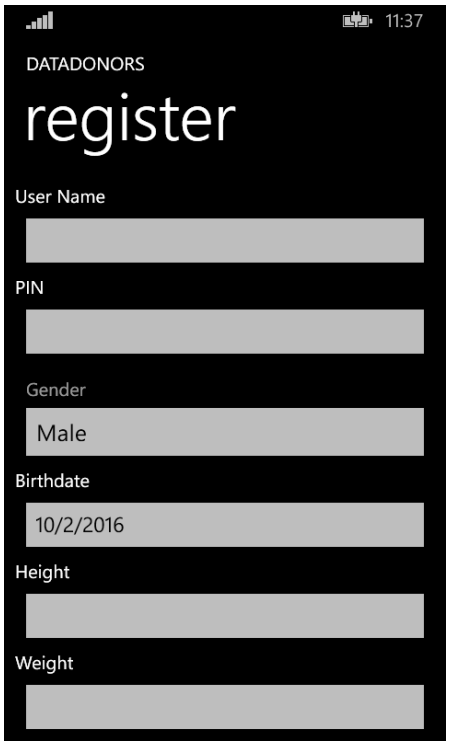

Figura V-3: registro de nuevo usuario

Identificación del usuario

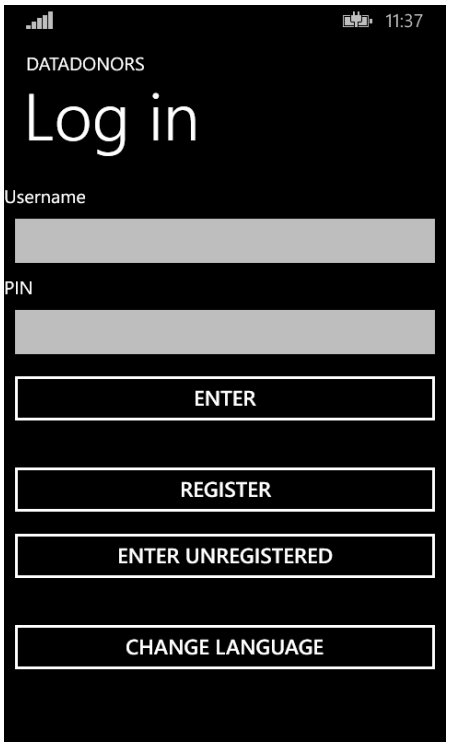

Figura V-4: identificación del usuario

Alta de medicamento

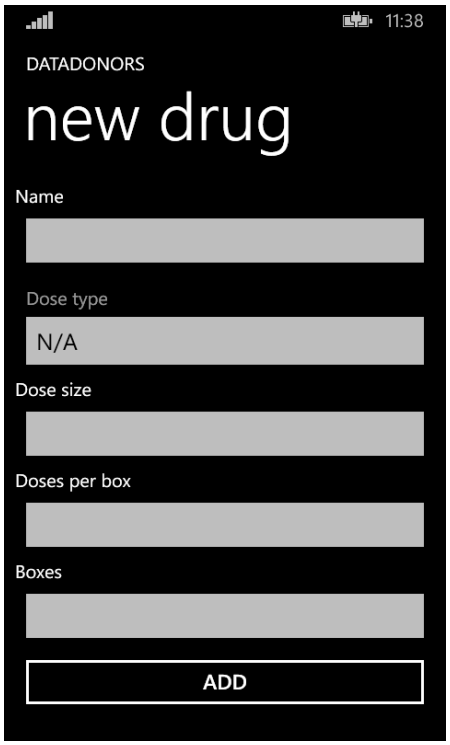

Figura V-5: alta de nuevo medicamento

Editar medicamento

Figura V-6: editar un medicamento existente

Nuevo recordatorio

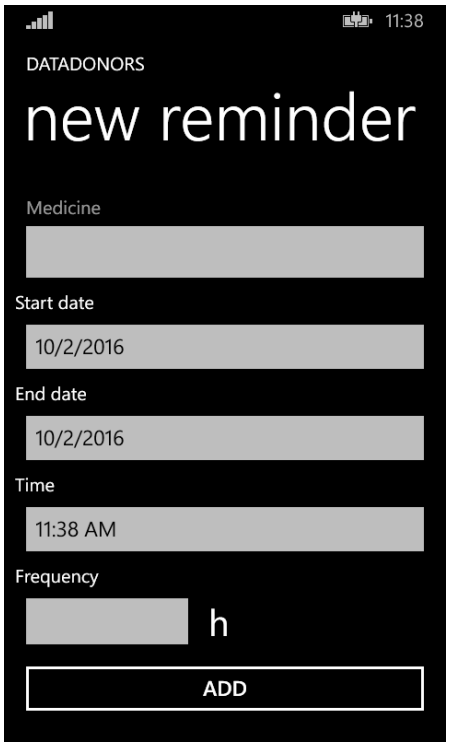

Figura V-7: alta de nuevo recordatorio

Editar recordatorio

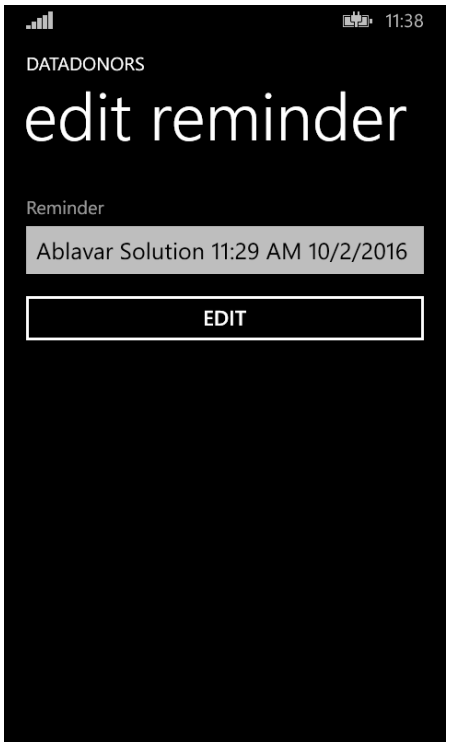

Figura V-8: editar un recordatorio existente

Notificación recordando consumo

Figura V-9: notificación recordando el consumo de medicamento

5.2. Pruebas

Se implementaron pruebas a componentes de la aplicación. La integración con el almacenamiento del teléfono, e integración con la API de DataDonors.

Se desarrolló el componente de almacenamiento como una interfaz. Y su implementación actual almacena la información serializada como JSON en la memoria interna del teléfono. La interfaz permite que fácilmente la implementación sea modificada por almacenamiento en base de datos.

Además, las interfaces permiten que las implementaciones puedan ser probadas con pruebas unitarias.

Dentro de la solución, se creó un proyecto de pruebas que tiene referencia a NUnit, componente con el que se ejecutan las pruebas en .NET; y clases que implementan las pruebas. Esto permite automatizar la ejecución de las pruebas en cada compilación del proyecto. Que asegura mantener el mismo comportamiento, sin errores, en futuros incrementos.

