



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico 1

User Stories - Tutor: Nicolás Rinaldi

Ingeniería del Software 2
Segundo Cuatrimestre de 2013

Grupo 1

Integrante	LU	Correo electrónico
Julián Sackmann	540/09	jsackmann@gmail.com
Juan Pablo Darago	272/10	jpgarago@gmail.com
Vanesa Stricker	159/09	vanesastricker@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice

1. Sobre la herramienta	3
2. Product backlog	3
3. Sprint backlog	10

1. Sobre la herramienta

Para el armado del Product y Sprint Backlogs utilizamos la herramienta Agilefant, accesible mediante la URL <http://agilefant.monits.com>. Se dispone de un usuario para el tutor, el cual tiene los siguientes datos de acceso.

- Nombre de usuario: nrinaldi
- Contraseña: monits

2. Product backlog

Incluimos a continuación las *user stories* con su valor, *story points* (que usamos como medida de su dificultad), criterio de aceptación y tareas asociadas. Estas son las correspondientes al *product backlog*.

1) Como atleta quiero tener disponible la lista de entrenamientos que ya concluí

- Story Points: 8
- Value: 5
- Criterio de Aceptación:
 - El atleta puede ver una lista donde tiene cada entrenamiento.
 - Al elegir uno de ellos con su celular, se mostrará el detalle del mismo
- Tareas:
 - Concebir cómo representar un entrenamiento.
 - Investigar opciones para mostrar la información de cada entrenamiento.
 - Crear una vista con una table view para mostrar la lista de entrenamientos concluidos.
 - Crear una vista para mostrar el detalle de entrenamiento.
 - Agregar funcionalidad para que al seleccionar un entrenamiento de la lista de entrenamiento se redirija a la vista de detalle.

2) Como atleta quiero poder ver estadísticas de los entrenamientos que ya hice

- Story Points: 13
- Value: 5
- Criterio de Aceptación:
 - El atleta puede ver para una fase de un entrenamiento realizado el recorrido que hizo y verlo en un mapa similar al que tiene disponible cuando hace el recorrido en una fase.
 - El atleta puede ver para cada fase de un entrenamiento, estadísticas sobre su velocidad y distancia.
 - El atleta puede ver cuánta distancia recorrió para cada entrenamiento, cuanto tardó en cada uno, y cual fue su velocidad máxima.
- Tareas:
 - Investigar si hay una librería que saca estadísticas sola.
 - Codificar las estadísticas.
 - Testear que las estadísticas sean correctas.
 - Probar de intentar sacar estadísticas de datos absurdos.

3) Como atleta quiero que la aplicación siga mi fase dentro del plan para que me avise si lo estoy siguiendo o tengo que modificar mi marcha.

- Story Points: 8
- Value: 21
- Criterio de aceptación:
 - El atleta debe poder saber en que fase de que entrenamiento esta.
 - El atleta debe recibir una notificación de que debe aumentar la marcha si está yendo más lento que el plan.
 - El atleta debe recibir una notificación de que debe disminuir la marcha si está yendo más despacio que el plan.
 - El atleta debe recibir una notificación en intervalos regulares si se está manteniendo en una marcha aceptable.
 - Cada alerta por marcha inválida se repetirá a intervalos regulares mientras persista la condición.
- Tareas:

- Investigar cómo reproducir una canción en cada formato estándar (mp3, wav, etc).
 - Investigar cómo guardar canciones en el teléfono.
 - Codificar la lógica para que si la velocidad no esta en el rango, se envíe una alerta.
 - Testear para los 3 tipos de condiciones de marcha válida e inválida.
- 4) Como atleta quiero poder publicar los recorridos de los entrenamientos en redes sociales y aplicaciones de geolocalización.
- Story Points: 10
 - Value: 1
 - Criterio de aceptación
 - El atleta puede seleccionar en qué red social o aplicación publicar.
 - El atleta puede seleccionar qué recorrido puede publicar.
 - El atleta puede escribir un mensaje.
 - La publicación es visible por los demás miembros de la red social de acuerdo a las reglas de privacidad de la misma.
 - Solo los datos explícitamente indicados por el usuario son publicados en la red social correspondiente.
 - Tareas:
 - Investigar como funcionan la APIs de Facebook
 - Investigar como funcionan la APIs de Google+
 - Investigar como funcionan la APIs de Foursquare
 - Investigar como funcionan la APIs de Nike Run
 - Investigar como funcionan la APIs de Hi5
 - Investigar como funcionan la APIs de MySpace.
 - Codificar las funciones de publicación que hacen uso de cada API
 - Investigar opciones de privacidad para proteger los datos del teléfono.
- 5) Como atleta quiero poder ingresar los datos de mi estado físico.
- Story Points: 5
 - Value: 8
 - Criterio de aceptación
 - El atleta podrá ingresar su peso en kilogramos.
 - El atleta podrá ingresar su altura en cm.
 - El atleta podrá especificar mayores detalles usando categorías basadas en si ya corrió una carrera o no, ya corrió un maratón o no, su mejor marca de distancia en una carrera y en un maratón.
 - La aplicación guardará registro del valor actual de ambos datos.
 - Tareas:
 - Investigar cómo crear un formulario y guardar los datos en la aplicación
 - Investigar cómo validar los campos de acuerdo al tipo de datos pedidos.
 - Pensar cómo representar esos datos del usuario
 - Testear ingresando datos inválidos / absurdos para verificar la consistencia de los mismos dentro de la aplicación.
 - Documentar límites de las validaciones
- 6) Como atleta quiero poder ingresar mi frecuencia semanal con la que puedo entrenar.

- Story Points: 1
 - Value: 5
 - Criterio de aceptación
 - El atleta debe ingresar la frecuencia semanal con la que puede entrenar.
 - Los valores ingresados deben ser una cantidad de días entre 1 y 7.
 - La aplicación guardará registro del valor.
 - Tareas:
 - Investigar implementación de fechas / calendarios en iOS
 - Testear con datos inválidos.
 - Crear una vista para que el usuario seleccione la frecuencia semanal.
 - Guardar la información ingresada por el usuario
- 7) Como atleta quiero poder ingresar el objetivo de mi entrenamiento.
- Story Points: 8
 - Value: 13
 - Criterio de aceptación
 - El atleta debe ingresar sus objetivos propuestos entre las opciones:
 - Correr 5 km sin tiempo.
 - Terminar un maratón olímpico.
 - Correr 7 km en 35 minutos.
- y otras opciones y posibilidades decididas durante la implementación
- Tareas:
 - Codificar la lista de tareas.
 - Investigar forma de agregar elementos a la lista si se fuera a actualizar.
 - Testear agregar objetivos.
 - Investigar que otros objetivos posibles serían interesantes para un potencial usuario.
- 8) Como atleta quiero poder ingresar el plazo estipulado para mi entrenamiento si así lo deseo.
- Story Points: 2
 - Value: 5
 - Criterio de aceptación
 - El atleta puede elegir un plazo estipulado para la finalización de cada uno de sus objetivos.
 - Tareas:
 - Investigar que tipo de granularidad y que tipo de duraciones se pueden soportar: intervalos válidos, etc.
 - Investigar posibles interfaces de usuario para ingresar tiempos, por ejemplo utilizar un calendario para poner una fecha de finalización.
 - Codificar validación de plazos si se utilizan fechas.
- 9) Como atleta quiero que las notificaciones de velocidad de la aplicación sean acordes al de batería seleccionado.
- Story Points: 8
 - Value: 5
 - Criterio de aceptación:

- Si el atleta eligió un consumo bajo, las notificaciones son pitidos y ocurren cada 1 minuto.
 - Si el atleta eligió un consumo alto de batería, las notificaciones son temas musicales preelegidos por la app y ocurren cada 10 segundos.
- Tareas:
 - Investigar cómo obtener la velocidad actual a la que se desplaza el teléfono.
 - Aproximar el consumo de batería de las actualizaciones en función de la frecuencia de la misma.
 - Investigar como controlar la frecuencia de muestreo de velocidad del dispositivo.
 - Escribir el código que permita modificar la frecuencia de muestreo.
 - Escribir el código que obtenga la velocidad del usuario.
- 10) Como atleta quiero que las actualizaciones de posición sean acordes al nivel de batería seleccionado.
- Story Points: 8
 - Value: 2
 - Criterio de aceptación:
 - La posición se actualiza cada 10 segundos si el nivel de consumo batería elegido es alto.
 - La posición se actualiza cada minuto si el nivel de consumo de batería es bajo.
 - Cuanto mayor es el nivel de batería, mayor sera la frecuencia de actualizaciones.
 - Tareas:
 - Investigar si se puede, y en caso de que se pueda como, seleccionar el nivel de actualización del GPS.
 - Investigar cuanto consume la actualización del GPS en función de la frecuencia de refresco establecida.
 - Codificar la funcionalidad de ajuste de frecuencia de actualizaciones del GPS.
- 11) Como atleta quiero poder ajustar el consumo de batería
- Story Points: 3
 - Value: 2
 - Criterio de aceptación:
 - El atleta puede seleccionar dentro de los niveles disponibles, como mínimo bajo, medio y alto
 - La aplicación debe poder correr más tiempo bajo un plan de consumo bajo que en uno alto.
 - El atleta puede determinar que impacto tiene en las funcionalidades de la aplicación el cambio de consumo de batería.
 - Tareas:
 - Investigar que niveles de batería permite el dispositivo.
 - Programar una opción seleccionable para cada nivel de batería y que sea accesible por los demás módulos
 - Documentar para el usuario cual es el impacto de cada nivel de batería en la funcionalidad de la aplicación y en la duración del teléfono (usando estimativos de ser necesario).
 - Incluir esta documentación como ayuda dentro de la aplicación
- 12) Como atleta quiero que la aplicación me de un plan de entrenamiento en base a los datos.

- Story Points: 13
 - Value: 21
 - Criterio de aceptación
 - Si el atleta estableció como objetivo que desea correr una maratón, el sistema creará un plan concentrado en larga duración y velocidad constante.
 - Si el atleta estableció que desea correr una determinada cantidad de kilómetros en un cierto tiempo, el sistema creará un plan con entrenamientos de velocidad progresivamente más difíciles hasta alcanzar el objetivo.
 - Si el atleta no estableció requerimientos ni de distancia ni de tiempo, el programa devolverá una serie de entrenamientos recreativos.
 - Si el atleta se encuentra en buen estado físico, los entrenamientos constarán de fases con mayor exigencia.
 - La duración y velocidad devueltas serán inversamente proporcionales al peso.
 - Tareas:
 - Determinar distancias, velocidades y duraciones para las fases acordes a un plan de entrenamiento, peso y objetivos, consultando a médicos y entrenadores físicos posiblemente.
 - Determinar cómo leer del celular las opciones almacenadas por el corredor.
 - Codificar las reglas de asignación de planes en base a los parámetros indicados en el criterio.
- 13) Como corredor quiero poder ver la velocidad promedio y la duración de cada fase de un entrenamiento para saber el criterio con el que la aplicación mide mi performance corriendo.
- Story Points: 8
 - Value: 13
 - Criterio de aceptación
 - El atleta debe poder elegir un entrenamiento de los que la aplicación ha preparado.
 - El atleta debe poder examinar las fases de un entrenamiento.
 - El atleta debe ver para cada fase un rango de velocidades en km/h que son aceptables.
 - El atleta debe poder ver para cada fase, cuanto tiempo dura la misma en minutos.
 - Tareas:
 - Investigar cómo mostrar los datos por la interfaz del celular.
 - Investigar un algoritmo para lograr calcular la velocidad promedio a medida que llegan los datos.
 - Testear que el promedio calculado es correcto incluso considerando actualizaciones de velocidad y tiempo poco frecuentes (por ejemplo en un modo de batería bajo).
 - Crear una vista para mostrar los datos
 - Implementar la lógica para calcular los datos.
- 14) Como atleta quiero que la aplicación me avise de la próxima fase del plan si ya pasó el tiempo.
- Story Points: 8
 - Value: 21
 - Criterio de aceptación
 - El app genera una notificación auditiva cuando se termine el tiempo de la fase actual.
 - No se genera esa notificación particular por otro motivo.
 - Tareas:
-

