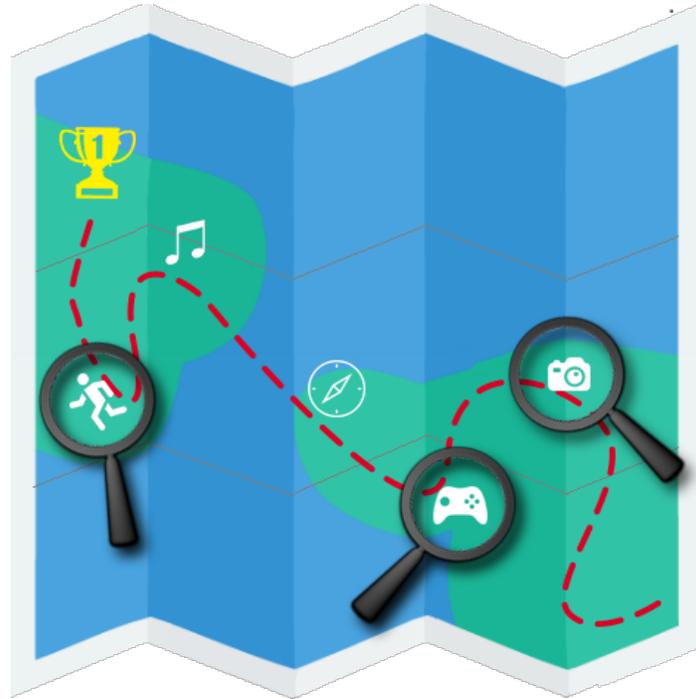


BeatIt!

Informe de verificación unitaria

Versión 2.2.2



Historial de Versiones

Fecha	Descripción	Versión	Autor
11/10/2014	Se realizaron prueba a componentes de negocio y no de interfaz gráfica	2.2.1	Joaquin Velazquez
12/10/2014	Revisión SQA	2.2.2	Pablo Olivera

Contenido

[Componente: Lógica de desafíos](#)

[Verificadores](#)

[Requerimientos Funcionales](#)

[Servidor](#)

1. Componente: Lógica de desafíos

a. Verificadores

- i. Grupo Android
- ii. Grupo Windows Phone

b. Requerimientos Funcionales

- i. **TestingChallengeDetail2, TestingFunction_GetSecondsToWakeMeUp**

Conjunto de entrada	Salida esperada	Salida Obtenida
Level = 1	El método GetSecondsToWakeMeUp debe devolver un arreglo conteniendo el valor 3	Salida esperada
Level = 2	El método GetSecondsToWakeMeUp debe devolver un arreglo conteniendo el valor 9	Salida esperada

- ii. **TestingChallengeDetail2, TestingFunction_CompleteChallenge**

Conjunto de entrada	Salida esperada	Salida Obtenida
cantCorrectWakeUp = 1	BestScore = 20, LastScore = 20	Salida esperada
cantCorrectWakeUp = 2	BestScore = 40, LastScore = 40	Salida esperada
cantCorrectWakeUp = 0	BestScore = 40, LastScore = 0	Salida esperada

- iii. **TestingChallenge, TestingFunction_GetDTChallenge**

Conjunto de entrada	Salida esperada	Salida Obtenida
Se utiliza todas las oportunidades para realizar un desafío.	El desafío debe estar en estado finalizado y la cantidad de oportunidades debe estar agotadas(cero)	Salida esperada

- iv. **TestingChallenge, TestingFunction_GetDurationString**

Conjunto de entrada	Salida esperada	Salida Obtenida
Fecha Inicio de ronda = Date.now Fecha Fin de ronda = Date.now + (6 días)	Se espera como resultado un mensaje mostrando los días restantes para completar desafío. Para este caso se espera "5 días".	Salida esperada.
Fecha Inicio de ronda = Date.now Fecha Fin de ronda = Date.now + (6 horas)	Se espera como resultado un mensaje mostrando los días restantes para completar desafío. Para este caso se espera "5 horas".	Salida esperada.
Fecha Inicio de ronda = Date.now Fecha Fin de ronda = Date.now + (6 minutos)	Se espera como resultado un mensaje mostrando los días restantes para completar desafío. Para este caso se espera "5 minutos".	Salida esperada.
Fecha Inicio de ronda = Date.now Fecha Fin de ronda = Date.now + (6 segundos)	Se espera como resultado un mensaje mostrando los días restantes para completar desafío. Para este caso se espera "menos de un minutos!!".	Salida esperada.

v. **TestingChallengeDetail3, TestingFunction_CompleteChallenge**

Conjunto de entrada	Salida esperada	Salida Obtenida
_countFacebook = 1, _countSms = 3, error = true	Se debe devolver un hash con key en truey value en 130	Salida esperada.
_countFacebook = 2, _countSms = 5, error = false	Se debe devolver un hash con key en false y value en 130	Salida esperada.

2. Servidor

a. Verificadores: Joaquin Velazquez

b. Requerimientos Funcionales, WebServices

i. **getRound**

Conjunto de entrada	Salida esperada	Salida Obtenida
Hay rondas en la base de datos.	Se espera tener un json conteniendo fecha de inicio y fin de la ronda, la lista de desafíos de la ronda, la lista de rankings correspondiente con la ronda de la semana	Salida esperada
No existen rondas en la BD	Se retorna un json conteniendo un mensaje de error.	Salida esperada

ii. **getRanking**

Conjunto de entrada	Salida esperada	Salida Obtenida
Para cualquier estado de la BD	[...{ numberWeek: result[i]._id, start_date: result[i].start_date, end_date: result[i].end_date, ranking: result[i].ranking }...]	Salida esperada

iii. **sendScore**

Conjunto de entrada	Salida esperada	Salida Obtenida
No se ingresa campo userID	Mensaje de error	{ error: true, message: "Usuario no existe." }
Se ingresan los parámetros score y	Se espera un mensaje de éxito y	{ error: false,

userID	ademas la instancia de la BD debe tener actualizado o no el puntaje del usuario en la ronda de la semana actual.	message: "Operación exitosa." }
Se ingresan los parámetros roundID, score y userID	Se espera un mensaje de éxito y además la instancia de la BD debe tener actualizado o no el puntaje del usuario en la ronda pasada como parámetro.	{ error: false, message: "Operación exitosa." }

iv. updateUser

Conjunto de entrada	Salida esperada	Salida Obtenida
No se pasa como parámetro ningún campo necesario	Mensaje de error, mostrando motivo	{ error: true, message: error.message }
Se pasa parámetro userID	Se muestra mensaje de éxito y en la instancia de la base de datos debe quedar registrado el nuevo usuario.	{ error: false, user: user.toJSON() }