Se realizan pruebas de caja blanca en el caso de las pruebas a las interfaces de almacenamiento, y pruebas de caja negra, en el de las pruebas a la API de DataDonors.

La clase APITests, contiene 9 métodos (pruebas) que verifican las llamadas a la API de DataDonors, según las especificaciones brindadas para su integración.

A continuación, se detallan esas pruebas, resultado esperado:

Nombre: **WhenRegistroThenHttpStatusOK** **Resultado esperado**: Realiza un llamado a la API de DataDonors para registrar un usuario. Los datos enviados son correctos, se espera como resultado un código HTTP 200.
Nombre: **WhenRegistroUsuarioExistenteThenHttpStatusConflict** **Resultado esperado**: Realiza un llamado a la API de DataDonors para registrar un usuario. El nombre de usuario enviado ya existe en DataDonors, se espera como resultado un código HTTP 409 (conflicto).
Nombre: **WhenRegistroUsuarioVacioThenHttpStatusBadRequest** **Resultado esperado**: Realiza un llamado a la API de DataDonors para registrar un usuario. El nombre de usuario se envía vacío, se espera como resultado un código HTTP 400 (request invalido).
Nombre: **WhenLoginUsuarioThenHttpStatusOKAndReturnUserTocken** **Resultado esperado**: Realiza un llamado a la API de DataDonors para autenticar un usuario. La información

enviada es válida, se espera como resultado un token que identifica al usuario, que será utilizado en cada request siguiente que requiera autenticación del usuario.
Nombre: **WhenLoginUsuarioVacioThenHttpStatusBadRequest** **Resultado esperado**: Realiza un llamado a la API de DataDonors para autenticar al usuario. El nombre de usuario se envía vacío, se espera como resultado un código HTTP 400 (request invalido).
Nombre: **WhenLoginUsuarioMalInfoThenHttpStatusNotAuthorized** **Resultado esperado**: Realiza un llamado a la API de DataDonors para autenticar al usuario. El nombre de usuario se envía correcto, pero el password no corresponde a ese usuario, se espera como resultado un código HTTP 401 (no autorizado).
Nombre: **WhenGetDrugListThenReturnDrugs** **Resultado esperado**: Realiza un llamado a la API de DataDonors para solicitar la lista de drogas. Se espera como resultado la lista de drogas.
Nombre: **WhenGetDrugMetricListThenReturnDrugsMetrics** **Resultado esperado**: Realiza un llamado a la API de DataDonors para solicitar la lista de métricas de las drogas. Se espera como resultado la lista de métricas de las drogas.
Nombre: **WhenPostInfoThenReturnHttpStatusOK** **Resultado esperado**: Realiza un llamado a la API de DataDonors para enviar información de consumo del usuario. Se espera como resultado un código HTTP 200 (solicitud OK).

Figura V-10: métodos de la clase APITests y sus resultados esperados

En la siguiente figura se pueden observar los resultados de estas pruebas:

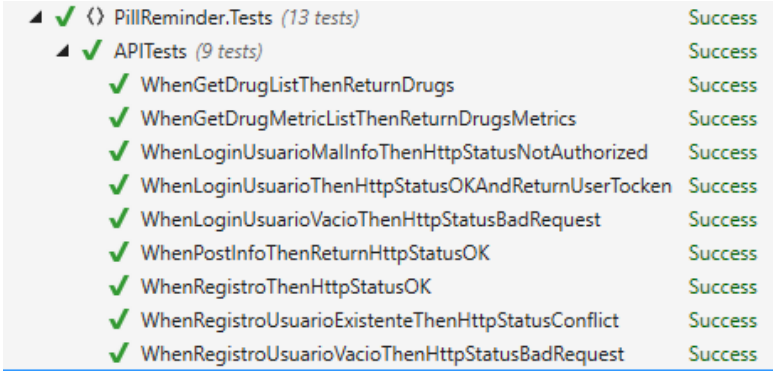

Figura V-11: resultado de ejecución de pruebas unitarias

6. Futuras líneas de desarrollo

Teniendo en cuenta lo desarrollado por DataDonors y el presente trabajo final, se evalúan posibles futuras líneas de desarrollo que se detallan a continuación, enfocado a la categoría "drogas" de DataDonors, que es la menos desarrollada:

1) Aplicación "Pill Reminder" (Recordatorio de toma de medicamentos) para otros sistemas operativos, Android, iOS y Blackberry.
   Poder tener disponible esta aplicación en todos los sistemas operativos va a incrementar el mercado de usuarios que utilicen la aplicación, por lo tanto, aumento de cantidad de datos generados para la categoría.

2) Aplicación conectada a Microsoft Health.
   Microsoft Health es la plataforma de salud de Microsoft. Los usuarios que utilizan cualquier aplicación relacionada a salud desarrollada por Microsoft envían la información a una base de datos centralizada que se puede consultar por APIs.
   La aplicación de DataDonors podría conectarse a la API de Microsoft Health y enviar esa información del usuario a la base de datos de DataDonors. Este mecanismo es utilizado actualmente por la aplicación de DataDonors para iOS, tomando la información de iHealth, sistema similar a Microsoft Health.

3) Aplicación conectada a Microsoft Band.
   Microsoft Band es el reloj inteligente de Microsoft, puede ser usado con teléfonos inteligentes con Android, iOS y Windows Phone. Tiene 11 sensores que permiten analizar:
   - Cantidad de pasos y distancia recorrida
   - Análisis de sueño
   - Calorías quemadas
   - Monitoreo de ritmo cardiaco
   - Temperatura corporal

La información del reloj es enviada a Microsoft Health, pero además puede ser leída por aplicaciones de los sistemas operativos que soporta.

Se puede desarrollar aplicaciones en los tres sistemas operativos que tomen información del reloj y la envíe a la base de DataDonors.

4) Aplicaciones de otras categorías para Windows Phone, iOS y Android.

   Desarrollar aplicaciones específicas de otros verticales además de la categoría "drogas".

7. Conclusiones

Proyectos como DataDonors son de gran importancia debido a que ofrecen a la comunidad científica información valiosa para la investigación en diferentes áreas. Pero, sin datos, DataDonors no va a poder ofrecer ese beneficio. Por lo que son necesarias aplicaciones como la que se desarrolló para este proyecto final, o integraciones con otros sistemas, como las que se plantean en la sección "futuras líneas de desarrollo".

La arquitectura planteada al inicio del proyecto involucraba un Modelo con clases de negocio dependientes de la implementación de la API de DataDonors. Al haber utilizado un proceso de desarrollo iterativo e incremental se pudo detectar este problema en una etapa temprana, evitando así una pérdida mayor de tiempo y recursos.

