



# Plan de Pruebas

02-2017

---

Universidad de Los Andes

MISO 4208 - Pruebas Automáticas

Felipe Martinez (f.martinez2@uniandes.edu.co)

Juan Hernández (js.hernandez15@uniandes.edu.co)

## Estrategia de pruebas

### Alcance

#### Aplicaciones

La ejecución de pruebas será realizada sobre las siguientes aplicaciones:

- [Habitica](#): es una aplicación web/juego diseñada como un juego con la intención de ayudar a los jugadores a mejorar hábitos de la vida real. Permite convertir todas las tareas de los jugadores en recompensas para un personaje tales como objetos y experiencia.
- [Habitica Mobil](#): aplicación Android para Habitica
- [GNU Cash](#): es una aplicación Android para control de gastos financieros. Sirve como complemento a la versión de escritorio con la que puede sincronizarse. Con ella se puede realizar control de gastos, trazabilidad de cuentas

#### Testers

- Felipe Martínez
- Juan Hernández

#### Tareas

Entre las tareas de los testers se encuentran:

- Investigación sobre las herramientas a usar para las pruebas automáticas.
- Pruebas de concepto de los frameworks y herramientas a utilizar.
- Configuración del entorno para cada una de las aplicaciones.

### Presupuesto

El presupuesto de pruebas para la segunda iteración está distribuido de la siguiente manera:

- Tiempo Hombre: 26 horas (13 horas 2 Testers)
- Tiempo máquinas físicas: 10 horas
- Tiempo servicios Cloud:
  - 100 minutos (BrowserStack)
- Equipos de cómputo
  - 2 Laptops
  - 1 Celular Samsung J5

## Niveles y Objetivos

### Sistema

- Ejecución de pruebas automáticas de tipo Behavior Driven Testing sobre múltiples exploradores.
- Ejecución de pruebas con el enfoque *Record and Replay*.
- Mutación del código fuente de las aplicaciones para verificar el comportamiento de las suites de pruebas de las aplicaciones compiladas con el código fuente mutado.

## Entregables

Los documentos entregables están enmarcados en los reportes generados por las herramientas usadas a través del curso de pruebas automáticas de software. Adicionalmente se relacionarán las evidencias de la ejecución de pruebas tales como capturas de pantallas o videos de acuerdo a las características de los frameworks que se utilicen para realizar las pruebas. Los artefactos están almacenados en el [repositorio del equipo](#) en la ruta [Documentación > Sprint 3](#)

## Plan de pruebas - Sprint 3

### Objetivo del sprint

- Aumentar la cobertura de pruebas usando *Behavior Driven Testing* con múltiples exploradores.
- Aumentar la cobertura de pruebas haciendo uso de la herramienta *Expresso Recorder*.
- Explorar los frameworks existentes para la aplicación de pruebas de mutación sobre las aplicaciones seleccionadas en el sprint
- Aplicar el framework MDroid+ a las aplicaciones android Habitica y GNU Cash

### Testing - App1: Habitica (Web)

#### Tipos de Pruebas

##### Sistema

- Behavior Driven Testing con Cucumber: ejecutar la pruebas basadas en escenarios escritos sintaxis Gherkin sobre las características de *Log In* y creación de Hábitos sobre múltiples exploradores (Chrome 62.0, IE 11.0, Firefox 56.0, Safari 10.1).

### Testing - App2: Habitica (Android)

#### Tipos de Pruebas

##### Sistema

- Mutación:

### Testing - App3: GNU Cash (Android)

#### Tipos de Pruebas

##### Sistema

- Ejecución de pruebas con el enfoque *Record and Replay*: Grabar pruebas sobre diversas funcionalidades del sistema.
- Mutación: consiste en seleccionar un conjunto de clases java que serán mutadas para verificar la efectividad de la suite de pruebas unitarias del proyecto.

1. Seleccionar un variante de construcción

(<https://developer.android.com/studio/build/build-variants.html>)



2. Determinar el estado de la suite de pruebas de la aplicación
3. Aplicar la mutación
4. Ejecutar la suite de pruebas sobre cada mutante
5. Determinar la efectividad de la suite de pruebas para encontrar al mutante