

BeatIt!

Modelo de Implementación

Version 3.1.3



Historial de revisiones

Fecha	Versión	Descripción	Autor
27/09/2014	2.1.1		Joaquin Velazquez
28/09/2014	2.1.2	Revisión SQA	Pablo Olivera
18/10/2014	3.1.1	Actualización de la fase de construcción	Gonzalo Javiel
25/10/2014	3.1.2	Nombres de los archivos que implementan cada componente	Gonzalo Javiel
26/10/2014	3.1.3	Revisión SQA	Pablo Olivera

Contenido

[1. Trazabilidad Desde el Modelo de Diseño](#)

[2. Subsistemas](#)

[2.1. Interfaz Gráfica](#)

[2.2. Lógica de Negocio](#)

[2.3. Subsistema de Datos](#)

[3. Componentes](#)

[3.1. UI Challenge1..ChallengeN](#)

[3.2. LogicaC1, ..., LogicaC2](#)

[3.3. Persistencia en Dispositivos](#)

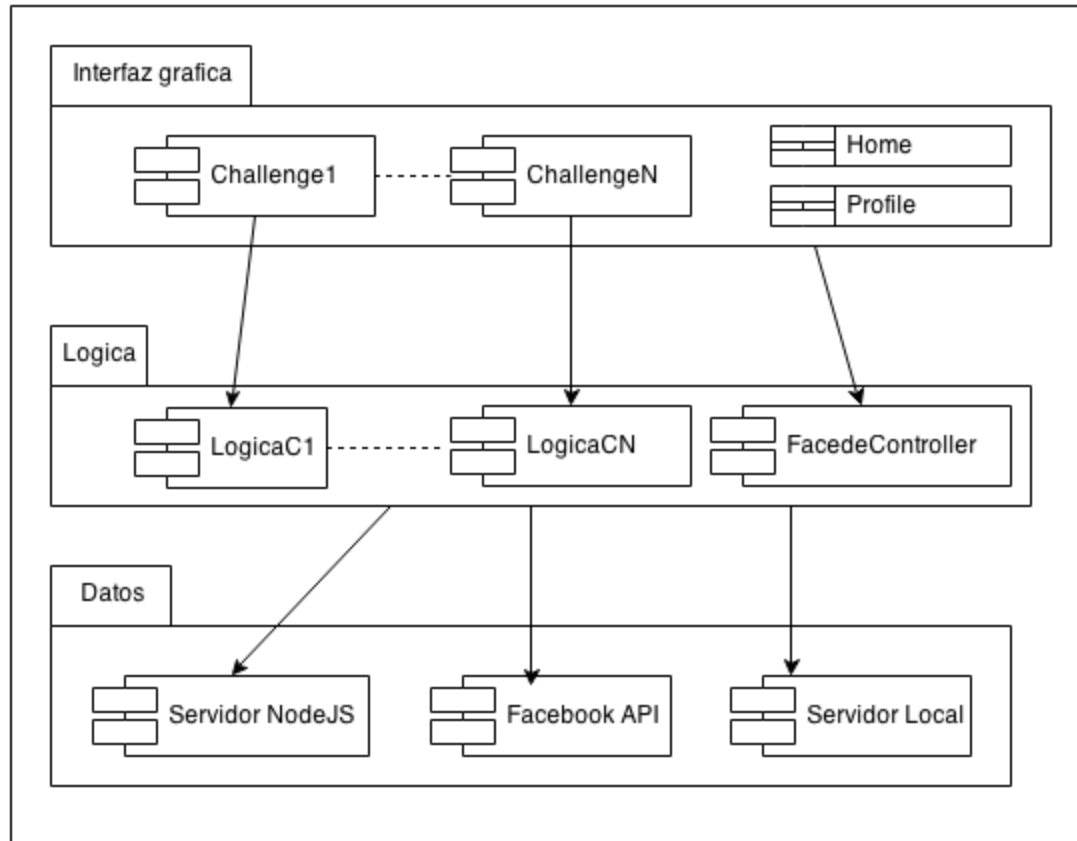
[3.4. WebServer](#)

[4. Interfaces](#)

[4.1. Windows Phone](#)

[4.2. Windows Phone](#)

1. Trazabilidad Desde el Modelo de Diseño



2. Subsistemas

2.1. Interfaz Gráfica

Este subsistema se encarga de implementar la UI. A través de eventos de interfaz generados por usuarios se logra la comunicación con el subsistema de Lógica. Existen dos plataformas disponible : Android y WIndows Phone.