La arquitectura final de la aplicación tiene la ventaja de tener una separación en capas bien definidas y con bajo acoplamiento, lo que permite, entre otras cosas:

- Adaptar la aplicación a las nuevas versiones de la API de DataDonors.
- Implementar otro sistema de persistencia de los datos.
- Realizar modificaciones, de ser necesario, de una manera más sencilla y menos costosa.
- Utilizar la capa de entidades de negocio (Business Entities) para desarrollar la misma aplicación para otros sistemas operativos.

La implementación de casos de prueba unitarios detectó inconsistencias en los códigos HTTP de respuesta de la API de DataDonors que fueron corregidos por el equipo de desarrollo de Wikilife. Además, que estos casos de prueba pueden ser usados independientemente de la aplicación que se esté desarrollando.

8. Bibliografía

Apple, Dev Programs, (en línea), Estados Unidos: 2014 <https://developer.apple.com/programs/> [Consulta: 04/05/2014]

Apple, Store de aplicaciones de Apple "AppStore", (en línea), Estados Unidos: 2014 <http://www.appstore.com> [Consulta: 06/05/2014]

Apple, Introduction to iOS development with XCode, (en línea), Estados Unidos <https://developer.apple.com/xcode/> [Consulta: 04/05/2014]

Benslimane, Djamal; Schahram Dustdar; Amit Shet: Services Mashups: The New Generation of Web Applications. IEEE Internet Computing, vol. 12, no. 5. Institute of Electrical and Electronics Engineers. pp. 13–15. (2008)

Bidulka, Brian; BlackBerry 2013 Annual Report (en línea), Canada, <http://press.blackberry.com/content/dam/rim/press/PDF/Financial/FY2013/Q4FY13_final_filing.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; BlackBerry 2014 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2014/Q4_FY14_Filing.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; BlackBerry 2015 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2015/Q4_Fiscal_2015/Q4%20FY15%20filing.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2003 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2003/2003rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2004 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2004/2004rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2005 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2005/2005rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2006 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2006/2006rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2007 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2007/2007rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2008 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2008/2008rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2009 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2009/2009rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2010 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2010/2010rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2011 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2011/2011rim_ar.pdf> [Consulta, 23 ago. 2015]

Bidulka, Brian; Research in Motion, 2012 Annual Report (en línea), Canada, <http://global.blackberry.com/content/dam/bbCompany/Desktop/Global/PDF/Investors/Documents/2012/2012rim_ar_40F.pdf> [Consulta, 23 ago. 2015]

Blackberry, Acerca de Blackberry, (en línea), Estados Unidos: 2014 <http://ar.blackberry.com/company/company.html> [Consulta: 5 may. 2014]

Blackberry, Recursos para desarrolladores, (en línea), Estados Unidos <http://global.blackberry.com/es/sites/developers/resources.html?LID=ar:bb:apps:blackberryappworld&LPOS=ar:bb:apps> [Consulta: 5 may. 2014]

Boehm, Barry; A Spiral Model of Software Development. IEEE Computer, May 1988.

Britch, David; Cheung, Francis; Kinney, Adam; Sharma, Rohit; Microsoft Patters and Practices, Developing an Advanced Windows Phone 7.5 App that Connects to the Cloud, Redmond, Estados Unidos, 2012. ISBN 978-1-62114-015-3.

Danova, Tony; Android Continues To Crowd Apple Out Of The Global Smartphone Market (en línea), Australia, 9 ago 2013 <http://www.businessinsider.com.au/android-grabs-80-of-smartphone-market-2013-8> [Consulta: 4 may. 2014]

Elgin, Ben; Bloomberg: Google Buys Android for Its Mobile Arsenal (en línea), Estados Unidos: 17 ago. 2005 <http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm> [Consulta: 15 may. 2016]

Dowling, Steve; Barney, Amy; Apple, iPhone Premieres This Friday Night at Apple Retail Stores, (en línea), Estados Unidos, Palo Alto: jun 2007 <http://www.apple.com/pr/library/2007/06/28iPhone-Premieres-This-Friday-Night-at-Apple-Retail-Stores.html> [Consulta: 04/05/2014]

Fortin, Michael; Upgrading existing Windows Phone 8.1 devices to Windows 10 Mobile (en línea), Estados Unidos: mar. 2016 <http://blogs.windows.com/windowsexperience/2016/03/17/upgrading-existing-windows-phone-8-1-devices-to-windows-10-mobile/> [Consulta: 15/05/2016]

Google Inc., Introduction to Android Development, (en línea), Estados Unidos <http://developer.android.com/develop/index.html> [Consulta: 04/05/2014]

Google Inc., Android Compatibility Definition, (en línea), Estados Unidos: 16 oct. 2015 <http://static.googleusercontent.com/media/source.android.com/en//compatibility/android-cdd.pdf > [Consulta: 15/05/2016]

Google Inc., Store de aplicaciones de Google Android "Play Store", (en línea), Estados Unidos: 2014 <https://play.google.com/> [Consulta: 06/05/2014]

Llamas, Ramon; Reith, Ryan; Shirer, Michael; IDC, Apple Cedes Market Share in Smartphone Operating System Market as Android Surges and Windows Phone Gains, According to IDC (en línea), Estados Unidos: 7 ago. 2013 <http://www.idc.com/getdoc.jsp?containerId=prUS24257413> [Consulta: 5 may. 2014]

Marqués, Asier: Conceptos sobre APIs REST. (en línea), España: abr. 2013 <http://asiermarques.com/2013/conceptos-sobre-apis-rest/> [Consulta: 21 sep. 2015]

Microsoft MSDN; Información de la plataforma de Desarrollo Windows Phone (en línea), Estados Unidos, 2015 < https://msdn.microsoft.com/es-ar/library/windows/apps/br211361.aspx > [Consulta: 23 ago. 2015]

Microsoft Corporation, Store de aplicaciones de Microsoft "Microsoft Store", (en línea), Estados Unidos: 2014 <https://www.microsoft.com/en-us/store/apps/windows-phone> [Consulta: 06/05/2014]

Nofal, Daniel; Gueron, Geraldine; The Wikilife Foundation, Data sources (en línea) <http://datadonors.org/learn-more/> [Consulta: 04/10/2014]

Nofal, Daniel; Gueron, Geraldine; The Wikilife Foundation, Misión (en línea), Argentina, 2014 <http://datadonors.org/mission/> [Consulta: 04/10/2014]

Nofal, Daniel; Gueron, Geraldine; The WikiLife Foundation; Wikilife, Apple Store (en línea), <https://itunes.apple.com/en/app/id443072007> [Consulta: 04/10/2014]

Petzold, Charles; Programming Windows Phone 7. 1a ed. Redmond, Washington: Microsoft Press, 2010. ISBN 978-0-7356-4335-2.

