

# Plataforma de gamificación

Iván Martín Maseda

**Resumen**—La utilización de mecánicas de juego en los negocios es algo cada vez más habitual. En este proyecto se ha desarrollado una plataforma de soporte a la aplicación de estrategias de gamificación sobre cualquier mercado. La plataforma, es una herramienta web open source que permite a la misma persona encargada de diseñar la estrategia de gamificación, aplicarla sobre un conjunto de usuarios.

**Paraules clau**—Plataforma de gamificación, webapp, MongoDB, schemaless database, mecánicas de juego, estrategia de gamificación, motivación de la interacción, compromiso de clientes.

**Abstract**—The use of game mechanics in business is something increasingly common. This project has developed a platform to support the implementation of gamification strategies for any market. The platform is an open source web tool that allows the person in charge of designing the gamification strategy, to apply it to a set of users.

**Index Terms**— Gamification platform, webapp, MongoDB, schemaless database, game mechanics, gamification strategy, motivation of interaction, customer engagement.

## 1 INTRODUCCIÓN

Gamificación es el uso de la mecánica del juego en otros contextos, de este modo, acabar atrayendo a los usuarios a la solución de problemas como se muestra en la figura 1. La gamificación se ha estudiado y aplicado en varios ámbitos, para mejorar la participación de los usuarios, el ejercicio físico, retorno de la inversión, calidad de los datos y el aprendizaje. Los estudios realizados sobre la gamificación detectan efectos positivos sobre los usuarios [10].

Este proyecto consiste en desarrollar una plataforma open source que permita que la misma persona que diseña la estrategia de gamificación, pueda llevarla a cabo. La plataforma facilita la aplicación de estrategias de gamificación a problemas concretos, como por ejemplo, asignación de premios de cualquier formato a los usuarios o clientes de una página web.

## 2 OBJETIVOS

El objetivo principal de este proyecto es desarrollar por completo la plataforma que permita al diseñador de estrategias de gamificación poder aplicar la misma sin precisar del soporte de un técnico informático.

Ésta constará de tres servicios mediante los cuales se podrá definir una estrategia concreta (configuración), indicar las acciones realizadas por los usuarios (tracking) y analizar el comportamiento de éstos, con el fin de valorar el resultado de aplicar la estrategia definida (analytics).

- E-mail de contacto: martin.maseda@gmail.com
- Mención realizada: Computació
- Trabajo tutorizado por: Jordi Gonzalez Sabaté (CVC)
- Tutor de empresa: Àngela Fàbregues (WWWOWWW)
- Curs 2013/14

FIGURA 1  
MECÁNICA DE LA GAMIFICACIÓN

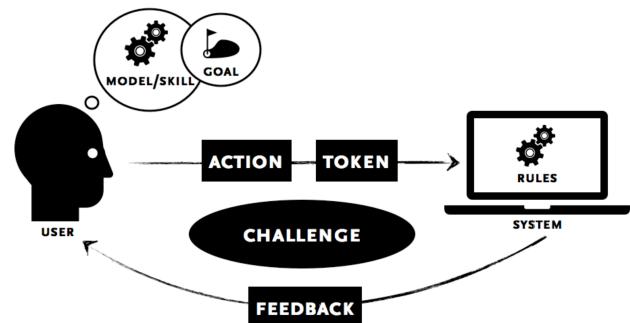


Diagrama representando la mecánica que utiliza un diseñador de estrategias de gamificación. [15]

**Configuración:** El objetivo de este servicio es plasmar la estrategia de gamificación en la plataforma. Las funcionalidades asociadas a este servicio son algunas como la definición de actividades, definición de reglas e inserción de posibles recompensas dentro de la plataforma.

**Tracking:** El objetivo de este servicio es mantener actualizada la información dentro de la plataforma. Los usuarios utilizarán una aplicación externa para informar de usuarios que hayan realizado una determinada acción o recompensas que ya hayan sido entregadas a los usuarios. Una funcionalidad imprescindible asociada a este servicio es obtener la información sobre qué usuarios deben recibir una recompensa.

**Analytics:** El objetivo de este servicio es el análisis de las actividades realizadas por los usuarios. Éste permitirá al administrador de la plataforma comprobar los resultados que se están obteniendo a partir del uso de la estrategia, y extraer conocimiento del comportamiento de los usuarios.

Existe una interfaz gráfica de usuario para el administrador desde la cual es posible acceder a los diferentes servicios de la plataforma. Por otro lado, también será posible acceder a las funcionalidades de los servicios conectando directamente con el controlador mediante una aplicación cliente.

### 3 ESTADO DEL ARTE

El número de empresas dedicadas a la implantación de plataformas de gamificación está creciendo. Solo en los estados unidos se han creado 76 empresas en los últimos 10 años. Las empresas líderes del mercado son actualmente bunchball, eMee y Enterprise Gamification.[16]

Actualmente existen 5 plataformas open source [17] que permiten aplicar gamificación. Cada una de estas plataformas se ha centrado en permitir utilizar alguna de las funcionalidades utilizadas por los sistemas de gamificación. No existe ninguna plataforma open source que permita implantar una estrategia de gamificación sin haber predefinido el tipo de recompensa sobre el que trabajar.

