

# BeatIt!

## Informe Final de Verificación

### Versión 4.1.2



## Historia de revisiones

| Fecha      | Versión | Descripción     | Autor            |
|------------|---------|-----------------|------------------|
| 22/11/2014 | 4.1.1   | Elaboración     | Emiliano Vázquez |
| 24/11/2014 | 4.1.2   | Revisión de SQA | Pablo Olivera    |
|            |         |                 |                  |
|            |         |                 |                  |

# Contenido

## [Resumen de la verificación](#)

### [Planificado vs. Realizado](#)

[Cambios de alto impacto](#)

[Cambios de mediano impacto](#)

[Cambios de bajo impacto](#)

### [Cantidad de errores encontrados](#)

#### [Documentos](#)

[Modelo de Dominio](#)

[Documento de Diseño del Prototipo](#)

[Documento de Requerimiento para el Prototipo](#)

[Plan del Proyecto](#)

[Descripción del Proyecto](#)

[Plan de la Integración de la Iteración](#)

[Modelo de Casos de Uso](#)

[Especificación de Desafíos](#)

[Documento de Riesgos](#)

[Especificación de Requerimientos](#)

[Descripción de la Arquitectura](#)

#### [Versiones y sus Casos de Uso](#)

##### [Prototipo 1 - Usain Bolt](#)

[Pruebas Unitarias](#)

[Pruebas de Integración](#)

[Pruebas del Sistema](#)

##### [Prototipo 2 - 4 desafíos en ambas plataformas](#)

[Pruebas Unitarias](#)

[Pruebas de Integración](#)

[Pruebas del Sistema](#)

##### [BeatIt! Versión Beta](#)

[Pruebas Unitarias](#)

[Pruebas de Integración](#)

[Pruebas del Sistema](#)

##### [BeatIt! Versión Alpha](#)

[Pruebas Unitarias](#)

[Pruebas de Integración](#)

[Pruebas del Sistema](#)

## [Evaluación](#)

[Cantidad de pruebas unitarias muy bajas](#)

[No utilización de TDD](#)

[Tecnologías disponibles no usadas a pleno](#)

[Aprobado con observaciones](#)

[Puesta en marcha del Sistema de Integración Continua](#)

[Aprobado](#)

[Informe final de las métricas de Pruebas Cubiertas](#)

[Planificado total de todas las pruebas](#)

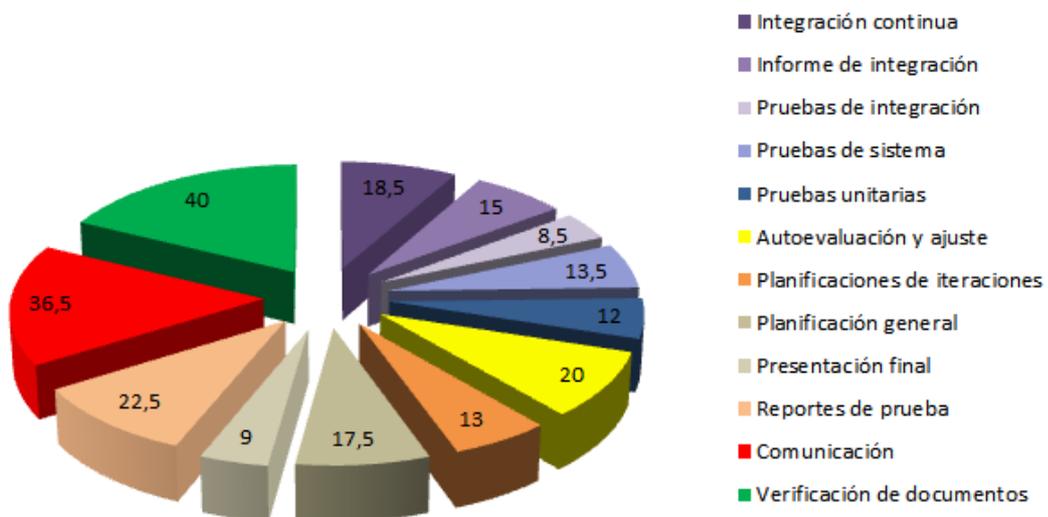
[Pruebas de sistema reales](#)

[Evaluación del tiempo invertido en verificación](#)

## 1. Resumen de la verificación

- Revisiones semanales de los principales documentos de la línea base y otros, en búsqueda de inconsistencias, insuficiencias en lo especificado y adecuada presentación de la información
- Diseño del modelo de casos de prueba unitarias
- Ejecución de pruebas de sistema
- Puesta en marcha del sistema de integración continua
  - Estado de compilación
  - Monitoreo de pruebas unitarias
  - Cobertura de código
  - Inspección de código severa
- Generación de métricas

Distribución de tiempo de dedicación del responsable de verificación



### 1.1. Planificado vs. Realizado

#### 1.1.1. Cambios de alto impacto

El plan original de verificación fue altamente modificado en el transcurso del desarrollo del producto. Comenzando con un plan centrado en pruebas unitarias que operarán con un sistema de Integración Continua y justificarán la creación de Mock Services puesto que la aplicación cuenta con una gran cantidad de componentes externos. A lo que realmente se hizo que fue una verificación más volcada a pruebas de sistema por parte de un gran número de integrantes del equipo y mucho menos centrada en pruebas unitarias. Igualmente se alcanzó la cobertura deseada, la cantidad de pruebas unitarias es poca pero cada una de estas comprenden múltiples funcionalidades del software.

A raíz de esta experiencia ganada en Proyecto de Ingeniería de Software, concluimos, que idealmente si es mejor un fuerte respaldo en pruebas unitarias, a modo de "documentar" el comportamiento que debería tener el software, lo cual alertará de defectos a nivel más granular y prematuro, así pues evitaremos que los mismos se manifiesten en pruebas de sistema, lo cual ha significado un esfuerzo en trazar desde capa de presentación a capa lógica los defectos en cuestión para sus correcciones.

También es recomendable evitar consolidar mucho funcionamiento en una misma prueba unitaria, y si hacer honor al término "unitario", de ser así invertiremos más

esfuerzo en codificar las pruebas pero será más puntual la alerta que obtendremos ante un fallo.

En resumen, los defectos detectados en pruebas de sistema han sido numerosos, las horas de retrabajo muchas.

Como contraparte a esta conclusión, a sido muy discutido la inclusión de Mock Services como herramienta para esta instancia. Actualmente se reconocen como una poderosa herramienta de testing que permite incluir en pruebas unitarias parte de la comunicación entre componentes (incluso externos al sistema).

Es real que librerías puntuales para controlar GPS, acelerómetro, cámara, demás sensores o incluso login contra facebook se han hecho presentes en este producto, generando un contexto propicio para la utilización de Mock Services, pero debido a plazos, cantidad de recursos y otras limitaciones la propia implementación de los Mock Services implicaban un costo muy alto y desplazaban otras tareas más prioritarias. Entonces concluimos como un acierto haberlos desestimado para BeatIt!.