Rivera, Janessa; Goasduff, Laurence; Gartner, Gartner Says Smartphone Sales Surpassed One Billion Units in 2014 (en línea), Egham, Reino Unido, 3 mar 2015 <http://www.gartner.com/newsroom/id/2996817 > [Consulta: 6 sep. 2015]

Rivera, Janessa; van der Meulen, Rob; Gartner, Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013 (en línea), Egham, Reino Unido, 13 feb 2014 <http://www.gartner.com/newsroom/id/2665715> [Consulta: 4 may. 2014]

Shirer, Michael; Llamas, Ramon; Restivo, Kevin; IDC, Worldwide Mobile Phone Market Forecast to Grow 7.3% in 2013 Driven by 1 Billion Smartphone Shipments, According to IDC (en línea), Estados Unidos: 4 sep. 2013 <http://www.idc.com/getdoc.jsp?containerId=prUS24302813> [Consulta: 5 may. 2014]

Statista; Number of available apps in the Apple App Store from July 2008 to June 2015 (en línea), Estados Unidos: 2016 <http://www.statista.com/statistics/263795/number-of-available-apps-in-the-apple-app-store/> [Consulta: 15 may. 2016]

Wastell, Blaine; Cheung, Francis, Delgado, Nelly; Sharma, Rohit; Corbisier, RoAnn; Microsoft, Developer's Guide to Microsoft Prism Library 5.0 for WPF (en línea), Estados Unidos: abr, 2014 <http://msdn.microsoft.com/en-us/library/gg405484(v=PandP.40).aspx> [Consulta: 6 may. 2014]

Wildermuth, Shawn; Microsoft, Model-View-ViewModel In Silverlight 2 Apps (en línea), Estados Unidos: mar. 2009 <http://msdn.microsoft.com/es-es/magazine/dd458800.aspx> [Consulta: 6 may. 2014]

Zinser, Billy; XCubeLabs, The Android Story (en línea), Estados Unidos: oct. 2013 <http://www.xcubelabs.com/the-android-story.php> [Consulta: 04/05/2014]

Anexo

ANEXO A: Aplicaciones existentes

MedCoach Medication Reminder (iOS)

En la plataforma iOS hay gran cantidad de aplicaciones de salud (2% del total, unas 18.000), pero la más alineada a los requerimientos de este trabajo final es MedCoach Medication Reminder. (Statista, 2016)

El usuario que utiliza esta aplicación puede crear recordatorios para la toma de un medicamento, tanto para horarios establecidos, como también por un período determinado.

Además, cuenta con la posibilidad de ingresar la cantidad de dosis que trae el medicamento, para llevar registrada la cantidad de stock que se dispone, y de esa forma recordar al usuario cuándo se acerca el momento de reponer el medicamento.

Cuando una alerta se dispara, informa al usuario que es momento de consumir una dosis de ese medicamento, y brinda la posibilidad de marcar si esa dosis fue consumida efectivamente. Con esta información se almacena un historial de consumo que se muestra al usuario.

La aplicación también permite almacenar información de contacto de la cartilla de médicos del usuario, tanto por especialidad como por cercanía al domicilio del paciente, y las farmacias más cercanas para poder comprar el medicamento consumido.

La interfaz de usuario es muy sencilla, y además al momento de dar de alta una alarma, mientras el usuario escribe el nombre del medicamento, el programa va recomendando nombres que coinciden, ya que la aplicación tiene una base de datos de medicamentos.

Como contrapartida se puede destacar la falta de imágenes de los productos, que pueden ayudar al usuario a identificarlos, además, solo se puede utilizar en inglés, y los medicamentos que recomienda son del mercado estadounidense.

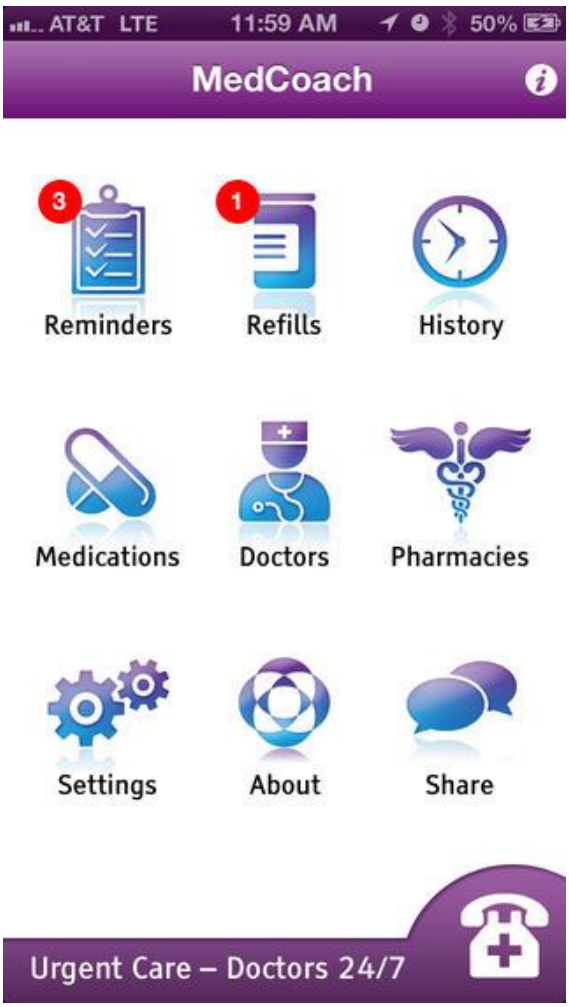

Figura AA-1: Captura de pantalla MedCoach.

Pill Reminder by Drugs.com (iOS)

Dada la cantidad de aplicaciones que dispone la plataforma iOS (aproximadamente 900.000), es que se analizaron dos alternativas: Pill Reminder by Drugs.com, y MedCoach Medication Reminder, seleccionadas por la cantidad de descargas y comentarios positivos (Statista, 2016).

Pill Reminder by Drugs.com brinda la posibilidad de generar alertas para recordar la toma de una dosis de medicamento, como también alertas tempranas ante la necesidad de compra de más dosis. También recomienda las drogas desde una base de datos en la pantalla de ingreso de alertas, muestra información muy detallada del medicamento y sus tipos de dosis, como por ejemplo contraindicaciones.

Figura AA-2: Captura de pantalla Pill Reminder By Drugs.com.

AnyTimer Pill Reminder (Android)

En la plataforma Android se analizó la aplicación AnyTimer Pill Reminder, que respecto a aplicaciones en iOS tiene acotada funcionalidad, pero es la más completa dentro de la categoría, aunque no parece una aplicación orientada a medicina.

Tiene un diseño básico que lo hace fácil de usar. Su facilidad se centra en que no tiene gran cantidad de funcionalidad, pero sí lo necesario para recordar el momento en que se debe tomar una dosis de medicamento. Permite el ingreso de un recordatorio tanto por horario fijo como por intervalos de tiempo fijo.

A diferencia de MedCoach Medication Reminder para iOS, no cuenta con una base de medicamentos para recomendar al usuario al momento de ingresar el nombre y generar el alerta.