- Investigar cómo generar alertas auditivas sencillas.
- Investigar cómo medir el paso del tiempo en el dispositivo (o al menos generar acciones a intervalos de tiempo regular)

15) Como atleta quiero poder ver mi posición en el mapa en tiempo real.

- Story Points: 13
- Value: 21
- Criterio de aceptación:
 - El atleta puede ver su posición actualizada a intervalos regulares en la pantalla.
 - Si se apaga la pantalla o bloquea el teléfono, al reanudar la aplicación la actualización de la posición se reanuda en forma automática.
 - Si se pierde señal de geolocalización, se notifica al usuario.
- Tareas:
 - Implementar la lógica para obtener la posición actual del teléfono
 - Incorporar el uso de mapas en la aplicación
 - Agregar una vista con un mapa en la pantalla.
 - Implementar lógica para centrar ese mapa en una posición indicada.
 - Investigar qué pasa cuando se pasa la aplicación al background y cuando vuelve.
 - Testear que efectivamente la posición se actualice al desplazar el teléfono
 - Testear que el mapa se centre correctamente
 - Testear qué ocurre cuando se manda la aplicación al background.
 - Implementar la lógica para dibujar un recorrido en el mapa

3. Sprint backlog

A continuación detallamos las *user stories* que incluimos en el actual *sprint*. Las mismas son:

- 3) Como atleta quiero que la aplicación siga mi fase dentro del plan para que me avise si lo estoy siguiendo o tengo que modificar mi marcha.
- 14) Como corredor quiero poder ver la velocidad promedio y la duración de cada fase de un entrenamiento para saber el criterio con el que la aplicación mide mi performance corriendo.
- 16) Como atleta quiero poder ver mi posición en el mapa en tiempo real.

y a continuación incluimos el detalles de las tareas involucradas y la dificultad asociada a cada una.

- Investigar cómo reproducir una canción en cada formato estándar (mp3, wav, etc).
Dificultad: 3
- Investigar cómo guardar canciones en el teléfono.
Dificultad: 3
- Codificar la lógica para que si la velocidad no esta en el rango, se envíe una alerta.
Dificultad: 5
- Testear para los 3 tipos de condiciones de marcha válida e inválida.
Dificultad: 2
- Investigar cómo mostrar datos numéricos de velocidad y duración por la interfaz del celular, y como actualizar la vista cuando estos cambian.
Dificultad: 3
- Investigar un algoritmo para lograr calcular la velocidad promedio a medida que llegan los datos.
Dificultad: 2
- Testear que el promedio calculado es correcto incluso considerando actualizaciones de velocidad y tiempo poco frecuentes (por ejemplo en un modo de batería bajo).
Dificultad: 5
- Crear una vista para mostrar los datos.
Dificultad: 3
- Implementar la lógica para calcular los datos de posición actual
Dificultad: 5
- Implementar la lógica para obtener la posición actual del teléfono.
Dificultad: 8
- Incorporar el uso de mapas de otras fuentes dentro de la aplicación
Dificultad: 3
- Agregar una vista con un mapa en la pantalla.
Dificultad: 5
- Implementar lógica para centrar ese mapa en una posición indicada.
Dificultad: 3
- Investigar qué pasa cuando se pasa la aplicación al *background* y cuando vuelve, implementar la lógica que mantenga actualizando a la aplicación incluso en background.
Dificultad: 3

- Testear que efectivamente la posición se actualice al desplazar el teléfono.

Dificultad: 2

- Testear que el mapa se centre correctamente.

Dificultad: 2

- Implementar la lógica para dibujar un recorrido en el mapa dados los puntos y un *timestamp* para cada uno.

Dificultad: 5