### **1.1.2. Cambios de mediano impacto**

Una desviación más ha sido incrementar de 70% como mínimo de cobertura aceptable a 75%, a pesar de haber sido un riesgo alto el cumplimiento de este criterio hasta el último momento, no suponíamos que 70% llegaba a reflejar calidad en el producto así pues, igualmente, decidimos incrementar un poco más y centrarlo en la lógica sin excluir del todo el código correspondiente a la capa de presentación.

Los documentos verificados, han sido seis promedio por semana en la primera mitad del proceso de desarrollo, para luego pasar a ser cuatro.

Estas cantidades han sido las planificadas y las realizadas, suponiendo un gasto de dos horas todos los domingos previos a las entregas. A su vez paralelamente el responsable de SQA como asistente de verificación y el responsable de V&V como asistente de SQA estaban altamente coordinados en pos de un aprovechamiento de trabajo más cooperativo.

A lo largo del proceso los documentos fueron estabilizándose y confluyendo hacia el producto que resulta ser hoy BeatIt!, así entonces las correcciones hechas se distribuyen en las primeras semanas.

### **1.1.3. Cambios de bajo impacto**

Nuestro modelo de proceso se ha basado en el MUM, el área de verificación y validación no fue la excepción. Todos los documentos que corresponden fueron entregados en los plazos que se estipulan salvo *Informe de verificación de integración (del prototipo)* en la semana tres que resultó entregado en la semana cuatro.

Como adaptación se suprimió la Revisión por Pares no entendiendo que sea una limitante para llegar a un excelente producto.

Y se fusionaron los documentos Evaluación de la Verificación con el Reportes de pruebas en un único archivo con dos secciones por entender que van de la mano con la información que manejan y sus enfoques, aparte también el Reporte de Prueba carece de plantilla en el MUM.

## **1.2. Cantidad de errores encontrados**

En esta sección se detallan los documentos verificados especificando la cantidad de veces que fue revisado (distintas versiones) y la cantidad de errores encontrados.

### **1.2.1. Documentos**

#### *1.2.1.1. Modelo de Dominio*

Versión 1.3 - Cinco errores (V&V)

#### *1.2.1.2. Documento de Diseño del Prototipo*

Versión 1.0 - Tres errores (V&V)

#### *1.2.1.3. Documento de Requerimiento para el Prototipo*

Versión 1.0 - Un error encontrado (V&V)

#### *1.2.1.4. Plan del Proyecto*

Versión 1.0 - Un error encontrado (V&V)

#### *1.2.1.5. Descripción del Proyecto*

Versión 1.4 - Cuatro errores (V&V)

#### *1.2.1.6. Plan de la Integración de la Iteración*

Versión 1.2.3 - Un error encontrado (V&V)

#### *1.2.1.7. Modelo de Casos de Uso*

Versión 1.2.4 - Cuatro errores encontrados (V&V)

Versión: 2.1.2 - Tres errores encontrados (SQA)

#### *1.2.1.8. Especificación de Desafíos*

Versión 1.2.3 - Un error encontrado (V&V)

Versión: 2.2.2 - Tres errores encontrados (SQA)

Versión 2.2.3 - Dos errores encontrados (V&V)

Versión: 3.2.1 - Un error encontrado (SQA)

#### *1.2.1.9. Documento de Riesgos*

Versión 2.1.2 - Tres errores encontrados (V&V)

#### *1.2.1.10. Especificación de Requerimientos*

Versión: 1.1.1.6 - Seis errores encontrados. (SQA)

Versión: 2.1.1 - Un error encontrado. (SQA)

#### *1.2.1.11. Descripción de la Arquitectura*

Versión: 1.2.2 - Cuatro errores encontrados. (SQA)

### **1.2.2. Versiones y sus Casos de Uso**

#### **1.2.2.1. Prototipo 1 - Usain Bolt**

##### *1.2.2.4.1 Pruebas Unitarias*

No determinadas aún, las pruebas unitarias en las primeras etapas se encontraron disminuidas como ya fue mencionado.

Cantidad de verificadores: 0

Cantidad de horas invertidas: 0

Cobertura de Código: No aplica.

##### *1.2.2.1.2 Pruebas de Integración*

No aplica, al tratarse del primer prototipo no hubo que hacer pruebas de integración contra sistemas anteriores. Por otro lado, esta versión no utilizaba aún servidor web, no habiendo grandes componentes que interopere entre sí.

Cantidad de verificadores: 0

Cantidad de horas invertidas: 0

#### 1.2.2.1.3 Pruebas del Sistema

Las pruebas de sistema registradas se han entregado en los Reportes de Pruebas quincenales, este documento es producto del registro en GitHub sección Issues de las pruebas de sistema (tanto las que están dentro, como las que están fuera de plan de verificación).

Las Issues describen los flujos de sistema para reproducir resultados inesperados, esta iteración presenta el reporte bajo el nombre de archivo VREVG6v4.1.2 y las métricas correspondientes se presentan al final de este documento.

Cantidad de verificadores: 5-7 (la mayoría de los implementadores y el responsable de verificación)

Cantidad de horas invertidas: 65.5

#### 1.2.2.2. Prototipo 2 - 4 desafíos en ambas plataformas

"Usain Bolt", "Callar al perro", "Despertame a tiempo" e "Invitá a tus amigos"