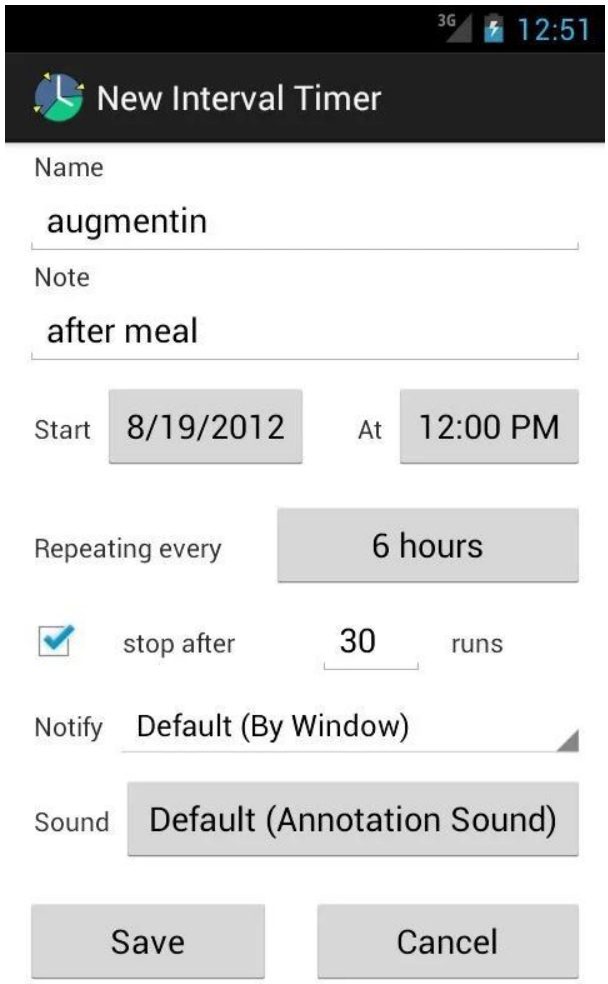

Figura AA-3: Captura de pantalla AnyTimer Pill Reminder.

Pill Reminder (Windows Phone)

En la plataforma Windows Phone existe solo una aplicación con las características similares a las que se plantea desarrollar, Pill Reminder.

Permite el ingreso de alarmas en tiempos determinados, a diferencia de las otras aplicaciones esta no tiene una lista de todas las alertas configuradas, tampoco lista las alertas en un calendario, tiene alto consumo de batería, carece de notificaciones y de integración con el Sistema Operativo.

Aunque la aplicación promete recordarte el momento en que se debe tomar una dosis de medicamento, solo utiliza un cronometro en cuenta regresiva para alertar cuando el tiempo finalizo.

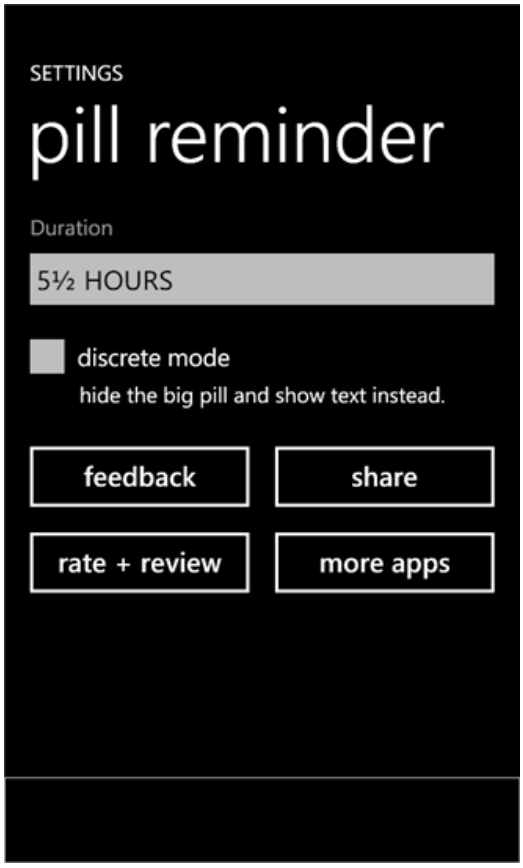

Figura AA-4: Captura de pantalla Pill Reminder (Windows Phone).

Cuadro comparativo de aplicaciones

Figura I-4: generaciones de equipos, requerimientos, versión del Sistema Operativo

	MedCoach Medication Reminder (iOS)	Pill Reminder by Drugs.com (iOS)	AnyTimer Pill Reminder (Android)	Pill Reminder (Windows Phone)
Alerta de toma de medicamento	SI	SI	SI	SI
Por intervalos	SI	SI	SI	NO
Por tiempo fijo	SI	SI	SI	SI
Lista de alertas	SI	SI	SI	NO
Recarga de dosis	SI	SI	NO	NO
Base de drogas	SI	SI	NO	NO
Información de medicamentos	SI	NO	NO	NO
Notificaciones	SI	SI	SI	SI
Otras	Doctores y farmacias	-	-	-

ANEXO B: Plataformas Móviles

Android

Android es un Sistema Operativo diseñado para dispositivos móviles basado en Linux. En sus comienzos fue desarrollado por Android Inc., con ayuda económica de parte de Google, hasta que fue comprada en 2005 (Zinser, 2013; Elgin, 2005).

El día de su anuncio, Google, junto con otras 77 empresas dedicadas a la construcción de hardware, desarrollo de software y telecomunicaciones conformaron un consorcio para generar estándares abiertos para dispositivos móviles, el Open Handset Alliance (Zinser, 2013).

Es un Sistema Operativo de código abierto desarrollado por la comunidad. Google liberó gran parte del código de Android bajo licencia Apache (libre y de código abierto). Cuenta con 12 millones de líneas de código distribuidos en XML, C, C++ y Java. La comunidad además extiende su funcionalidad a través de aplicaciones, que al momento alcanzan las 975000 publicadas en la tienda de aplicaciones, llamada Google Play (Google Inc., 2014).

La plataforma es lo suficientemente flexible para ser utilizada tanto en teléfonos inteligentes, como también en computadoras personales, laptops, tabletas, relojes pulsera, auriculares, sistemas de televisión, electrodomésticos y anteojos inteligentes. Android es utilizado actualmente en 900 millones de dispositivos en 190 países, alcanzando casi el 80% del mercado total de dispositivos móviles. (Google Inc., 2014; Llamas, 2013; Shirer, 2013).

A diferencia de iOS y Windows Phone, Android es desarrollado de forma abierta y se puede acceder fácilmente al código fuente, como también a un sistema de control de incidencias, que la comunidad utiliza para reportar y solucionar.

El disponer de la última versión de Android no garantiza su instalación en cualquier dispositivo. Para poder instalarlo es necesario disponer de los controladores de hardware de cada fabricante que no son públicos. Además, desde la primera versión, el sistema operativo fue incrementando sus requerimientos de hardware (memoria RAM y procesador), haciendo imposible la instalación de nuevas versiones en dispositivos antiguos. (Google Inc., 2015)

La primera versión beta fue lanzada en noviembre de 2007 y la primera versión comercial, Android 1.0, en septiembre de 2008. Luego que comenzó a desarrollarse por Google, en vinculación con el Open Handset Alliance, comenzaron a lanzarse actualizaciones con nombres propios (codenames), ordenados alfabéticamente; Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3), y KitKat (4.4), Lollipop (5.0-5.1), Marshmallow (6.0).