Las técnicas de gamificación más modernas utilizan en la estrategia elementos como puntos, emblemas y tableros de clasificación entre grupos concretos de usuarios. Estos elementos de la gamificación aprovechar nuestra adicción a la competencia y la recompensa, y lo utilizan para animar a los usuarios a realizar ciertas acciones. Este es el concepto central de gamificación: cambiando la forma de pensar acerca de la conducta, la gamificación puede cambiar los hábitos de los usuarios.

### 4 METODOLOGÍA

La metodología utilizada sigue el método de bottom up a partir de la base de datos. El proyecto está definido en 4 clases conceptuales muy específicas y se procede a realizar una iteración completa para cada una de las clases. Éstas se definen al inicio del proyecto y se enlazan a lo largo del mismo, para formar clases más completas.

A medida que se avanza en el proceso de desarrollo del proyecto, se implementan de forma incremental las funcionalidades necesarias para comunicar los diferentes componentes. Cada uno de los componentes puede precisar de información proveniente de otros subapartados de la aplicación, éstos han sido simulados hasta que cada uno de los apartados ha sido puesto en funcionamiento.

Para analizar las necesidades de esta plataforma se realiza en este apartado un análisis de los requisitos del sistema. El análisis de requisitos es el proceso de estudio de las necesidades de los usuarios, para llegar a una definición de los requisitos de la plataforma. Por tanto, configura una lista de los puntos que debe cumplir el sistema para satisfacer la especificación.

Una iteración completa sobre una clase consistiría en la definición de los conceptos que deben estar incluidos en ella (análisis), posteriormente tomar las decisiones de diseño asociadas a esos conceptos (diseño), seguido de la implementación de cada una de las funcionalidades (implementación). Por último, se realiza una fase de testing

que deberá comprobar el correcto funcionamiento de todos los pasos anteriores. Para cada clase conceptual será realizado este proceso iterativo completo.

## 5 ANÁLISIS DE LA PLATAFORMA

En esta sección describimos el proyecto a partir de sus requisitos, modelo de dominio y casos de uso.

### 5.1 Requisitos

Los requisitos de la plataforma son:

- La plataforma deberá poder ser utilizada por un administrador sin conocimientos técnicos de programación. Deberá contar con una interfaz visual que permita a este administrador configurar la estrategia de gamificación.
- Existirá una aplicación cliente externa que conectará con el servicio de tracking de la plataforma para notificar las acciones realizadas por los usuarios de la aplicación y añadir nuevos usuarios.
- La plataforma proporciona información sobre qué recompensas debe recibir cada usuario en función de la estrategia definida.
- La plataforma lleva el control de las recompensas entregadas y pendientes.
- Deberá mantenerse un control del stock de las recompensas que lo precisen, asegurando que nunca pueda asignarse a un usuario una recompensa inexistente.

### 5.2 Modelo de dominio

La plataforma de gamificación está formada por 4 conceptos principales, estos conceptos son los siguientes:

- **User:** Un usuario es alguien que realiza acciones en una aplicación externa y del cual se registra en nuestra plataforma el nombre, apellidos, contraseña y más información adicional, como el histórico de acciones realizadas.
- **Action:** Una acción identifica cada una de las posibles actividades que puede realizar un usuario.
- **Rule:** Una regla es el elemento principal de una estrategia de gamificación. Las reglas las define el diseñador de la estrategia, que es el administrador de la plataforma. Relaciona un conjunto de acciones a realizar con una determinada recompensa en caso de cumplirse la condición especificada.
- **Reward:** Una recompensa es un premio que recibe un usuario en caso de cumplir una determinada regla.

Las relaciones entre estos conceptos pueden expresarse del siguiente modo: Un usuario podrá realizar una de las acciones que ha sido definida en la plataforma y la información relativa a este evento será añadida al histórico de este usuario, por otro lado, podrá recibir recompensas por sus acciones. Las acciones realizadas por los usuarios contienen información acerca de qué acción ha sido realizada y deben ser fácilmente identificables por el administrador. Las reglas serán utilizadas por el administrador para plasmar la estrategia de gamificación, relacionando mediante una regla lógica, las acciones con las

recompensas. Por último, las recompensas son los incentivos que deberán conseguir que los usuarios de la plataforma tengan un comportamiento que siga la intención de la estrategia. En caso de cumplirse alguna de las reglas, la plataforma indicará qué recompensa debe entregarse al usuario.

#### Ejemplo:

- El usuario U realiza las acciones A y B.
- Una regla X determina que en caso de que un usuario realice las acciones A y B, deberá entregarse una recompensa C. ( $A \wedge B \rightarrow C$ )
- El usuario U recibe la recompensa C.

FIGURA 2  
DIAGRAMA DE CASOS DE USO

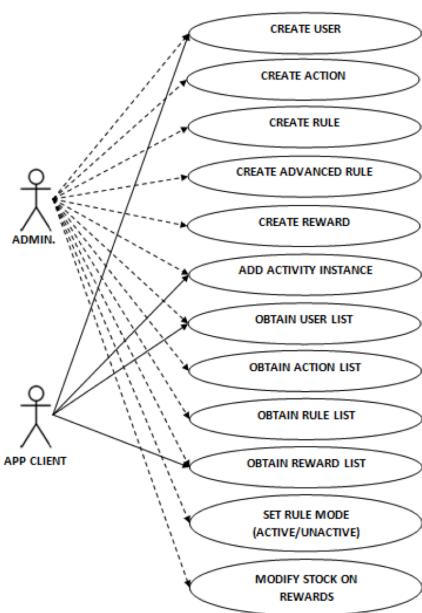


Diagrama representando los diferentes actores que interactúan con la aplicación.

