BeatIt!

Descripción de la Arquitectura

Versión 1.3

Historia de revisiones

Fecha	Versión	Descripción	Autor
20/08/2014	1.0	Descripción inicial	Gonzalo Javiel
30/08/2014	1.1	Vista del modelo de casos de usos	Martin Alayon
30/08/2014	1.2	Actualización de Diagramas, y descripciones de subsistemas y componentes	Gonzalo Javiel
30/08/2014	1.3	Revisión de SQA	Pablo Olivera

Contenido

1. Introducción
    1.1. Propósito
    1.2. Alcance
    1.3. Definiciones, siglas y abreviaturas.
    1.4. Referencias
    1.5 Visión general
2. Descripción General
3. Motivación
4. Vista del Modelo de Casos de Uso
    2.1. Diagrama de Casos de Uso relevantes a la Arquitectura
    4.2. Casos de Uso relevantes a la Arquitectura
        4.2.1. Iniciar Sesión
        4.2.2. Listar Ranking
        4.2.3. Completar Desafío
5. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño
        Esta sección sera definida en próxima versiones del documento.
6. Vista del Modelo de Diseño
    6.1. Descomposición en Subsistemas
        6.1.1. WindowsPhone Device
        6.1.2. AndroidPhone Device
        6.1.2. WebServer
        6.1.2. DataStorage
    6.2. Diseño de Clases
    6.3 Diseño de Casos de Uso
7. Trazabilidad desde el Modelo de Diseño al Modelo de Implementación
8. Vista del Modelo de Implementación
9. Vista del Modelo de Distribución
    9.1. Diagrama de Distribución
    9.2. Nodos
        9.2.1. Android Device
        9.2.2. Windows Phone Device
    9.3. Conexiones
        9.3.1. Conexión Android Device - Web Server

1. Introducción

Este documento proporciona una descripción de la arquitectura que sera utilizada en el proyecto BeatIt!

1.1. Propósito

Este documento brinda una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Intenta capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

1.2. Alcance

El presente documento tiene como objetivo asistir tanto a los diseñadores como los implementadores en la comprensión del sistema, brindando una guia clara para el desarrollo de solución.

1.3. Definiciones, siglas y abreviaturas.

Los conceptos utilizados en este documento están descriptos en el documento "Glosario".

1.4. Referencias

[1] - Propuesta de Proyecto para el curso Proyecto de Ingeniería de Software 2014 - 15/08/2014

[2] - RQDRQ - Especificación de Requerimientos - Versión 1.5 - 28/08/2014

[3] - Glosario - Versión 1.4 - 28/08/2014

1.5 Visión general

El documento se organiza en secciones. En la primera de ellas se realiza un enfoque de los casos de usos relevantes a la arquitectura, posteriormente se describe la trazabilidad entre el Modelo de Casos de Uso y el Modelo de Diseño, para luego si adentrarse en la descomposición de este ultimo modelo.

En las secciones posteriores se describe la trazabilidad entre las entidades de diseño del Modelo de Diseño y los componentes del Modelo de implementación, y a continuación se profundiza en dichos componentes

Finalmente se describirá la distribución física del sistema en términos de como se distribuye la funcionalidad entre los nodos computacionales.

2. Descripción General

En el proyecto se plantea la utilización del estilo arquitectónico cliente/servidor en tres nivel, donde es relevante indicar que el cliente ejecutara gran parte del procesamiento de la aplicación y la presentación. El servidor por otro lado sera el responsable de la gestión de los datos junto al servidor de base de datos y en menor medida que el cliente, también sera responsable de una parte del procesamiento de la aplicación.

La plataforma elegida para el armado de la infraestructura sera Microsoft Azure. En particular se utilizara los servicios de “Web Site”, que nos brinda servicios como

el escalado automático, la alta disponibilidad, la implementación continua y la posibilidad de conectarse de una manera simple a a las bases datos elegidas.

El servidor, implementado en NodeJS, sera el encargado de proveer servicios a las aplicaciones clientes mediante una API RESTful. A su vez el servidor sera el encargado de realizar las conexiones con aplicaciones de terceros para validar desafíos que así lo requieran.

Por ultimo los datos del sistema se almacenarán en una base de datos NoSQL, MongoDB

3. Motivación

Los motivos de elección de NodeJS y MongoDB si bien tienen el riesgo asociado de ser tecnologías recientes y poco conocidas en el equipo, nos brindan muchas ventajas. Los objetos se pueden manejar con un esquema muy similar en ambas herramientas, a su vez comparten la misma estructura de datos que manejan los WebService REST. Son tecnologías diseñadas para ser escalables un requerimiento siempre presente en aplicaciones móviles.