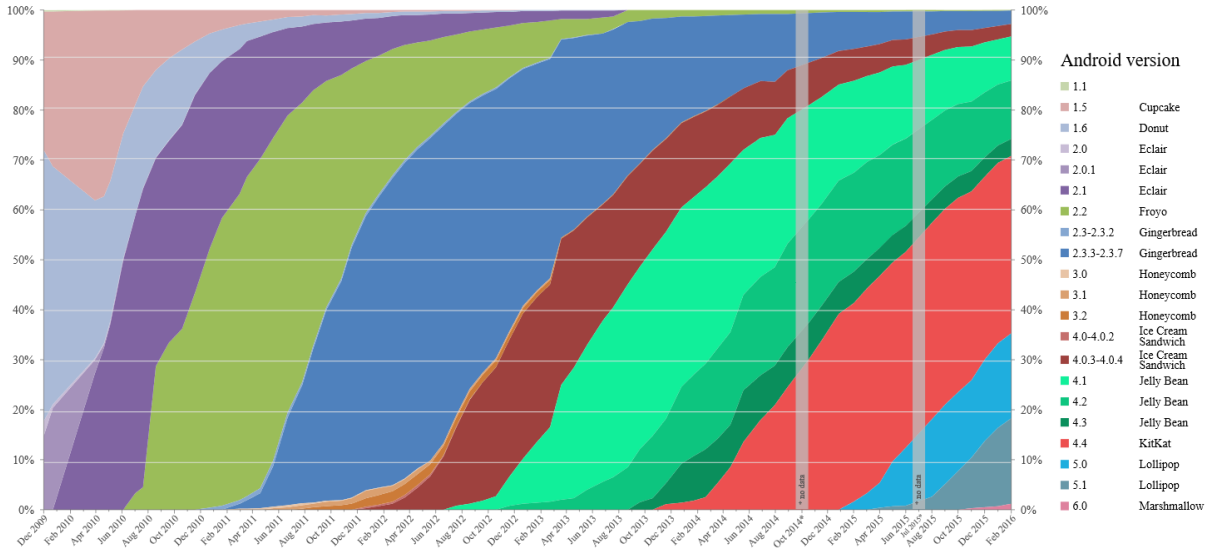

Figura AB-1: distribuciones de Android.

Las aplicaciones son desarrolladas en lenguaje Java junto con Android Software Development Kit (Android SDK). Adicionalmente, existen otras herramientas de desarrollo. La plataforma también permite el desarrollo de extensiones en lenguajes C y C++ (Zinser, 2013).

Cuando una aplicación ya está desarrollada, el siguiente paso es distribuirla a través de la tienda. Un desarrollador debe registrarse en Google Play pagando una suscripción

anual de USD 25 (dólares americanos), que le da acceso no solo a la publicación de sus aplicaciones en la tienda, sino también a un detallado portal para conocer tendencias de descargas y uso de las aplicaciones (Google Inc., 2014).

iOS

Apple Inc. es fabricante de dispositivos móviles como iPhone, iPod, iPad; además desarrolló su propio Sistema Operativo propietario basado en Unix.

El primer iPhone, lanzado el 17 de junio de 2007, incluyó como Sistema Operativo iPhone OS desarrollado especialmente para este dispositivo. Posteriormente se lanzaron productos como el iPod e iPad que también incluyeron el mismo Sistema Operativo, que paso a llamarse iOS (Dowling, 2007).

Con el lanzamiento del kit de desarrollo (SDK) se abrió la posibilidad a desarrolladores de acceder a la API del Sistema Operativo para poder realizar aplicaciones que interactúen con la tecnología de cada dispositivo.

Para el desarrollo de aplicaciones es necesario obtener Xcode (IDE, solo corre en plataforma Mac) y el kit de desarrollo (iOS SDK). El kit de desarrollo incluye un emulador de dispositivo, para poder probar la aplicación sin necesidad de un equipo físico. La descarga de Xcode y la creación de una aplicación no tienen costo para el desarrollador. Para distribuirlas, en cambio, es necesaria la compra de una suscripción de desarrollo que tiene un costo aproximado de USD 99 (dólares americanos). Esta suscripción permite obtener ganancia sobre las aplicaciones, con posibilidad de venta, publicidad sobre la aplicación, y compras dentro de la misma. Toda ganancia es repartida con un 70% para el desarrollador, y un 30% en concepto de comisión para Apple. (Apple, 2014)

Xcode permite el desarrollo de aplicaciones en lenguajes C, C++, Objective C, Objective C++, Java, AppleScript, Python y Ruby. Adicionalmente existe compatibilidad con otros lenguajes (Apple, 2014).

Para distribuir aplicaciones, Apple dispone de una tienda de aplicaciones (App Store) que contiene actualmente más de 900.000 disponibles. Desde su lanzamiento en 2008 se realizaron más de 50.000.000.000 de descargas, alcanzando un promedio de 50.000 descargas por aplicación, lo que hace a la plataforma una gran oportunidad para el desarrollo (Apple, 2014).

BlackBerry

BlackBerry es una compañía canadiense (antes Research In Motion o RIM) que fabrica teléfonos inteligentes (SmartPhones), integrando un servicio de mensajería propio, conocido como BlackBerry Messenger, además de aplicaciones típicas de un teléfono inteligente.

La compañía se destacó por la alta penetración de sus teléfonos en mercados corporativos, ofreciendo conectividad constante con los servicios de email, mensajería, e internet, y encriptación de la información por hardware. Además, incorporaban un teclado QWERTY para facilitar la escritura, aunque nuevos modelos incluyen pantalla táctil, eliminando el teclado físico.

BlackBerry no solo fabrica el hardware, sino también desarrolla su Sistema Operativo. Los primeros modelos contaban con distintas versiones de BlackBerry OS. A principios de 2013, BlackBerry lanzó dos modelos nuevos de teléfonos (Q10 y Z10) con un nuevo Sistema Operativo, el BlackBerry 10 (Blackberry, 2014).

BlackBerry OS es un Sistema Operativo de licencia propietaria desarrollado por BlackBerry. Su primera versión fue lanzada en enero de 1999, y acompañando los lanzamientos de nuevos dispositivos, fue actualizándose.

Versión	Equipos soportados
4.5	BlackBerry series 8100 y 8800 BlackBerry Curve 8300/8310/8320
4.6	BlackBerry Pearl Flip.

5.0	BlackBerry Curve 8330/8350i/8520/8530/8900 BlackBerry Bold 9000 BlackBerry Storm 1 y 2 BlackBerry Tour
6.0	BlackBerry Bold 9650/9700/9780 BlackBerry Curve 9300/9330 BlackBerry Pearl 9100/9105 BlackBerry Style 9670 BlackBerry Torch 9800.
7.0	BlackBerry Bold 9790/9900/9930
7.1	BlackBerry Curve 9310/9315/9350/9360/9370/9380/9220/9320 BlackBerry Torch 9810/9850/9860.