### 5.3 Casos de uso

Existen dos tipos de actores, el administrador, que es el diseñador de la estrategia, y la aplicación externa (APP CLIENT), que es la que informa a la plataforma sobre las acciones que realizan los usuarios. En el diagrama de casos de uso de la figura 2, se presenta qué funcionalidades puede realizar cada actor.

El administrador tiene acceso a todas las funcionalidades de la plataforma. Por otro lado, APP CLIENT, solamente puede informar de las acciones realizadas por los usuarios y conocer el estado actual de la plataforma.

Las funcionalidades disponibles (casos de uso) son:

- CREATE USER: Los dos actores pueden introducir información sobre un nuevo usuario y opcionalmente puede introducir información extra.
- CREATE ACTION: El administrador define una acción y un código único la identificará a partir de ahora.

- CREATE RULE: El administrador define la regla lógica en función del número de veces que se ha realizado una acción y el conjunto de recompensas que deberá recibir el usuario al cumplir la condición definida anteriormente.
- CREATE ADVANCED RULE: El administrador podrá introducir reglas lógicas complejas utilizando un intérprete de Python. Existe la posibilidad de utilizar todos los comandos lógicos que incluye este lenguaje.
- CREATE REWARD: El administrador define una recompensa.
- ADD ACTIVITY INSTANCE: Los dos actores pueden indicar que un usuario ha realizado una acción concreta.
- OBTAIN USER LIST: Los dos actores pueden recibir la información de cuáles son todos los usuarios en la plataforma.
- OBTAIN ACTION LIST: Los dos actores pueden recibir la información de todas las acciones que han sido creadas en la plataforma.
- OBTAIN RULE LIST: Los dos actores pueden recibir la información de todas las recompensas que han sido creadas en la plataforma. Conjunto con la información básica de la regla, la plataforma también retorna el estado de cada una de las reglas (Active/Unactive).
- OBTAIN REWARD LIST: Los dos actores pueden recibir la información de todas las recompensas que han sido creadas en la plataforma. Junto con la información básica de la recompensa, la plataforma también informa de la cantidad de stock disponible de cada una de las recompensas.
- SET RULE MODE (ACTIVE/UNACTIVE): El administrador puede modificar el estado de cada una de las reglas de la plataforma.
- MODIFY STOCK ON REWARDS: El administrador podrá modificar la cantidad de stock disponible de cada una de las recompensas.

FIGURA 3  
ARQUITECTURA MODEL VIEW CONTROLLER

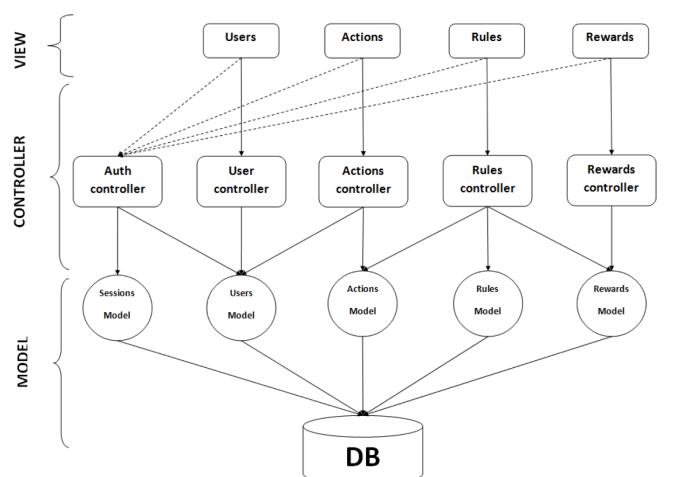


Diagrama representando la arquitectura software separando M-V-C.

## 6 DISEÑO DE LA PLATAFORMA

A partir de los requisitos analizados en el apartado anterior se ha optado por seguir una arquitectura software Model View Controller. Siguiendo el modelo de dominio, en este apartado se presenta el modelo de datos utilizado.

### 6.1 Arquitectura software

Model View Controller es un patrón de desarrollo que permite separar la gestión de los datos de la visualización de los mismos mediante el uso de controladores. Se han utilizado 5 controladores, de los cuales 4 controladores son los dedicados a gestionar los cuatro conceptos principales de la plataforma (User, Action, Rule, Reward). Por otro lado existe un último controlador encargado de la gestión de la autenticación para acceder a la plataforma. El modelo total de la plataforma está dividido en 4 modelos específicos para cada concepto. Esta separación se ha realizado para gestionar las funcionalidades específicas de cada concepto por separado. Estos modelos específicos conectan directamente con la base de datos y trabajan directamente sobre el documento almacenado en formato JSON.