Es importante también mencionar que NodeJS utiliza el lenguaje JavaScript conocido por todo el equipo de desarrollo

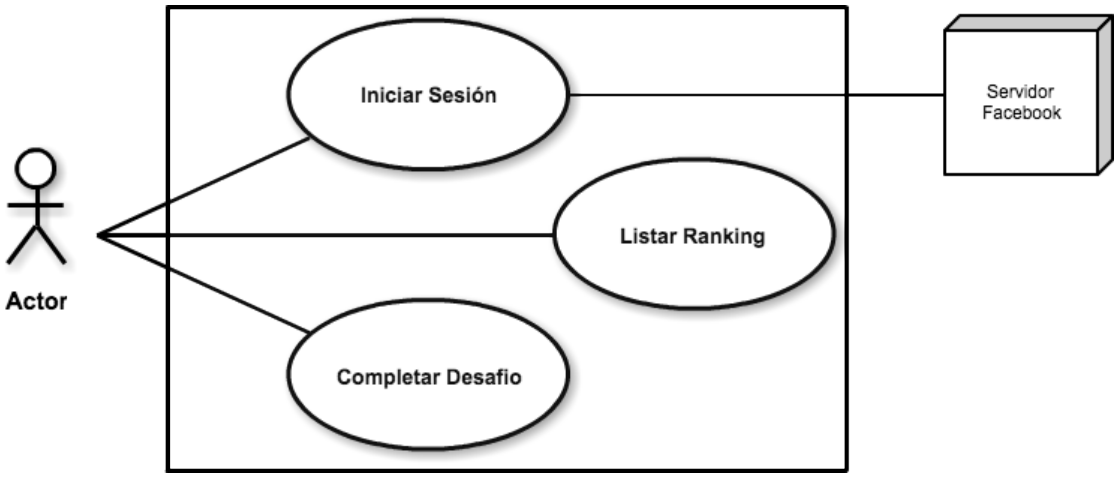
En el diseño de la arquitectura se buscó un bajo acoplamiento con la utilización de servicios, para que sea sencillo, en caso de ser necesario, la sustitución de un componente por otro disminuyendo ese riesgo presente en todo proyecto.

Se definió una capa de servicios para el servidor web que unifica y centraliza la comunicación de los distintos dispositivos con el servidor, el objetivo atrás de esta decisión esta en facilitar la inclusión de nuevos dispositivos con sistemas operativos distintos cumpliendo un los objetivos primordiales del proyecto como es la condición de ser multiplataforma.

Se eligieron las herramientas buscando tener una arquitectura sin bloqueo, lo que permite que las aplicaciones respondan en todo momento, satisfaciendo uno de los requisitos no funcionales más importantes como es la respuesta en pocos segundos.

4. Vista del Modelo de Casos de Uso

2.1. Diagrama de Casos de Uso relevantes a la Arquitectura



4.2. Casos de Uso relevantes a la Arquitectura

A continuación se detallan los casos de uso relevantes a la Arquitectura.
Los demás casos de uso se desarrollan internamente en los dispositivos moviles.

4.2.1. Iniciar Sesión

El caso de uso se inicia cuando el Usuario desea iniciar sesión en la aplicación.
El sistema se comunica con Facebook para obtener los datos del usuario, en caso de error se muestra el error correspondiente y en caso de éxito se debe comunicar con el Servidor para iniciar sesión en la aplicación y obtener los datos del usuario.

4.2.2. Listar Ranking

El caso de uso se inicia cuando el Usuario desea consultar el ranking. El Usuario debe ingresar a la pantalla del ranking de la aplicación. Para desplegar el ranking se debe obtener los datos del Servidor.