##### 1.2.2.4.1 Pruebas Unitarias

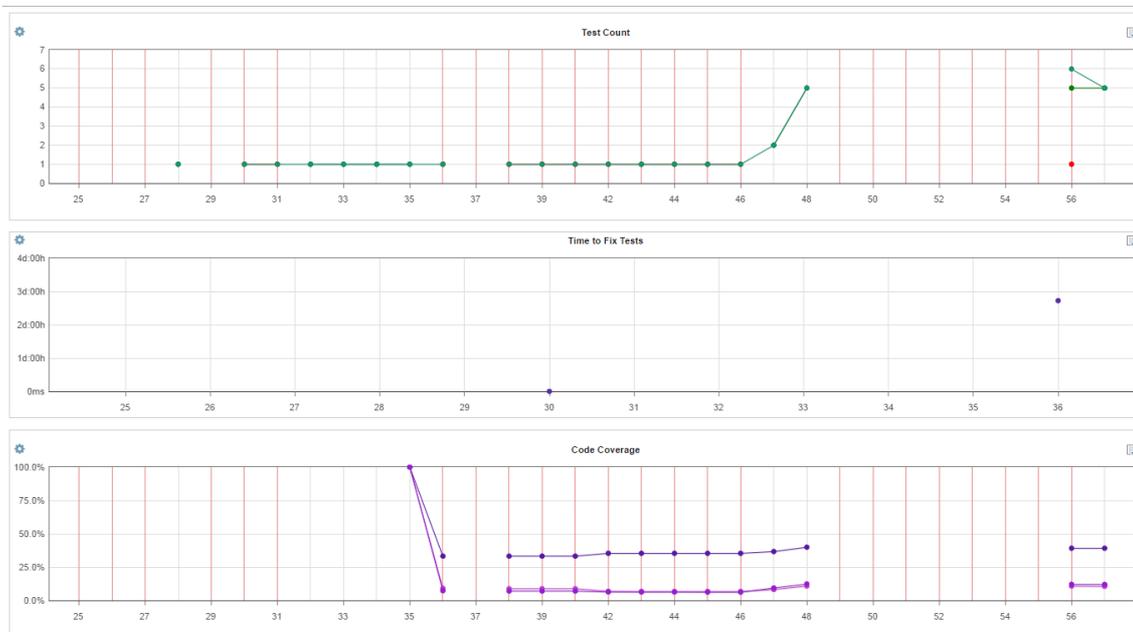
Gráfica y Tabla de Datos

Iteración 1 (planificado)



Para el segundo prototipo, recién el equipo de implementadores empezaba a incursionar en las pruebas unitarias, con pruebas muy simples que dieron lugar en las próximas fases a pruebas más complejas y más abarcativas.

Los siguientes gráficos tiene por eje de las Xs el número de compilación (ligeramente superior al número de versionado de entregas al repositorio).



Explicación Figura 1, se muestra la evolución de la cantidad de tests, como en la compilación número 36 comenzó a fallar el único que había para pasar a ser solucionado en casi tres días (como indica la Figura 2).

En esta iteración se disponía de sistema de integración continua hace relativamente poco e incluso luego continuó siendo configurado. Por esto mismo se puede observar un problema técnico entre el rango de días que encierra las compilaciones 48 hasta la 56, momento en el cual se retoma el servicio.

Las referencias de colores pueden consultarse en los gráficos de la siguiente iteración.

Por último destacar que el servicio de "code coverage" fue configurado luego de la puesta en marcha, se puede ver que fue activado en el build #35 y configurado los modulos a considerarse en el build #36.

Cantidad de verificadores: 2 (un asistente verificador/implementador y un implementador, de distintas plataformas)

Cantidad de horas invertidas: 10

Cobertura de Código:

Overall Coverage Summary for Assembly: BeatIt!

| Namespace | Class, %      | Method, %     | Statements, %    |
|-----------|---------------|---------------|------------------|
| all types | 35.4% (17/48) | 9.3% (31/334) | 10.6% (341/3225) |

Coverage Breakdown

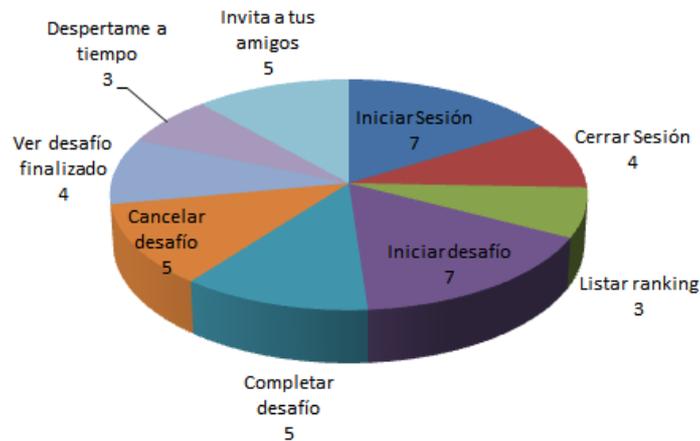
| Namespace                     | Class, %     | Method, %     | Statements, %   |
|-------------------------------|--------------|---------------|-----------------|
| BeatIt_                       | 0% (0/2)     | 0% (0/13)     | 0% (0/59)       |
| BeatIt_AppCode.Challenges     | 100% (10/10) | 57.7% (15/26) | 63.3% (150/237) |
| BeatIt_AppCode.Classes        | 100% (4/4)   | 100% (7/7)    | 100% (60/60)    |
| BeatIt_AppCode.Controllers    | 50% (1/2)    | 28% (7/25)    | 32.9% (110/334) |
| BeatIt_AppCode.CustomControls | 0% (0/2)     | 0% (0/4)      | 0% (0/33)       |
| BeatIt_AppCode.Datatypes      | 100% (2/2)   | 100% (2/2)    | 100% (21/21)    |
| BeatIt_AppCode.Pages          | 0% (0/9)     | 0% (0/50)     | 0% (0/600)      |
| BeatIt_Pages                  | 0% (0/5)     | 0% (0/41)     | 0% (0/597)      |
| BeatIt_Resources              | 0% (0/1)     | 0% (0/32)     | 0% (0/101)      |
| SQLite                        | 0% (0/11)    | 0% (0/134)    | 0% (0/1183)     |

### 1.2.2.2.2 Pruebas de Integración

Gráfica y Tabla de Datos

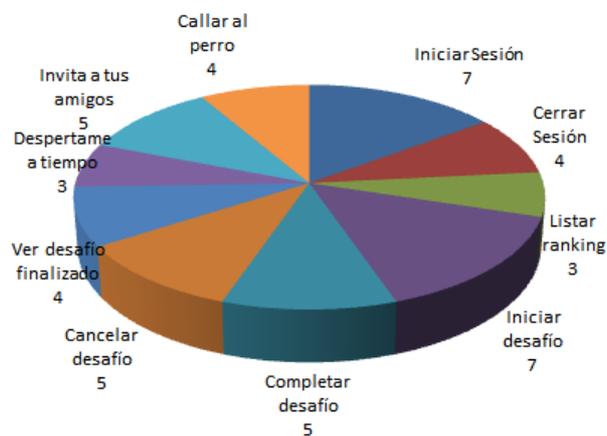
Iteración 1 (planificado)

## Pruebas de Integración



Iteración 2 (planificado)

## Pruebas de Integración



En aspectos generales nuestro sistema en esta fase se encontró en diferentes situaciones un mismo día, dado el dinamismo de esta temprana etapa, el sistema dejaba de compilar varias veces por día, como indica la Figura 1, que por ejemplo señala que el 4 de octubre solo el 40% del día el sistema compilaba.

Asimismo se puede ver que el 30 de septiembre, 2 de octubre y 3 de octubre el sistema ante nuevas integraciones dejó de compilar.

Las integraciones no solo afectan al estado de compilación sino también a la duración de la misma como muestra la Figura 2 (demás esta decir que también afecta los resultados de las pruebas unitarias).



La figura 4 si fue la última funcionalidad configurada el nuestro sistema de integración continua, e indica a través de una inspección severa de código cuantos antipatrones detecta.