FIGURA 4  
DIAGRAMA DE FLUJO

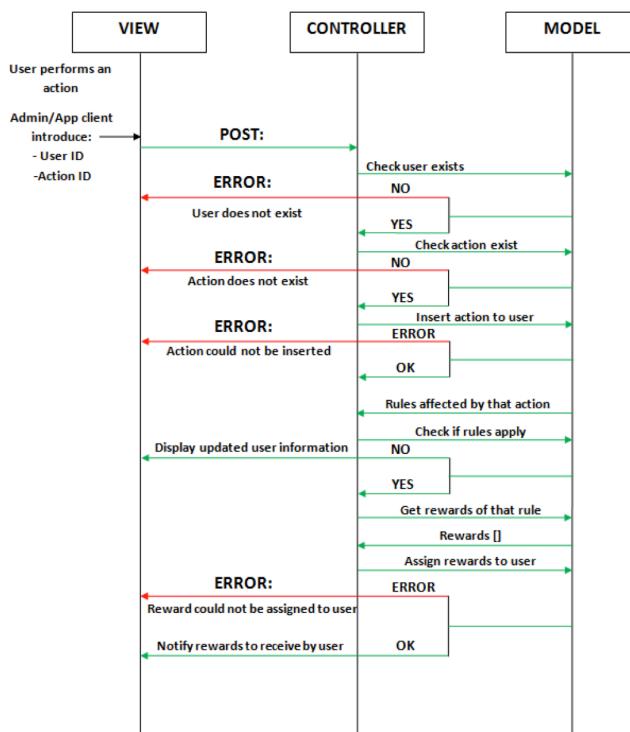


Diagrama representando el flujo de la aplicación al recibir la información indicando que un usuario ha realizado una acción.

En el diagrama de la figura 3 se puede observar como se ha estructurado la plataforma siguiendo la arquitectura MVC. La plataforma es dividida en tres partes interconectadas, a fin de separar las representaciones internas de información. El componente principal, el modelo, se

componen de datos de la aplicación, se podrá acceder a ellos mediante los modelos específicos que nos devolverán la información en el formato adecuado para trabajar con ellos y mostrárselo al usuario.

Una vista puede ser cualquier representación de salida de información, como una tabla o un diagrama en un explorador o una aplicación externa. Múltiples vistas serán utilizadas para acceder a la información, esto permite al administrador tener un control de toda la información fácil y rápido. El controlador, acepta ciertas entradas y las convierte en comandos para el modelo o la vista.

Con el fin de presentar un ejemplo dentro de la interacción entre M, V y C, se presenta un diagrama de flujo en la figura 4. En el diagrama se puede observar el flujo de ejecución cuando un actor indica que un usuario ha realizado una acción. Se puede observar en el diagrama de flujo que vista y modelo están completamente aislados. Para la comunicación entre estos dos componentes se utiliza en todo momento los controladores, que permiten estructurar la información y minimizar la cantidad de información transferida a la vista.

La arquitectura, además de dividirse en modelo, vista y controlador, se divide por clases según el tipo de funcionalidad que proporcionan. La funcionalidad de creación de usuario y de obtención de la lista de recompensas entregadas a un usuario se encuentra incorporada dentro del UserModel. También será posible indicar que un usuario ha realizado un acción concreta mediante este modelo específico. La creación de nuevas acciones y el tracking de las mismas está implementada en ActionModel. La creación de nuevas reglas mediante el método simple o el complejo será accesible a través del modelo específico RuleModel. Mediante el RuleModel también es posible el control del estado de las reglas específicas, es posible tanto activar como desactivar cada una de las reglas específicamente. La asignación de recompensas a las reglas, precisa del modelo específico RewardModel, al igual que la ceración de las propias recompensas. Las recompensas, al tratarse de objetos físicos, tienen incorporado un control de número de unidades. Este número (stock) podrá ser modificado también a partir de las funcionalidades del modelo RewardModel. Cabe destacar que todos las clases conceptuales pueden ser listadas y mostradas en la vista específicamente diseñada para cada una de ellas. Se encuentra disponible la funcionalidad que retorna toda la lista de componentes existentes en modelo específico correspondiente a la clase en concreto.

### 6.2 Estructura de datos

Se define a continuación la estructura de datos. Se ha seleccionado un diagrama de clases que está mayoritariamente definido por los conceptos de la plataforma. El diagrama de clases se ha definido siguiendo el diagrama de la figura 5.

Las relaciones entre los componentes de la plataforma no son dependientes entre sí, a excepción de las reglas, tal y como se puede observar en el diagrama de la figura 5. La razón por la que esto ocurre, es porque en la plataforma deben poder registrarse usuarios previamente

a la activación de las reglas y de este modo preparar la plataforma para empezar a operar. Por otro lado tanto las acciones como las recompensas tampoco son dependientes entre si. El administrador puede introducir todas las recompensas que tiene disponibles sin haber definido aún ninguna de las reglas que acabarán definiendo la estrategia. Por último, sí que será imprescindible que hayan sido definidas previamente tanto las acciones como las recompensas para que puedan ser utilizadas en la definición de una regla.

## 7 IMPLEMENTACIÓN

El proyecto completo ha sido programado utilizando el lenguaje de programación Python 2.7[11], a excepción de las vistas que han sido desarrolladas con HTML y CSS simple. La implementación ha sido desarrollada siguiendo la metodología bottom up tal y como se ha especificado en el correspondiente apartado, se han utilizado diferentes librerías para cada una de las partes de la implementación, que serán especificadas a continuación.

### 7.1 Base de datos

La base de datos de la plataforma ha sido programada utilizando MongoDB, tecnología de base de datos schemaless (noSQL) open source orientada a documentos de alto rendimiento. MongoDB se caracteriza por tener un alto rendimiento, alta disponibilidad y escalabilidad automática.