4.2.3. Completar Desafío

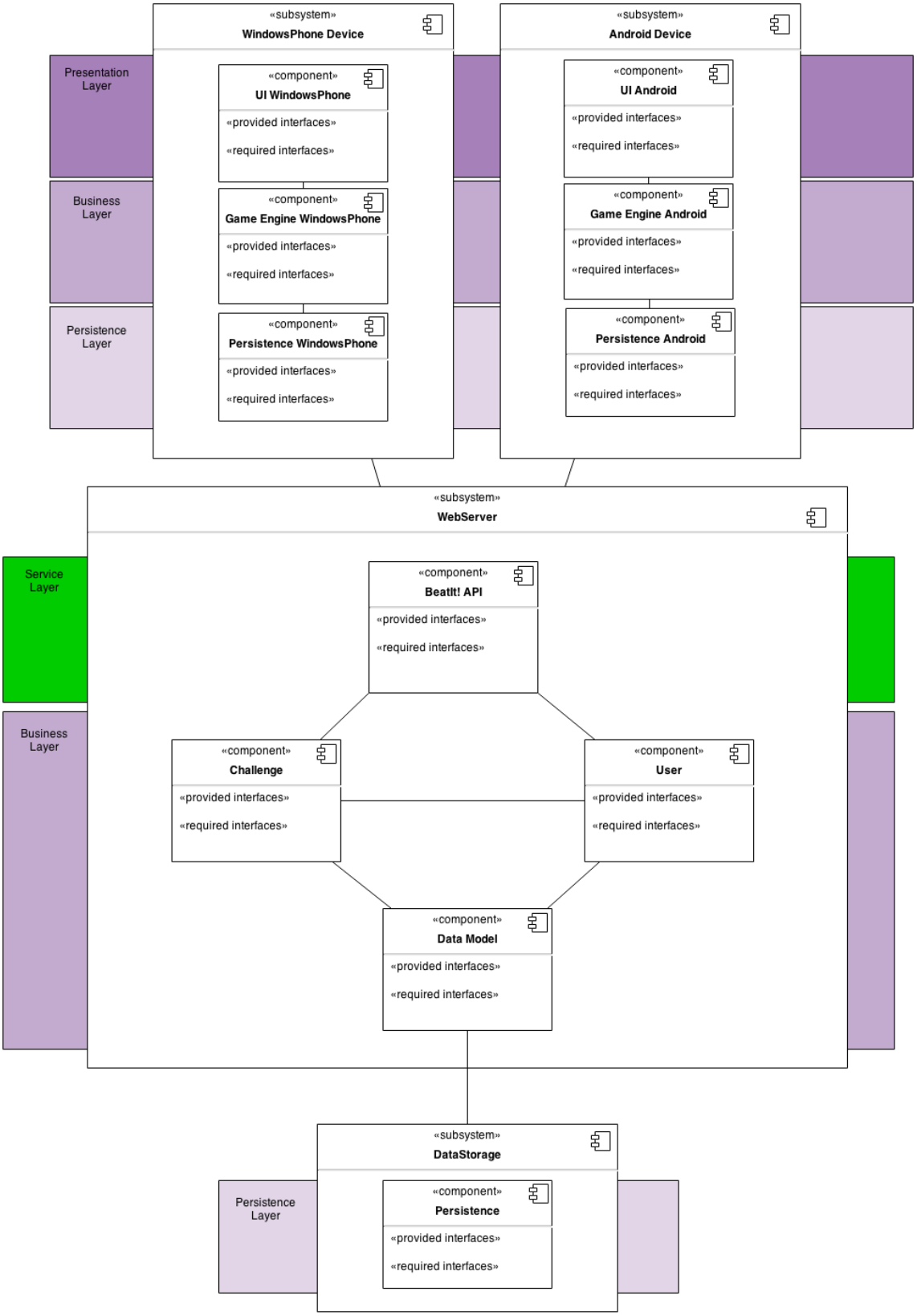
El caso de uso se inicia cuando el Usuario desea realizar un desafío. Para ello selecciona un desafío en curso o inicializado y el Sistema muestra la pantalla del mismo.
Cuando el usuario completa el ultimo desafío se debe enviar la información correspondiente de todos los desafíos al Servidor.

5. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño

Esta sección sera definida en próxima versiones del documento.

6. Vista del Modelo de Diseño

6.1. Descomposición en Subsistemas



6.1.1. WindowsPhone Device

El subsistema "WindowsPhone Device" es la aplicación para el sistema operativo WindowsPhone, como se menciona anteriormente los clientes móviles son del tipo grueso.

Este subsistema tendrá tres grandes componentes:

- "UI WindowsPhone" es la interfaz de usuario. Este componente forma parte de la capa de presentación.
- "GameEngine WindowsPhone" encargado del motor de la aplicación. Este componente forma parte de la capa lógica.
- "Persistence WindowsPhone" encargado de la persistencia del dispositivo móvil. Este componente forma parte de la capa de persistencia.

6.1.2. AndroidPhone Device

El subsistema "Android Device" es la aplicación para el sistema operativo Android, como se menciono anteriormente los clientes moviles son del tipo grueso.

Este subsistema tendrá tres grandes componentes:

- "UI WindowsPhone" es la interfaz de usuario. Este componente forma parte de la capa de presentación
- "GameEngine WindowsPhone" encargado del motor de la aplicación. Este componente forma parte de la capa lógica.
- "Persistence WindowsPhone" encargado de la persistencia del dispositivo móvil. Este componente forma parte de la capa de persistencia.

6.1.2. WebServer

El subsistema "WebServer" es el correspondiente al servidor web, en el mismo consta cuatro grandes componentes:

- "BeatIt API!" esta API es la encargada de publicar los servicios para que sean consumidos por los dispositivos. Este componente forma parte de la capa de servicios del sistema.
- "Challenge" este componente es el encargado de la lógica referente a los desafíos (ranking, validación de desafios). Este componente forma parte de la capa lógica del sistema.
- "User" este componente es el encargado de la lógica referente a la gestión de usuarios (registro, login, listas de amigos).
- "DataModel" este componente sera utilizado para encapsular el acceso a la persistencia de los datos. Este componente forma parte de la capa lógica del sistema.