Cantidad de verificadores: 3 (dos implementadores de cada plataforma y el verificador)

Cantidad de horas invertidas: 16

### 1.2.2.2.3 Pruebas del Sistema

Gráfica y Tabla de Datos

Iteración 1



Iteración 2

## Pruebas de Sistema



Las pruebas de sistema han sido las planificadas y muchas otras tantas sin registrar llevadas a cabo por integrantes no pertenecientes al equipo de verificación.

Los registros de fallos se presentan en las métricas finales y fueron muchos.

Cantidad de verificadores: 6 (dos implementadores, tres asistente de verificación y el responsable de verificación)

Cantidad de horas invertidas: 72

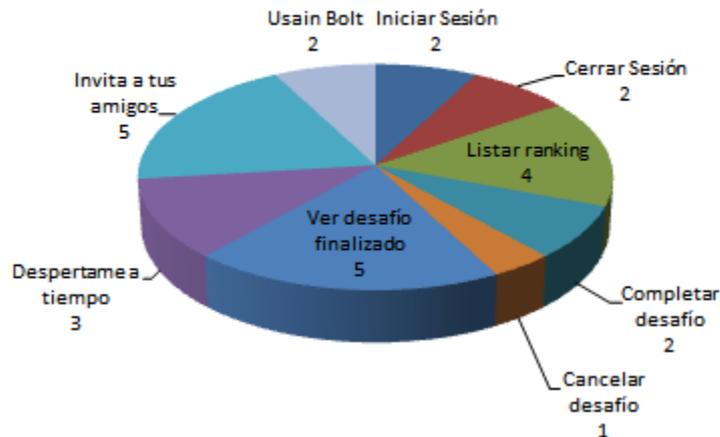
### 1.2.2.3. **BeatIt! Versión Beta**

#### 1.2.2.4.1 Pruebas Unitarias

Gráfica y Tabla de Datos

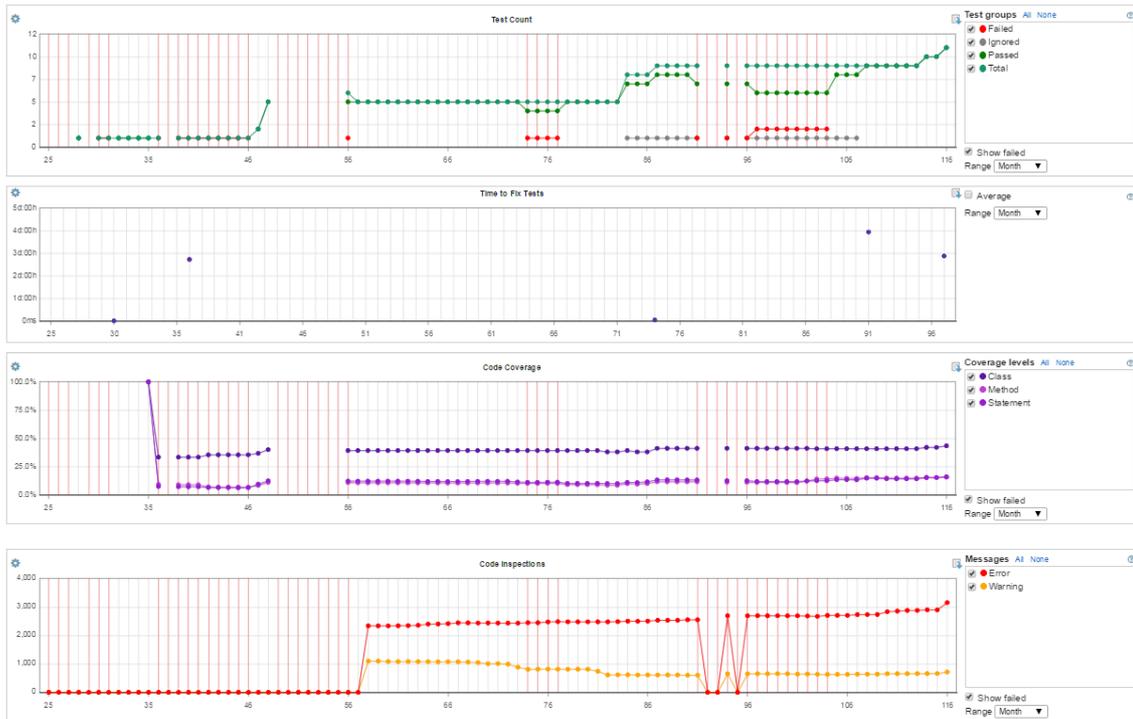
Iteración 1 y 2 (planificado)

## Pruebas unitarias



En esta iteración, tuvo mayor presencia las pruebas unitarias, los equipos comenzaban a acostumbrarse a dedicar el 20% del tiempo en verificación. Se acordó por plataforma un coordinador con el responsable de verificación (sea este asistente verificador o no).

La comunicación fue vital para dar a conocer los Modelos de Casos de Prueba, Planificaciones de las Verificaciones de cada Iteración, Evaluar y ajustar.



Cantidad de verificadores: 4-6 (la mayoría de los implementadores)

Cantidad de horas invertidas: 27

Cobertura de Código:

Windows Phone

Overall Coverage Summary for Assembly: BeatIt!

| Namespace | Class, %      | Method, %    | Statements, %    |
|-----------|---------------|--------------|------------------|
| all types | 34.8% (16/46) | 14% (62/443) | 12.4% (507/4091) |

| Namespace                     | Class, %     | Method, %     | Statements, %   |
|-------------------------------|--------------|---------------|-----------------|
| BeatIt                        | 0% (0/2)     | 0% (0/13)     | 0% (0/59)       |
| BeatIt_AppCode.Challenges     | 100% (10/10) | 67.5% (27/40) | 64.9% (289/445) |
| BeatIt_AppCode.Classes        | 100% (2/2)   | 75% (3/4)     | 90.5% (38/42)   |
| BeatIt_AppCode.Controllers    | 50% (1/2)    | 19.2% (5/26)  | 26.1% (79/303)  |
| BeatIt_AppCode.CustomControls | 0% (0/2)     | 0% (0/4)      | 0% (0/33)       |
| BeatIt_AppCode.Datatypes      | 100% (2/2)   | 100% (2/2)    | 100% (21/21)    |
| BeatIt_AppCode.Pages          | 0% (0/14)    | 0% (0/134)    | 0% (0/1742)     |
| BeatIt_Resources              | 100% (1/1)   | 29.1% (25/86) | 30.4% (80/263)  |
| SQLite                        | 0% (0/11)    | 0% (0/134)    | 0% (0/1183)     |