Figura AB-2: versión de SO BlackBerry por equipo

Actualmente BlackBerry OS está discontinuado, para luego lanzar de BlackBerry 10. Este último trae un nuevo teclado inteligente, una aplicación que mejora el uso de la cámara, multitarea permitiendo tener hasta ocho aplicaciones ejecutándose simultáneamente, una nueva pantalla de inicio que permite el agregado de información, un nuevo HUB de notificaciones, que incluye información sobre mensajería, email, llamadas y actividad en redes sociales; además de mejoras en la usabilidad y principal foco en aprovechar las pantallas táctiles (Blackberry, 2014).

Tanto BlackBerry OS como BlackBerry 10 permiten a desarrolladores trabajar con sus respectivas interfaces (APIs), para ser utilizadas por aplicaciones. Se pueden desarrollar aplicaciones utilizando C/C++, Javascript/CSS/HTML, ActionScript o Java. Por la gran variedad de opciones es que existen muchas herramientas para el desarrollo, tanto Adobe Flex para compilar Adobe Air, un IDE como Eclipse o Visual Studio para C/C++, Javascript/HTML, como también es posible la re-compilación de aplicaciones Android para BlackBerry (Blackberry, 2014).

De un mercado de más de 1 billón de teléfonos inteligentes, BlackBerry domina solamente el 2,7%, y la tendencia es decreciente, estimándose una caída hasta llegar a 1,7% del mercado en 2017 (Llamas, 2013).

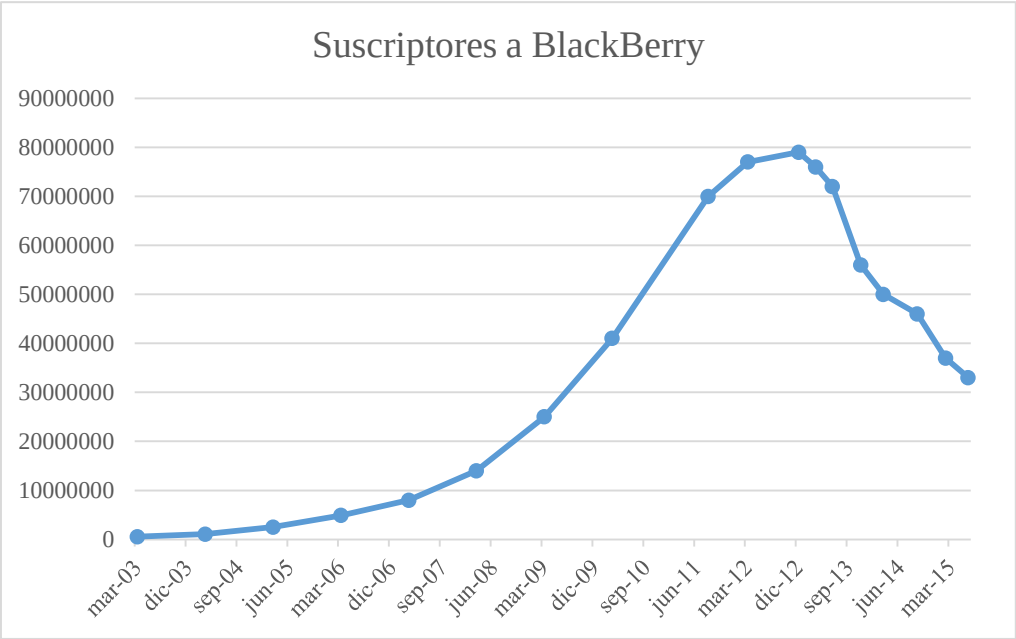

Figura AB-3: Suscriptores a BlackBerry.

Fuente: “Bidulka, 2003-2015 annual reports”

ANEXO C: API DataDonors

API REST

Una interfaz de programación de aplicaciones (API) es un conjunto de rutinas, protocolos y herramientas para la construcción de aplicaciones de software.

Una API hace que sea más fácil desarrollar un programa, proporcionando todos los bloques de construcción ("building blocks") para que un programador luego ponga los bloques juntos.

Hay APIs de todo tipo, detallaremos a continuación las APIs Web, que son implementadas por DataDonors para interactuar con su sistema.

Una API web es una interfaz de programación a la que se accede utilizando un protocolo web HTTP (Hyper-Text Transfer Protocol). En esta se exponen servicios que son consumidos por 'requests'. Estos 'request' pueden contener información con formatos estándar.

Anteriormente se utilizaban los Web Services (o servicios web) utilizando los estándares SOAP (Simple Object Access Protocol), SOA (arquitectura orientada a servicios) y ROA (arquitectura orientada a los recursos).

Los estándares más utilizados hoy son XML (eXtensible Markup Language) y JSON (JavaScript Object Notation), ya que reduce considerablemente el tamaño del request, ahorrando ancho de banda al realizar el request al servidor con menos encabezados. (Benslimane, 2008)

Existen diversas arquitecturas de APIs Web. DataDonors implementa una Web API basada en la arquitectura REST (REpresentational State Transfer). REST se definió en el 2000 por Roy Fielding, coautor principal de la especificación del protocolo HTTP.

Existen dos consideraciones de calidad a la hora de aplicar REST:

- Uso correcto de URIs (Uniform Resource Identifier)
- Uso correcto de HTTP

Uso correcto de las URIs

En una web o una aplicación web, las URLs permiten acceder a cada uno de las páginas, secciones o documentos; a esto lo llamamos recursos.

El recurso, por lo tanto, es la información a la que queremos acceder o que queremos modificar o borrar, independientemente de su formato.

Las URL (Uniform Resource Locator) son un tipo de URI (Uniform Resource Identifier), que además de permitir identificar de forma única el recurso, permite localizarlo para acceder a él o compartir su ubicación.

Una URL se estructura de la siguiente forma:

{protocolo}://{dominio o hostname}[:puerto (opcional)]/{ruta del recurso}?{consulta de filtrado}

Por ejemplo, http://www.microsoft.com/busqueda?termino=hola

Existen varias reglas básicas para poner nombre a la URI de un recurso:

- Los nombres de URI no deben implicar una acción, por lo tanto, debe evitarse usar verbos en ellos.
- Deben ser únicas, no debe haber más de una URI para identificar un mismo recurso.
- Deben ser independiente de formato.
- Deben mantener una jerarquía lógica.

- Los filtrados de información de un recurso no se hacen en la URI.

Uso correcto de HTTP

Para desarrollar APIs REST los aspectos claves que hay que dominar y tener claros son:

1. Métodos HTTP
2. Códigos de estado
3. Aceptación de tipos de contenido
4. Métodos.

A la hora de nombrar URIs no se deben incluir verbos que impliquen acción.