MongoDB se caracteriza por tener un alto rendimiento en las estructuras de datos embedadas que nos permitirá almacenar mucha información de cada una de las actividades de la plataforma. El alta disponibilidad de esta tecnología es debida a la replicación en replica sets que proporciona la tecnología de forma automática, cuenta con funcionalidades como recuperación automática, redundancia de datos y por lo tanto, como consecuencia, una alta disponibilidad de los mismos. Por último, la escalabilidad horizontal automática que proporciona MongoDB se consigue mediante lo que se conoce como sharding. Sharding es una de las funcionalidades bases de MongoDB, es un método que consiste en distribuir los datos almacenados en varias máquinas distintas. Utilizar este método permite mejorar el tiempo de acceso a los datos.

La plataforma de gamificación utiliza MongoDB como base de datos. Esta tecnología utiliza colecciones para organizar los documentos. En el caso de la plataforma se han utilizado 5 colecciones que únicamente están relacionadas mediante el identificador “\_id” para referenciarlas. Las 5 colecciones son las siguientes:

- USERS: Colección donde es almacenada la información relativa a cada uno de los usuarios. Los usuarios son identificados con un documento. La clave única, identificada como “\_id” representa el nombre de usuario.
- ACTIONS: Colección donde es almacenada la información relativa a cada una de las acciones. La clave única es el nombre que identifica la acción, aunque también se ha incorpora-

do una secuencia única de 5 dígitos para identificar la acción mediante una URL.

- REWARDS: Colección donde es almacenada la información relativa a cada una de las recompensas. La clave única es el nombre que identifica la recompensa.
- RULES: Colección donde es almacenada la información relativa a cada una de las reglas. La clave única es el nombre que identifica la regla.
- SESSIONS: Colección donde es almacenada la información relativa a cada una de las sesiones. La clave única es generada aleatoriamente cada vez que un usuario inicia sesión en la plataforma, esa secuencia aleatoria es única pero puede ser utilizada en la próxima sesión en caso de que coincidiera.

FIGURA 5  
DIAGRAMA DE CLASES

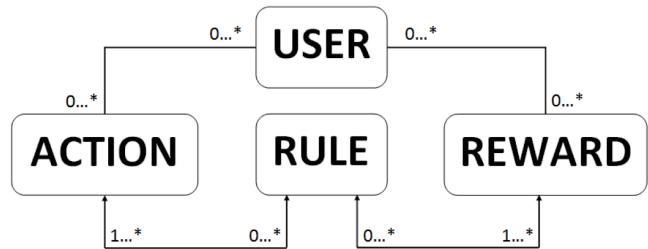


Diagrama representando las relaciones entre las diferentes clases conceptuales de la plataforma.

Como ya se ha indicado, MongoDB utiliza documentos para almacenar la información, estos documentos son almacenados en formato JSON. A continuación mostraremos un ejemplo muy característico para demostrar la forma en que MongoDB gestiona los documentos.

USER document example:

```

{
  "_id" : "Iván Martín",
  "performed_actions" : [{ "date" : ISODate(), "actionID" : "Buy new car" },
  "user_info" : {
    "Phone number" : "658432022",
    "Number" : "102",
    "Street" : "Major de Sarria",
    "Postal Code" : "08658"
  },
  "applied_rules" : [
    {
      "ruleID" : "For the car owner, welcome back",
      "date" : ISODate(),
      "rewards_given" : [ "The 20% discount", 10 ]
    }
  ],
  "received_rewards" : {
    "received_rewards_list" : [
      {
        "date" : ISODate(),
        "rewardID" : "The 20% discount"
      }
    ],
    "num_received_rewards" : 1
  },
  "password" : "83553306b.....e8b1353,OWYSd",
  "email" : "TFGivan@TGFivan.com"
}
  
```

Como se puede observar en el ejemplo, existe un cam-

po estándar en MongoDB que supone la clave única, este campo es el que se identifica con “`_id`” y en la plataforma es utilizado para hacer referencia a documentos que se encuentran en otras colecciones. Este campo es utilizado de forma similar al que suele utilizarse la clave externa en las bases de datos SQL convencionales. La diferencia con estos sistemas puede apreciarse en campos como el de “`user_info`”, en este campo utiliza la técnica conocida como embedding que consiste en introducir un segundo documento dentro de un campo del primero. Al poder incorporar documentos dentro de otros documentos la flexibilidad del sistema aumenta notablemente. Esto nos permite añadir nuevos campos dentro de ese segundo documento y no incorporar campos de los cuales se desconozca su valor.

## 7.2 Back end

El lenguaje del lado del servidor utilizado en la plataforma ha sido la librería estándar de Python 2.7 más las funcionalidades obtenidas de la librería pyMongo. PyMongo son los drivers de acceso a la base de datos MongoDB.

La razón por la que ha sido seleccionado Python como lenguaje servidor es porque es una lenguaje de programación eficiente con la gestión de estructuras de alto nivel y muy sencillo para el manejo de objetos. Por último, Python ha sido seleccionado por la existencia de los diccionarios de Python que facilitan mucho la gestión de documentos en formato JSON. Por otro lado, en relación a los drivers utilizados para acceder a MongoDB desde Python, existen varias alternativas, pero los drivers oficiales recomendados por la MongoDB University han permitido un rápido aprendizaje de las herramientas y han facilitado el desarrollo de la plataforma.