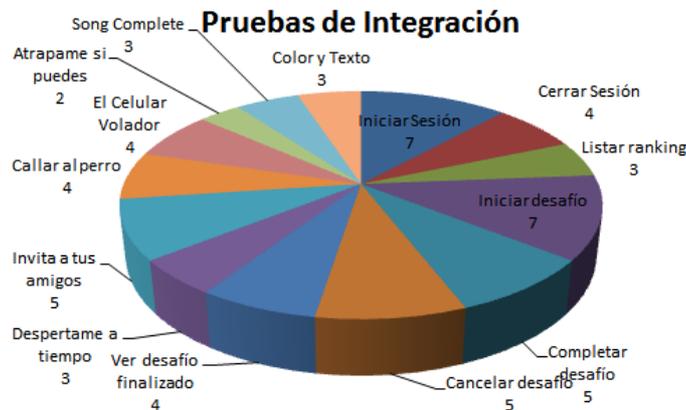
## Android

| Element                        | Coverage | Covered Instructio... | Missed Instructions | Total Instructions |
|--------------------------------|----------|-----------------------|---------------------|--------------------|
| Tests (02/11/2014 03:00:38 PM) |          |                       |                     |                    |
| src                            | 59,3 %   | 3.343                 | 2.290               | 5.633              |
| controllers                    | 59,5 %   | 3.343                 | 2.272               | 5.615              |
| datatypes                      | 0,0 %    | 0                     | 1.058               | 1.058              |
| connection                     | 40,1 %   | 315                   | 471                 | 786                |
| challenges.invitefriends       | 0,0 %    | 0                     | 458                 | 458                |
| entities                       | 77,9 %   | 215                   | 61                  | 276                |
| challenges.usainbolt           | 96,4 %   | 1.156                 | 43                  | 1.199              |
| challenges.shutthedog          | 87,4 %   | 285                   | 41                  | 326                |
| challenges.wakemeup            | 93,1 %   | 323                   | 24                  | 347                |
| interfaces                     | 85,8 %   | 145                   | 24                  | 169                |
| challenges.songcomplete        | 0,0 %    | 0                     | 18                  | 18                 |
| challenges.bouncinggame        | 84,5 %   | 82                    | 15                  | 97                 |
| challenges.selfiegroup         | 85,7 %   | 84                    | 14                  | 98                 |
| challenges.catchme             | 92,1 %   | 152                   | 13                  | 165                |
| challenges.textandcolor        | 95,6 %   | 259                   | 12                  | 271                |
| challenges.throwthephone       | 94,6 %   | 194                   | 11                  | 205                |
| gen                            | 93,7 %   | 133                   | 9                   | 142                |
| gen                            | 0,0 %    | 0                     | 18                  | 18                 |

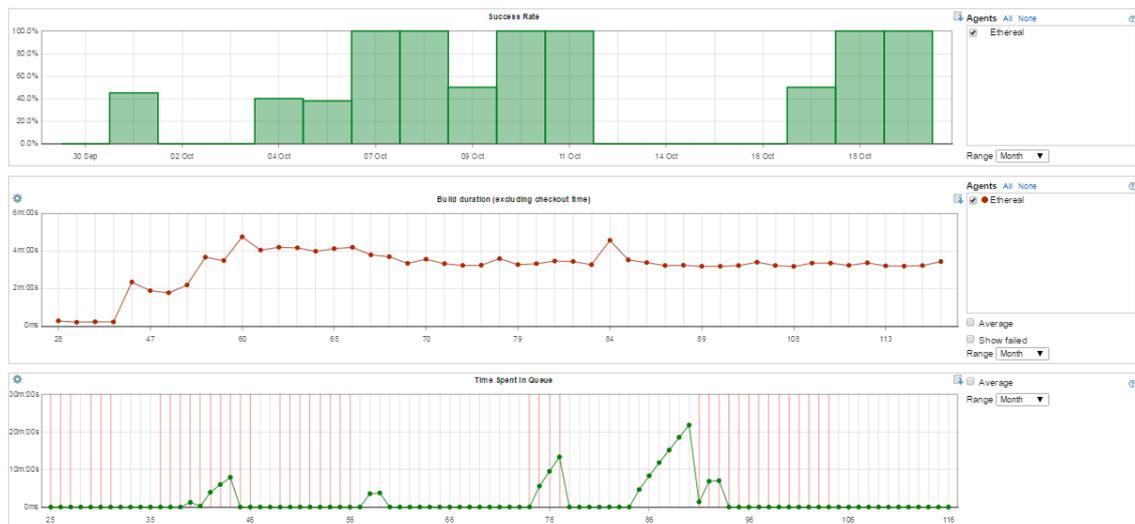
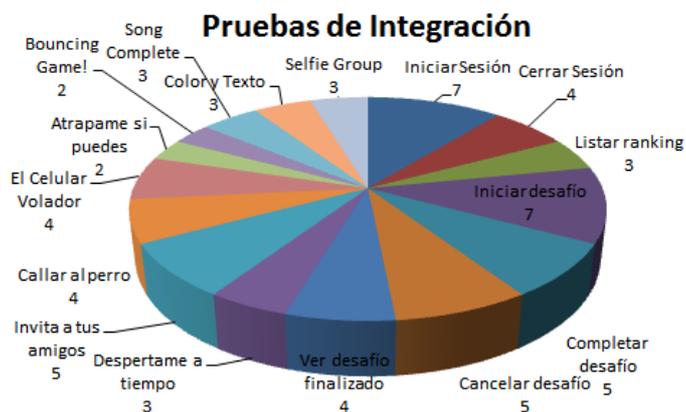
### 1.2.2.3.2 Pruebas de Integración

Gráfica y Tabla de Datos

## Iteración 1 (planificado)



## Iteración 2 (planificado)



El análisis de los gráficos es el siguiente, se muestra en la Figura 1 como el sistema tiene relativamente alta disponibilidad, es decir la versión del repositorio compila.

La Figura 2 expresa el aumento de la complejidad en la solución puesto que la duración de la compilación es un buen indicador del tamaño del software (para una máquina dedicada como es el caso).

La Figura 3 por último es interesante porque indica el tiempo que los implementadores han esperado para que el sistema de integración continúe les ejecute las tareas correspondientes a sus entregas al repositorio, los picos en este

gráfico demuestran concurrencia en el recurso, puesto que cada cambio al repositorio conforma una cola de solicitudes.

