**RubyTrick**

**Acta de Reunión de Requerimientos**

**Versión 1.0**

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 31/08/2014 | 1.0 | Primera reunión | Pablo Da Costa Porto |
| 31/08/2014 | 1.1 | Revisión de SQA | Santiago Gómez |
|  |  |  |  |
|  |  |  |  |

**Fecha:** 27/08/2014

**Responsables:** Sebastián Melgar, Pablo Da Costa Porto.

**Participantes:** Andreas Fast – *Product owner*.

Grupo 3 – Desarrolladores.

**Índice**

[**RubyTrick 1**](#id.gjdgxs)

[**Acta de Reunión de Requerimientos 1**](#id.30j0zll)

[**Versión 1.0 1**](#id.1fob9te)

[**Historia de revisiones 1**](#id.3znysh7)

[**Temas Tratados : 2**](#id.2et92p0)

[Funcionalidades y características del producto : 2](#id.tyjcwt)

[Tareas para la siguiente reunión : 3](#id.1t3h5sf)

# Temas Tratados :

El primer tema a tratar de la reunión fue la determinación de las funcionalidades y características del producto, tarea realizada por el equipo y el *Product owner* en conjunto, mediante un método planteado por este último llamado *Impact mapping*. Éste método consiste en un esquema en el que en el centro se posiciona a los objetivos principales del producto; luego, se le agrega una adyacencia por cada funcionalidad o carecterística que servirá para cumplir esos objetivos al formar parte del producto. Cada una de estas adyacencias se puede ir refinando al agregarle nuevas adyacencias. La idea es que las hojas de este diagrama tengan un nivel de especificidad suficiente para representar las historias de usuario del *backlog*.

Una vez cerrado el primer tema, el *product owner* prosiguió a encargarnos tareas para la siguiente reunión.

## Funcionalidades y características del producto :

El *product owner* planteó como principales objetivos lograr una herramienta de aprendizaje a través de lo lúdico y que además debe ser efectivo para retener a los usuarios. Se habló, siempre desde la perspectiva de una interfaz web, de la necesidad de una página de inicio atractiva (*landing page*), que invite a los visitantes a registrarse o iniciar sesión. Las cuentas de usuario deberán estar integradas con Facebook y Twitter, permitiendo el registro a través de estas redes sociales, además del registro usual a través de e-mail. Los usuarios tendrán sus opciones usuales de gestión de cuentas (*login/logout*, ver perfil, modificar datos, agregar/aceptar/borrar amigos, etc.). Una vez registrados, se les invitará a realizar un tutorial o a empezar a jugar.

La aplicación le permitirá a cada usuario gestionar equipo de fútbol, idear estrategias para éste y jugar partidos y campeonatos contra otros usuarios y (posiblemente) equipos del sistema.

El usuario contará con un equipo inicial, al cual con el desarrollo del juego podrá transferir, entrenar y despedir jugadores. Con este equipo podrá jugar partidos amistosos y campeonatos contra otros equipos. Los campeonatos podrán ser tanto oficiales (organizados por el sistema) como organizados por los usuarios.

Cada jugador tendrá un conjunto de características (a determinar), de las cuales algunas podrán ser mejoradas con entrenamiento.

Las estrategias para cada partido serán escritas en Ruby por los usuarios. Estas pueden ser guardadas para ser reutilizadas.

Una funcionalidad importante de la aplicación sera un motor de simulación de partidos, que a partir de los equipos y las estrategias de cada partido deberá generar la simulación completa del partido, la cual deberá poder ser visualizada gráficamente por los usuarios. El diseño de este motor será el que determine las bases para la generación de estrategias. El cliente explicitó que dejará el diseño de este motor en manos del equipo, pero que aclaró que espera que alcance un nivel de complejidad un poco mayor que el de una aplicación similar ya existente llamada RubyGoal.

## Tareas para la siguiente reunión :

El *product owner* dejó al equipo ciertas tareas para la siguiente reunión. Éstas son:

* **Plan de capacitación en tecnologías/metodologías/herramientas**: esto es la planificación de cómo se capacitará al grupo para el desarrollo de la aplicación.
* **Propuestas de nombre del producto.**
* **Escribir las historias visualizadas hasta el momento.**
* **Iterar el impact map inicial**: refinamiento del esquema y visualización de historias.
* **Investigar/analizar/documentar funcionalidades de RubyGoal**: RubyGoal es una aplicación *open source* muy similar al producto a desarrollar y su análisis puede ser de ayuda..
* I**nvestigar/analizar/documentar funcionalidades de otros productos**: hay varios productos ya existentes que implementar algunas características del nuestro. Ejemplos son Hattrick y Football Manager.
* **Propuesta visual de simulador de partido.**
* ***Setup* de ambiente y usuarios GitHub**: esto es el *setup* del sistema de SCM.