Existen dos procesos principales que se llevan a cabo en el lado del servidor. En primer lugar el proceso descrito en la figura 4, encargado de gestionar la entrega de recompensas a los usuarios una vez realizan una acción. Este es el proceso principal de la plataforma y funciona del siguiente modo: Cuando un usuario realice una acción, la plataforma comprobará todas las reglas a las que esa acción afecte. Para toda la lista de reglas deberá comprobar que la regla esté activa, que no haya sido aplicada anteriormente sobre ese usuario y por último comprobar si cumple con todos los requisitos. En caso de que todas las condiciones se cumplieran, la plataforma deberá obtener del la regla la lista de recompensas de las que ese usuario es merecedor y asignárselas. En este punto se inicia el segundo proceso que controla la plataforma. Para la entrega de recompensas, existe un proceso dedicado a comprobar si para cada una de las recompensas existe suficiente stock. La plataforma deducirá el stock de cada recompensa cada vez que sea otorgada a un usuario. En caso de existir suficiente stock para recompensar al usuario, el administrador recibirá un correo electrónico indicando el usuario, cuál ha sido la regla aplicada, y la recompensa que le ha sido otorgada. En caso de que no hubiera suficiente stock de alguna de las recompensas que deben entregarse al usuario, se otorgarían todas las recompensas de ese tipo disponibles a ese usuario hasta

finalizar el stock. Cuando una recompensa finaliza su stock, la plataforma desactiva automáticamente todas aquellas reglas que la consumían, de ese modo la plataforma nunca podrá intentar entregar una recompensa de la cual no exista stock. El administrador recibirá una notificación especial cuando las existencias de una recompensa en concreto hayan finalizado y en caso de querer continuar ofreciendo esa recompensa, deberá introducir más stock en la plataforma y volver a activar la regla correspondiente.

## 7.3 Front end

En el front-end se ha utilizado Python directamente sobre las plantillas HTML (.tpl) mediante Bottle [6] para mostrar la información recibida del controlador. Algunas de las funcionalidades JavaScript han sido obtenidas de fuentes como Bootstrap [12] o Foundation [13] para mostrar los resultados de forma adecuada.

Como Web Server Gateway Interface (WSGI), se ha utilizado Bottle que es un micro web framework para desarrollar en Python. Este framework tiene incorporada la compatibilidad con servidores http de desarrollo como Paste [14] que ha sido utilizado para gestionar varias conexiones de forma simultanea.

## 8 TEST

Cada una de las funcionalidades del proyecto tiene asignada una batería de test que pone a prueba cada una de las funcionalidades de las clases. Las pruebas realizadas comprueban tanto el correcto funcionamiento de la función, como entradas incorrectas para la misma. En caso de que una función de cualquiera de las clases de la plataforma reciba valores inapropiados, la plataforma debe continuar con su comportamiento habitual. Se incluyen pruebas en las que se generan errores a propósito para comprobar el control de errores de la aplicación. Cada una de las baterías de pruebas pone a prueba cada componente por separado. La batería de test final comprueba no solamente errores de funciones concretas, sino que también comprueba conjuntos de funciones con el objetivo de simular el funcionamiento final de la plataforma.

## 9 ESTADO FINAL DE LA IMPLEMENTACIÓN

La plataforma ha sido desarrollada con todas las funcionalidades definidas en los objetivos iniciales. Actualmente es posible modificar todos los contenidos que los servicios definidos pueden necesitar. El controlador principal tiene incorporada la llamada a cada una de las clases, permitiendo acceder a todas las funcionalidades definidas en los servicios de la figura 2:

- **CREATE USER:** Es posible añadir nuevos usuarios a la plataforma, esta funcionalidad cuenta con una interfaz visual para llevarse a cabo. Más funcionalidades ha sido desarrolladas para la versión no visual como añadir información extra que no ha sido definida en el formulario de inscripción.