Cantidad de verificadores: 2 (un implementador y el responsable de verificación)

Cantidad de horas invertidas: No determinado

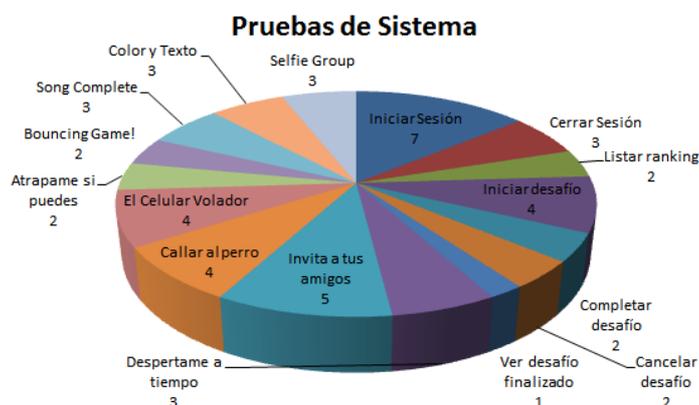
### 1.2.2.3.3 Pruebas del Sistema

Gráfica y Tabla de Datos

Iteración 1 (planificado)



Iteración 2 (planificado)



En esta iteración solo dos asistentes verificadores y el responsable llevaban a cabo el plan de verificación. Por otro lado en la versión Beta del producto, los 7 implementadores (restantes) invertían esfuerzo en pruebas de sistema también.

Cantidad de verificadores: 9 (seis implementadores, un implementador/asistente verificador, un asistente verificador y el responsable)

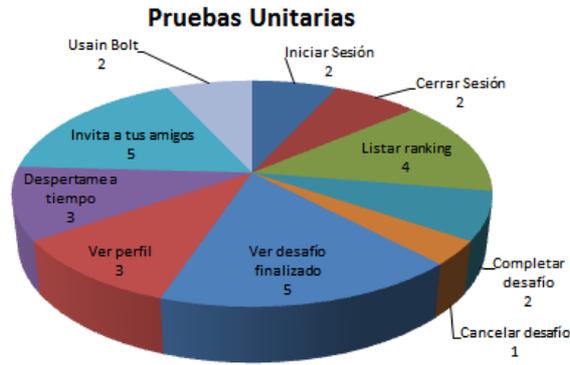
Cantidad de horas invertidas: 80.5

### 1.2.2.4. BeatIt! Versión Alpha

#### 1.2.2.4.1 Pruebas Unitarias

Gráfica y Tabla de Datos

Iteración única (planificado)



Finalmente en un alto esfuerzo de compensación ambas plataformas logran alcanzar el mínimo de cobertura por pruebas unitarias establecido. A este punto casi todos los implementadores tuvieron experiencia de pruebas unitarias y en esfuerzo conjunto se superó 75% de cobertura.

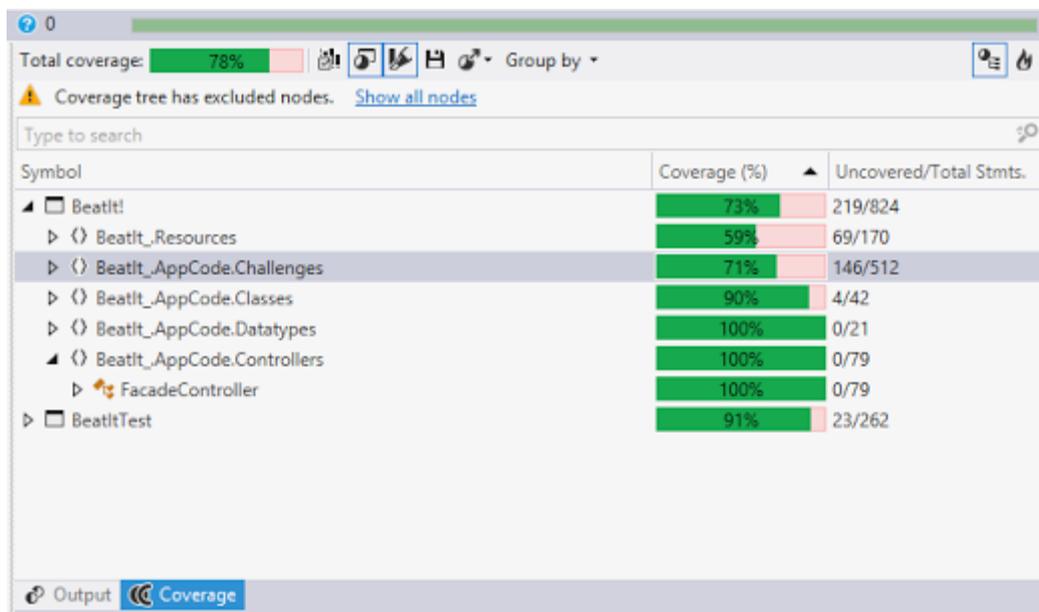
El conjunto de pruebas unitarias no corresponden en su totalidad con lo planificado, las realizadas en ocasiones comprenden más de una de las propuestas en el Plan de Verificación de la Iteración.

Cantidad de verificadores: 5-7 (la mayoría de los implementadores)

Cantidad de horas invertidas: 78

Cobertura de Código:

Windows Phone

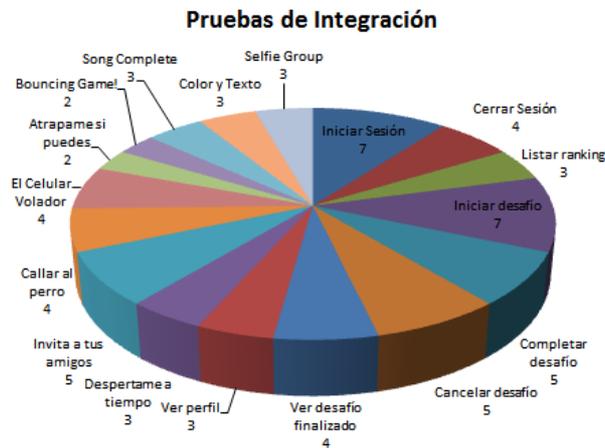


Android

| Element                  | Coverage | Covered Instructio... | Missed Instructions | Total Instructions |
|--------------------------|----------|-----------------------|---------------------|--------------------|
| Tests                    | 75,4 %   | 4.374                 | 1.425               | 5.799              |
| src                      | 75,7 %   | 4.374                 | 1.407               | 5.781              |
| connection               | 0,0 %    | 0                     | 458                 | 458                |
| datatypes                | 46,8 %   | 368                   | 418                 | 786                |
| contollers               | 79,4 %   | 972                   | 252                 | 1.224              |
| challenges.invitefriends | 77,9 %   | 215                   | 61                  | 276                |
| challenges.usainbolt     | 87,4 %   | 285                   | 41                  | 326                |
| entities                 | 96,7 %   | 1.159                 | 40                  | 1.199              |
| challenges.shutthedog    | 93,1 %   | 323                   | 24                  | 347                |
| challenges.wakemeup      | 85,8 %   | 145                   | 24                  | 169                |
| interfaces               | 0,0 %    | 0                     | 18                  | 18                 |
| challenges.songcomplete  | 84,5 %   | 82                    | 15                  | 97                 |
| challenges.selfiegroup   | 92,1 %   | 152                   | 13                  | 165                |
| challenges.catchme       | 95,6 %   | 259                   | 12                  | 271                |
| challenges.bouncinggame  | 88,8 %   | 87                    | 11                  | 98                 |
| challenges.textandcolor  | 94,6 %   | 194                   | 11                  | 205                |
| challenges.throwthephone | 93,7 %   | 133                   | 9                   | 142                |
| gen                      | 0,0 %    | 0                     | 18                  | 18                 |