2.2. Lógica de Negocio

Brinda un conjuntos de servicios interactuando con el subsistema de datos para mostrar información e implementar la lógica de los desafíos.

2.3. Subsistema de Datos

Encargado de salvar, consultar datos a través de webservices o interfaces de conexión local.

3. Componentes

A continuación se detallan los componentes de las aplicaciones, en los casos que se encuentren diferencias entre ambas plataformas se detallarán las mismas.

3.1. UI Challenge1..ChallengeN

La aplicación "BeatIt!" contiene una interfaz para cada una en las plataformas Android y Windows Phone. Cada uno de estos Challenge son la implementación de la presentación de los desafíos disponibles en el sistema.

Android

Implementado en el paquete "challenges", en particular la UI está desarrollada en los archivos que tienen como post-fijo "UI".

A su vez cada UI tiene asociado su xml en "BeatIt/res/layout/" donde se define la estructura de la interfaz.

Este componente corresponde al subsistema "UI Android" definido en el documento "DSARQ - Descripción de la Arquitectura"

Windows Phone

Implementado en el namespace "BeatIt_.AppCode.Pages", los archivos con extensión xaml contienen la estructura de la UI, mientras que los archivos con extensión xaml.cs contiene el desarrollo de dichas interfaces.

Este componente corresponde al subsistema "UI Windows Phone" definido en el documento "DSARQ - Descripción de la Arquitectura"

3.2. LogicaC1, ..., LogicaC2

Cada uno de estos componentes atiende los eventos generados por la UI, a su vez esta consume de servicios brindados por el SO del dispositivo, como GPS, Acelerómetro, etc. Además devuelve datos a la UI a través de servicios brindados por el subsistema de Datos.

Android

Implementado en el paquete "challenges", en particular cada desafío contiene un subpaquete donde se encuentran las implementaciones de la lógica de cada desafío.

A su vez se implementó en esta plataforma clases dentro del paquete "connection" encargadas de realizar las comunicaciones con el Servidor.

Este componente corresponde al subsistema "GameEngine Android" definido en el documento "DSARQ - Descripción de la Arquitectura"

Windows Phone

Implementado en el namespace "BeatIt_.AppCode.Challenges", en particular cada desafío tiene una clase independiente, donde sus nombres son de la forma ChallengeDetail[ID_DESAFIO].cs

A su vez se implementó en esta plataforma la clase FacadeController.cs, que brinda servicios a la GUI como el listado de rankings, o detalles de la ronda actual, y WebServicesController.cs que es la encargada de centralizar las comunicaciones con el Servidor. Ambas clases se ubican en el namespace "BeatIt_.AppCode.Controllers"

Este componente corresponde al subsistema "GameEngine WindowsPhone" definido en el documento "DSARQ - Descripción de la Arquitectura"

3.3. Persistencia en Dispositivos

En ambas plataformas se utiliza SQLite.

Android

En esta plataforma se centralizó en la clase "DataManager.java" del paquete "controllers" el manejo de la persistencia en el dispositivo .

Este componente corresponde al subsistema "Persistence Android" definido en el documento "DSARQ - Descripción de la Arquitectura".

Windows Phone

En esta plataforma se centralizó en la clase que implementa "IFacadeController", "FacadeController.cs" ubicada en el namespace "BeatIt_.AppCode.Controllers" el manejo de la persistencia en el dispositivo.

Este componente corresponde al subsistema "Persistence WindowsPhone" definido en el documento "DSARQ - Descripción de la Arquitectura"

3.4. WebServer

Se implementaron los archivos "challenge.js" y "round.js" la lógica correspondiente al subsistema "Challenge" y en "user.js" la lógica del subsistema "User" .

En el archivo app.js se publican los servicios descritos en el subsistema "BeatIt API!".

En cuanto al "DataStorage" el sub-sistema "Persistence" se implementó junto al servidor web, definiendo el modelo en el archivo schemas.js

4. Interfaces

4.1. Windows Phone

IFacadeController

Esta interfaz está definida en IFacadeController.cs en el namespace namespace BeatIt_.AppCode.Interfaces, y es implementada en el archivo FacadeController.cs en namespace BeatIt_.AppCode.Controllers.

Es la encargada de centralizar el acceso a las funcionalidades globales de la aplicación.

4.2. Android

En Android no se definieron interfaces.