Para manipular los recursos, HTTP especifica los siguientes métodos con los cuales operar:

- GET: Para consultar y leer recursos
- POST: Para crear recursos
- PUT: Para editar recursos
- DELETE: Para eliminar recursos.
- PATCH: Para editar partes concretas de un recurso.

HTTP cuenta con una amplia lista de códigos de estado que cubre todas las posibles indicaciones que pueden ocurrir al manipular el recurso. Estos códigos tienen que añadirse en nuestras respuestas cuando las operaciones han ido bien o mal.

HTTP permite especificar en qué formato recibir el recurso, pudiendo indicar varios en orden de preferencia, para ello se utiliza el header “Accept”.

La API devolverá el recurso en el primer formato disponible y, de no poder mostrar el recurso en ninguno de los formatos indicados por el cliente mediante el header “Accept”,

devolverá el código de estado HTTP 406 (No aceptable, el servidor no puede entregar la información).

En la respuesta, se devolverá el header "Content-Type", para que el cliente sepa qué formato se devuelve. (Marqués, 2013)

API DataDonors

DataDonors desarrolló una API REST para interactuar con su sistema. Permite realizar muchas operaciones, solo se detallan las que utiliza este proyecto final.

Crear usuario

Permite la creación de un usuario en DataDonors, no se identifica a la persona, solo se pide un nombre de usuario y un PIN. Como resultado, se obtiene el ID del usuario creado.

Request:

```
POST http://api.wikilife.org/4/user/account/ HTTP/1.1
Accept: */*
Content-Length: 241
Accept-Encoding: identity
User-Agent: NativeHost
Host: api.wikilife.org
Connection: Keep-Alive
Pragma: no-cache

{
"birthdate":"1998-10-05",
"city":"Buenos aires",
"country":"USA",
"device_id":"43fcf5703df2588c9860eb7a1d039c6f58729897",
"gender":"Masculino",
"height":180,
"pin":"1234",
"region":"buenos aires",
"timezone":"GMT",
"userName":"username",
"weight":75
}
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 06 Oct 2015 02:45:01 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
Content-Length: 4
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
Access-Control-Max-Age: 86400
Server: WikilifeAPI tornado/2.4
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type

1
```

Login

Permite identificar al usuario. El resultado es un token que sirve para autenticar los siguientes requests que requieran de permisos para acceder, por ejemplo, el request de guardado de información (o log).

Request:

```
POST http://api.wikilife.org/4/user/login/ HTTP/1.1
Accept: */*
Content-Length: 36
Accept-Encoding: identity
User-Agent: NativeHost
Host: api.wikilife.org
Connection: Keep-Alive
Pragma: no-cache

{"pin":"1234","userName":"username"}
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 06 Oct 2015 02:48:09 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
Content-Length: 43
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
Access-Control-Max-Age: 86400
Server: WikilifeAPI tornado/2.4
X-User-Message: Login success
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type

{"oauth_token": "2S61SQOkc_Oc9DmmN7lAEZzI"}
```

Obtener medicamentos

En el grafo de categorías, los medicamentos tienen ID 16, por lo que se deben obtener los hijos de esa categoría. Devuelve los resultados paginados, por lo que hay que obtener una página a la vez para listar todos los medicamentos.

Request:

```
GET http://api.wikilife.org/4/meta/children/16 HTTP/1.1
User-Agent: Fiddler
Host: api.wikilife.org
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 06 Oct 2015 02:49:36 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
Content-Length: 3796
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
```

```
Access-Control-Max-Age: 86400
Server: WikilifeAPI tornado/2.4
Etag: "20c25e9969a0d57bba7251c769884c73dab2fa98"
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type

{
"pageIndex": 0,
"itemsCount": 25,
"pageSize": 25,
"pageCount": 292,
"itemsTotal": 7280,
"items":
       [{
       "otherNames": "aspirin, 8-hour bayer extended release, tablet",
       "origId": 283420,
       "type": "MetaNode",
       "id": 5314,
       "name": "8-Hour Bayer Extended Release, Tablet"
       },
       (…)],
"itemsTo": 24,
"itemsFrom": 0

}
```

Obtener métricas

Cada medicamento se dosifica utilizando unidades que dependen del mismo. Para conocer la forma de dosificar un medicamento hay que obtener sus métricas.

Request:

```
GET http://api.wikilife.org/4/meta/withmetrics/5533 HTTP/1.1
User-Agent: Fiddler
Host: api.wikilife.org
```

Response:

```
HTTP/1.1 200 OK
Date: Tue, 06 Oct 2015 02:55:01 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
Content-Length: 420
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
Access-Control-Max-Age: 86400
Server: WikilifeAPI tornado/2.4
Etag: "ee6329a7cad6205875d6ebd85e9b648717d458f1"
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type

{
       "metrics": [
       {
              "name": "Strength-Value Node",
              "default": 0,
              "origId": 293008,
              "id": 5535,
              "type": "TextMetricNode",
              "options": "eq 0.1mg base/vial"
       }
       ],
       "name": "Acthrel Injectable",
       "origId": 293006,
```

```
        "otherNames": "corticorelin ovine triflutate, acthrel injectable",
        "type": "MetaNode",
        "id": 5533
}
```


Log de información

De la siguiente forma la API recibe los datos de los usuarios, especificando el ID del nodo a registrar (medicamento), el ID de la métrica a usar (tipo de dosis), y el valor.

Request:


```
POST http://api.wikilife.org/4/logs/?oauth_token=ylWvKrO6mI16a2LrlTJ4Tt6C HTTP/1.1
Accept: */*
Content-Length: 155
Accept-Encoding: identity
User-Agent: NativeHost
Host: api.wikilife.org
Connection: Keep-Alive
Pragma: no-cache

[
  {
    "userId": "ylWvKrO6mI16a2LrlTJ4Tt6C",
    "start": "2015-10-06-0300",
    "end": "2015-10-06-0300",
    "source": "wp",
    "nodes": [
      {
        "nodeId": 5533,
        "metricId": 5535,
        "value": 1
      }
    ]
  },

]
```


Response:


```
HTTP/1.1 200 OK
Date: Tue, 06 Oct 2015 23:32:44 GMT
Connection: keep-alive
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
Access-Control-Max-Age: 86400
Server: WikilifeAPI tornado/2.4
Etag: "fb2a0bb71977f45c11268fe2757a015df139bc29"
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Content-Type
```

ANEXO D: Carta de aceptación

Viernes, 24 de Junio de 2016

La fundación The Wikilife Foundation, recibió y revisó la primer versión del proyecto "Pill Reminder" para Windows Phone, la cual utiliza nuestra plataforma para el registro de medicamentos, en el marco del proyecto final de ingeniería del alumno Alejandro Banzas.
Consideramos que esta aplicación es muy útil para el usuario final, y estamos muy conformes con el trabajo realizado. Continuaremos apoyando el proyecto y lo es muy valioso el uso de la plataforma.

Juan M. Gargiulo
CTO, The Wikilife Foundation