### 1.2.2.4.2 Pruebas de Integración

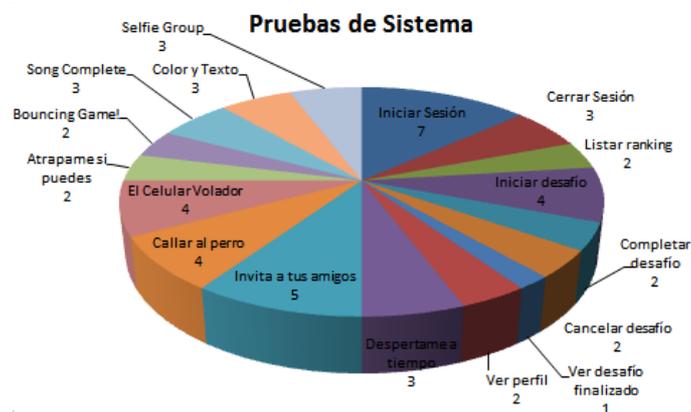
Gráfica y Tabla de Datos  
Iteración única (planificado)



Cantidad de verificadores:  
Cantidad de horas invertidas: No determinado

### 1.2.2.4.3 Pruebas del Sistema

Gráfica y Tabla de Datos  
Iteración única (planificado)



Al igual que la fase anterior, el mismo par de asistentes y el responsable verificador cumplían una rutina a los 3 días previos a la liberación de la ejecución de las pruebas de sistema planificadas, pero a ello se suma de manera muy significativa el esfuerzo colectivo de todo el resto de los integrantes. En esta fase a diferencia de la anterior no solo todos los implementadores realizan pruebas durante la mayor parte del tiempo sino que el equipo en su totalidad.

Cantidad de verificadores: 12 (todo el equipo)

Cantidad de horas invertidas: 502

## **2. Evaluación**

### **2.1. Cantidad de pruebas unitarias muy bajas**

#### **No aprobado**

Es notable que no hay cultura de verificación, puesto que se percibe el reflejo inmediato demostrado ante plazos severos de suprimir las tareas de verificación.

El plan de implementación pasa a ser prioritario desplazando por completo el plan de verificación y validación, seguramente esto se deba a que el producto del plan de implementación tiene mayor visibilidad por el cliente.

Es de suponer que el cliente este más conforme ante nueva funcionalidad integrada que a la correctitud o perfeccionamiento de lo ya hecho (aunque este punto es altamente debatible).

En una opinión personal, como responsable de verificación concluyo que ante una situación de poco tiempo, justamente se requieren más que nunca invertir el 20% del esfuerzo preestablecido para verificación, puesto que esta área ya a demostrado no solo ser fundamental sino también apremiar en ahorro de tiempo por retrabajo.

### **2.2. No utilización de TDD**

#### **Aprobado**

Otra resolución aplicada a este proyecto fue no incorporar la disciplina "codificación dirigida por pruebas", si bien este medio logra evitar que uno se autovalide inconscientemente lo que pretendió codificar, TDD requiere un entrenamiento de tiempo prolongado no aplicable al presente proyecto.

### **2.3. Tecnologías disponibles no usadas a pleno**

#### **Aprobado con observaciones**

En otro enfoque, las herramientas planificadas han sido utilizadas con éxito por lo que se considera un acierto pese a que no todas las tecnologías de testing fueron aplicadas.

Como ya fue mencionado Mock Services requerían en la mayoría de los casos un esfuerzo no justificado, no así el descarte total se considera una medida acertada.

### **2.4. Puesta en marcha del Sistema de Integración Continua**

#### **Aprobado**

Por la capacidad de automatismo, la minuciosidad de sus tareas y la posibilidad de un sistema de notificaciones inmediatas y personalizadas bajo un contexto de trabajo en equipo. Se considera un acierto su inclusión en BeatIt!.

El trabajo generado por este sistema equivale al esfuerzo de un integrante extra en el equipo como gran conclusión.

## **3. Informe final de las métricas de Pruebas Cubiertas**

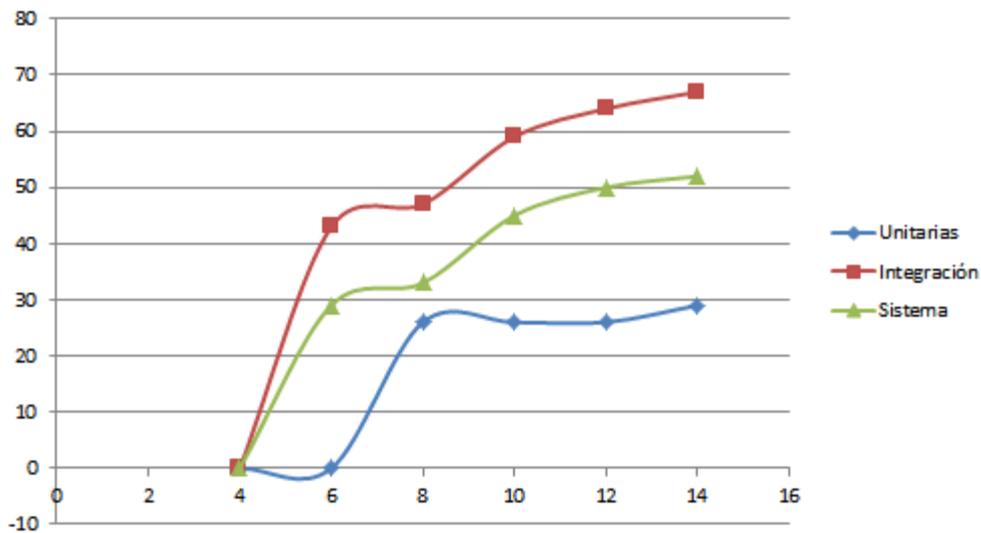
Comparativa de implementación entre ambas plataformas.

Dentro del mismo alcance, Windows Phone ha demostrado mayor eficacia, dado que a mitad del plazo un integrante de este equipo migró para sumarse al equipo de implementadores de Android. Resultando una gran diferencia lo logrado de mano de

tres integrantes o cinco. Interpretamos que lo sucedido se debe al mayor retrabajo que tuvo la tecnología Android. Estamos ante las métricas del proceso completo y es evidente que el camino recorrido por Android fue más accidentado, perdiendo en horas de retrabajo el buen índice que mantenía de funcionalidad liberada por horas de implementación.