- **CREATE ACTION:** Es posible crear nuevas Acciones. Existe una interfaz visual que permite realizar esta actividad mediante un formulario.
- **CREATE SIMPLE/ADVANCED RULE:** Es posible crear nuevas Reglas. Es posible crear nuevas reglas de 3 modos distintos. En primer lugar existe la forma visual simple desde la que podrán crearse reglas simples con estructuras lógicas "AND" entre las acciones. En segundo lugar se ha desarrollado una interfaz visual para usuarios avanzados donde es posible introducir reglas lógicas más complejas mediante la utilización de los comandos lógicos de Python. Por último, pueden utilizarse los dos métodos definidos anteriormente, accediendo al sistema directamente al conectar al controlador desde una cliente externo a la aplicación. Se ha incorporado la posibilidad de desactivar las reglas una vez creadas, de este modo puede permitirse la finalización de algún sector de la estrategia definida.
- **CREATE REWARD:** Es posible crear nuevas recompensas. Las recompensas que serán finalmente entregadas al usuario pueden ser definidas para que pueda realizarse un control de stock sobre las mismas. El sistema nunca entregará una recompensa de la que no exista stock y en caso de finalizar el stock de una recompensa se desactivarán temporalmente las reglas que estuvieran entregando esa recompensa ya que los usuarios no podrán recibirla.
- **ADD ACTIVITY INSTANCE:** Es posible indicar que un usuario ha realizado una acción. Una vez esta información es introducida a la plataforma se activa el proceso definido en la figura 4, donde se comprueba si la nueva información introducida permite aplicar alguna regla de las activas en el momento de introducir la información.
- **OBTAIN USER LIST:** Es posible recibir una lista de usuarios. La interfaz visual de esta funcionalidad ha sido implementada y permite tener una visión resumida de cada usuario.
- **OBTAIN ACTIONS:** Es posible recibir una lista de acciones. La interfaz visual de esta funcionalidad ha sido implementada y permite identificar cada una de las acciones con su nombre y descripción. Ha sido incorporado un contador que indica el número de veces que esa acción ha sido realizada por cualquier usuario.
- **OBTAIN RULE LIST:** Es posible recibir una lista de reglas. La interfaz visual de esta funcionalidad ha sido implementada y permite identificar cada una de las reglas a partir de su nombre y descripción. Las informaciones de las reglas incluye la lógica que aplica sobre las acciones y las recompensas que serían otorgadas al usuario que cumpla con la condición definida.
- **OBTAIN REWARD LIST:** Ha sido desarrollada la funcionalidad para obtener un overview de las recompensas posibles. Por otro lado existe también la posibilidad de mostrar que recompensas han sido entregadas a los usuarios y cuál ha sido la regla en concreto que las ha activado. En el momento en que una regla se aplique sobre un usuario y se confirme su cumplimiento, el administrador de la plataforma será notificado y deberá realizar la entrega de las recompensas pertinentes.
- **SET RULE MODE (ACTIVE/UNACTIVE):** Ha sido desarrollada la funcionalidad para permitir al administrador activar o desactivar una regla en concreto. En caso de que un usuario realizara una acción que le permitiera cumplir con esta regla, si el administrador la mantiene desactivada, no se le otorgarían las recompensas pertinentes.
- **MODIFY STOCK ON REWARDS:** Han sido desarrolladas las funcionalidades necesarias para la modificación de stock, el stock de una recompensa puede ser incrementado en el momento en que el administrador reciba la recompensa en concreto y será decrementado en el momento en que una de las reglas entregue a un usuario esa recompensa en concreto.

## 10 MANUAL DEL PRODUCTO FINAL

En esta fase se explicará el funcionamiento de la plataforma de modo que una persona sin conocimientos técnicos pueda acceder a ella de manera satisfactoria.

### 10.1 Funcionamiento general de la plataforma:

Dentro de la plataforma se encuentran las diferentes acciones que un usuario puede realizar. El administrador o cliente externo será el encargado de notificar a la plataforma que un usuario ha realizado una acción. Por otro lado, la plataforma permite definir unas ciertas reglas encargadas de relacionar un conjunto de acciones con unas recompensas. Finalmente, según el tipo y la cantidad de acciones realizadas, el usuario recibirá una recompensa determinada.

El administrador de la plataforma tendrá la posibilidad de definir una estrategia de gamificación, para ello, podrá utilizar las reglas que permitirán modificar el interés de los usuarios en realizar una acción u otra.

### 10.2 Cómo registrar un usuario:

Para registrar un usuario, la persona encargada o administrador deberá hacerlo mediante la dirección e-mail de la persona en cuestión. Una vez creada la cuenta, deberá registrarse los datos e información adicional, la cual quedará registrada en la base de datos de la plataforma.

### 10.3 Cómo registrar una acción:

Para registrar una acción, la persona encargada o administrador deberá hacerlo mediante un nombre de acción que identificará esa acción dentro de la plataforma a partir de ese momento. Una acción podrá tener incorporada una descripción que facilite la especificación de sus características. Las actividades deberán representar la interacción de los usuarios con la plataforma y el administrador deberá cerciorarse de inscribir todas aquellas actividades que puedan definir la actuación de un usuario. Es imprescindible que los usuarios finales sean notificados de acciones que deben realizar para avanzar dentro de la plataforma, de este modo se motivará de forma intrínseca su forma de actuar.

## 10.4 Cómo registrar una recompensa:

Para registrar una recompensa, la persona encargada o administrador deberá hacerlo mediante un nombre de recompensa que identificará esa recompensa dentro de la plataforma a partir de ese momento. Una recompensa podrá tener incorporada una descripción que facilite la especificación de sus características. En el momento de introducir la nueva recompensa se pedirá al administrador de introduzca una cantidad de stock de la misma, la plataforma mantendrá un control para que no pueda asignarse recompensas a usuarios una vez terminado el stock. La cantidad de stock de una recompensa podrá ser incrementada más adelante, incluso durante el funcionamiento activo de la plataforma.

## 10.5 Cómo introducir reglas:

El objetivo de la plataforma de gamificación es poder plasmar la estrategia de gamificación en un conjunto de reglas que permitan al administrador mantener un control de la asignación de recompensas. La plataforma consta de dos métodos de inserción de reglas. Para los administradores básicos, existe una interfaz visual simple, sobre la que se podrán crear reglas sencillas. Este primer método permite introducir reglas como la siguiente:

(Acción1 >= 4) and (Acción2 >= 10) -> Recompensa1

Esta regla implicaría que el usuario deberá realizar la Acción1 4 o más veces y la Acción2 10 o más veces para recibir la Recompensa1. Este método para la introducción de reglas es muy sencillo de utilizar y si la estrategia de gamificación del administrador se basa en estructuras lógicas que únicamente utilicen reglas AND, es muy útil.