6.1.2. DataStorage

El subsistema "DataStorage" es el correspondiente al servidor de base de datos, el mismo consta de un componente:

- "Persistence" este componente es el encargado de persistir los datos en la base de datos. Este componente forma parte de la capa de persistencia del sistema.

6.2. Diseño de Clases

Esta sección sera definida en próxima versiones del documento.

6.3 Diseño de Casos de Uso

Esta sección sera definida en próxima versiones del documento.

7. Trazabilidad desde el Modelo de Diseño al Modelo de Implementación

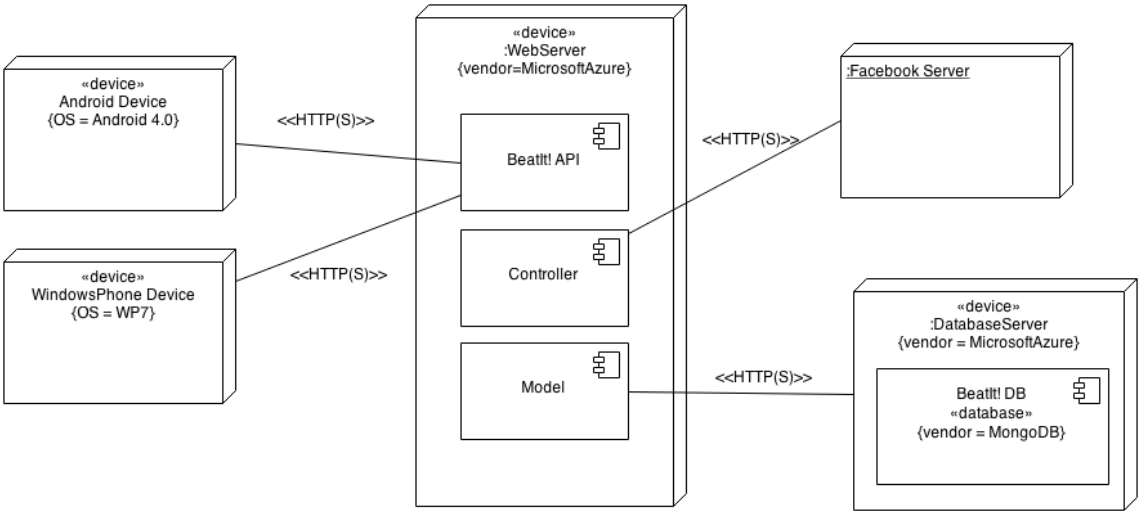
Esta sección sera definida en próxima versiones del documento.

8. Vista del Modelo de Implementación

Esta sección sera definida en próxima versiones del documento.

9. Vista del Modelo de Distribución

9.1. Diagrama de Distribución



9.2. Nodos

En esta sección se da más detalle de los nodos relevantes a la arquitectura del sistema

9.2.1. Android Device

En este nodo se alojara la aplicación a desarrollar, es un dispositivo móvil ejecutando el sistema operativo Android.

9.2.2. Windows Phone Device

En este nodo se alojara la aplicación a desarrollar, es un dispositivo móvil ejecutando el sistema operativo Windows Phone.

9.2.3. Web Server

En este nodo se alojara el Servidor Web implementado en NodeJS, se utilizaran los servicios de Microsoft Azure para alojar este nodo, en particular Azure Web Sites, pudiéndose instalar posteriormente en una Maquina Virtual de ser necesario.

9.2.7. DatabaseServer

En este nodo se alojara la base de datos MongoDB, la misma estará ubicada en el servicio Cloud Microsoft Azure.

9.2.8. Facebook Server

Este nodo es externo a nuestro sistema, pero es pertinente mencionarlo ya que se utilizaran servicios brindados por Facebook para la autenticación de usuarios y listas de amigos.

9.3. Conexiones

9.3.1. Conexión Android Device - Web Server

Esta conexión se realizar sobre HTTP utilizando servicios REST brindados por el servidor web.

9.3.2. Conexión WindowsPhone Device - Web Server

Esta conexión se realizar sobre HTTP utilizando servicios REST brindados por el servidor web.

9.3.4. Conexión Web Server - Database

Esta conexión en el despliegue inicial sera interna en Microsoft Azure, se utiliza https sobre TCP.

Dado que es uno de los servicios brindados por la plataforma Microsoft Azure se asume una disponibilidad alta en esta conexión.

9.3.4. Conexión Web Server - Facebook Server

La conexión entre el servidor web y la API del servidor de Facebook se hará por un canal https sobre TCP.

Se asume que el servidor de Facebook tiene una alta disponiblidad.