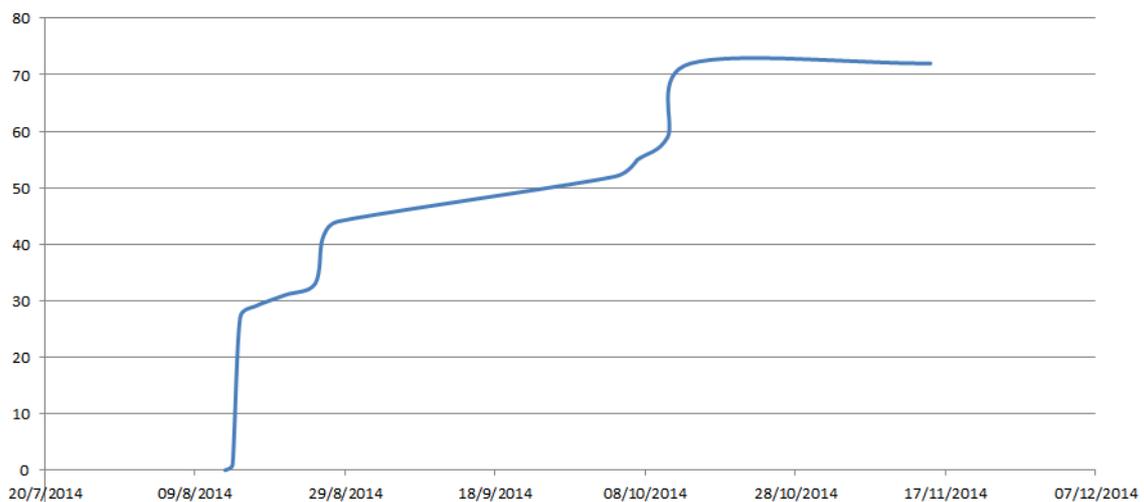
Windows Phone tuvo un menor índice de liberación de nuevas funcionalidades por hora pero en un recorrido más seguro puesto que presentó 30% menos de defectos de software, así pues con más holgura de plazos presentó un nivel superior de cobertura de código por pruebas y un feedback más positivo tanto del director de proyecto como del propio cliente.

### 3.1. Planificado total de todas las pruebas

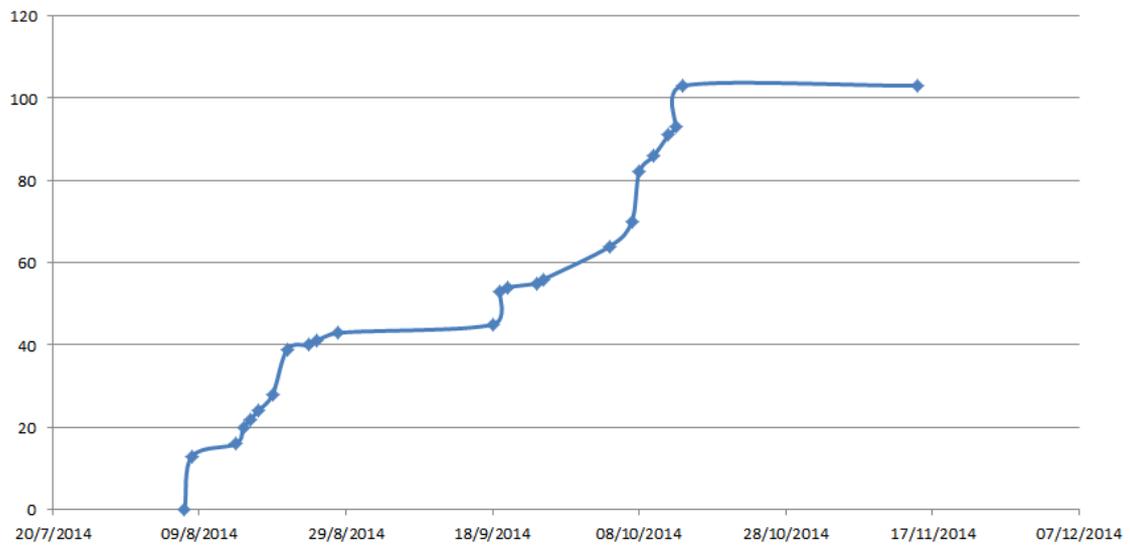


### 3.2. Pruebas de sistema reales

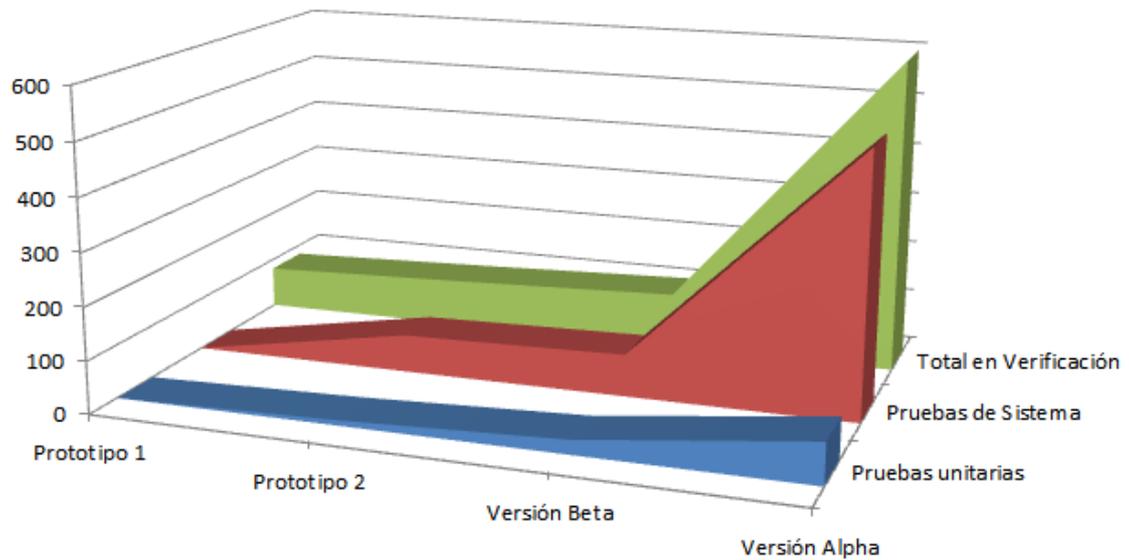
#### 3.2.1. Windows Phone



#### 3.2.2. Android



### 3.3. Evolución del tiempo invertido en verificación



Cabe mencionar que el tiempo total de verificación incluye el tiempo de pruebas de integración y de verificación de documentos.

Asimismo cada curva contempla el tiempo respectivo de planificación, documentación y demás tareas.