Por otro lado la plataforma cuenta con método de inserción de reglas para administradores avanzados, donde el administrador puede utilizar los comandos lógicos de Python para definir reglas más complejas. En el editor de reglas avanzado, el administrador cuenta con un campo de texto donde podrá introducir las condiciones lógicas que la regla debe cumplir. Para utilizar este método el usuario debe saber que deberá rodear los nombres de las acciones con los caracteres “espacio”+“\_” (“\_”) y “\_”+“espacio” (“\_”) del siguiente modo “\_accion1\_”. De este modo el sistema identifica que es el nombre de una variable y en el momento de evaluar la expresión, modificará estos campos con el valor de el número de veces que cada usuario ha realizado la acción. Un ejemplo de éste método sería el siguiente:

( \_action1\_ <2) or ( \_action2\_ >10)

Esta regla implicaría que el usuario podrá recibir la recompensa asociada a la regla en caso de que haya realizado un máximo de 2 veces la action1 o un mínimo de 10 veces la action2. Utilizar este método para insertar las reglas de la plataforma permite combinaciones mucho más complejas e interesantes para tener en cuenta en el diseño de la estrategia de gamificación.

## 11 CONCLUSIÓN

Se ha desarrollado una plataforma que permite al diseñador de una estrategia de gamificación gestionar la entrega de recompensas a todos los usuarios registrados. La plataforma

permite registrar un conjunto de usuarios y definir qué acciones de estos usuarios quieren tenerse en cuenta. Por otro lado el administrador podrá diseñar la estrategia de gamificación, mediante un conjunto de reglas, que indicarán al sistema qué recompensa deberá entregarse a cada usuario en caso de cumplir la condición definida.

La plataforma ha sido diseñada teniendo en cuenta que en futuros proyectos podría incorporarse la posibilidad de analizar las tendencias de los usuarios en función de su comportamiento. Para el futuro análisis se ha incorporado información relativa a los tiempos de ejecución de cada funcionalidad, permitiendo analizar la información en cualquier momento futuro.

Dentro de las limitaciones, el proyecto se ha podido finalizar y está disponible en la URL: <http://rimel.wwwowww.me>, funcionando en el puerto 8300. Para poder realizar pruebas se ha dado de alta un usuario “Ivan Martin” con contraseña “TFGivanTFGivan”.

## 12 AGRADECIMIENTOS

El autor quiere agradecer a Jordi González Sabaté, a Àngela Fàbregues y a la empresa WWWOWWW por haber dirigido el proyecto y facilitado las herramientas necesarias para su desarrollo. Por otro lado se quiere agradecer a MongoDB University por haber impartido el curso de M101P: MongoDB for developers.

## 13 BIBLIOGRAFÍA

- [1] Foro de consultas de programación StackOverflow. <http://stackoverflow.com/>
- [2] Curso M101P: MongoDB for Developers de MongoDB University en marzo de 2014.
- [3] Sitio web de referencia sobre gamification en abril de 2014. <http://www.gamification.co/getting-started/>
- [4] Curso HTML & CSS de Codecademy en mayo de 2014. <http://www.codecademy.com/es/courses/web-beginner-en-HZA3b/0/1>
- [5] Como referencia para consultas sobre HTML, CSS y JavaScript entre los meses marzo y mayo de 2014. <http://www.w3schools.com/>
- [6] Como referencia para consultas sobre el framework bottle entre los meses marzo y mayo de 2014. <http://bottlepy.org/docs/dev/index.html>
- [7] Como referencia para consultas sobre MongoDB 2.4 entre los meses marzo y mayo de 2014. <http://docs.mongodb.org/v2.4/>
- [8] The Beginner’s Guide to Gamification en junio de 2014. <http://technologyadvice.com/gamification/smart-advisor/>
- [9] Open Source Gamification Platforms en junio de 2014. <http://www.42terabytes.com/open-source-gamification-platforms/>
- [10] Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does Gamification Work? – A Literature Review of Empirical Studies on Gamification. In proceedings of the 47th Hawaii International Conference on System Sciences, Hawaii, USA, January 6-9, 2014.
- [11] Como referencias para consultas sobre Python 2.7 entre los meses de marzo y mayo de 2014. <https://docs.python.org/2.7/tutorial/index.html>
- [12] Como fuente de plantillas para diseño con CSS entre los meses de marzo y mayo de 2014. <http://getbootstrap.com/css/>

- [13] Como fuente de plantillas para diseño con CSS entre los meses de marzo y mayo de 2014.  
<http://foundation.zurb.com/docs/css.html>
- [14] Como referencia de consultas sobre Paste entre los meses de marzo y mayo de 2014. <http://pythonpaste.org/>
- [15] Sebastian Deterding (2013). Skill Atoms as Design Lenses for User-Centered Gameful Design. Hans Bredow Institute for Media Research Warburgstr. 8-10 D-20354 Hamburg, Germany, April 27 – May 2, 2013.
- [16] Como referencia para la localización de compañías existentes dedicadas a la gamificación se ha utilizado esta fuente como referencia entre los meses de marzo y mayo 2014. <http://technologyadvice.com/gamification/smart-advisor/>
- [17] Como referencia para la localización plataformas open source ya existentes se ha usulizado esta fuente como referencia entre los meses de marzo y mayo 2014. <http://42terabytes.com/open-source-gamification-platforms/>