Interpool

Descripción de la Arquitectura

Versión 7.1

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 21/08/10 | 1.0 | Descripción inicial de la Arquitectura | Marcos Sander |
| 29/08/10 | 1.1 | Agregado de un caso de uso | Marcos Sander |
| 29/08/10 | 1.1 | Revisión de Documento | Alejandro García |
| 02/09/10 | 3.0 | Cambio de algunos casos de uso | Marcos Sander |
| 04/09/10 | 3.0 | Revisión de Documento | Alejandro García |
| 22/09/10 | 7.0 | Agregado de las secciones faltantes | Marcos Sander |
| 25/09/10 | 7.1 | Revisión de SQA | Javier Madeiro |

Contenido

1. Introducción 4

1.1. Propósito 4

1.2. Alcance 4

1.3. Definiciones, siglas y abreviaturas. 4

1.4. Referencias 4

1.5. Visión general 4

2. Vista del Modelo de Casos de Uso 5

2.1. Diagrama de Casos de Uso relevantes a la Arquitectura 5

2.2. Casos de Uso relevantes a la Arquitectura 5

2.2.1. Iniciar Iteración 5

2.2.2. Interrogar personaje 5

2.2.3. Guardar Estado 5

2.2.4. Actualizar datos ciudades 5

3. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño 6

3.1. Iniciar Iteración 6

3.2. Interrogar personaje 7

3.3. Guardar Estado 8

3.4. Actualizar datos ciudades 9

4. Vista del Modelo de Diseño 10

4.1. Descomposición en Subsistemas 10

4.1.1. Subsistema Servicios 10

4.1.2. Subsistema Negocios 10

4.1.3. Subsistema Datos 10

4.1.4. Subsistema Servicios Externos 10

4.2. Diseño de Clases 10

4.2.1. Clase User 11

4.2.2. Clase Game 11

4.2.3. Clase NodePath 11

4.2.4. Clase City 11

4.2.5. Clase CityProperty 11

4.2.6. Clase Famous 11

4.2.7. Clase News 11

4.2.8. Clase Clue 11

4.2.9. Clase Suspect 11

4.2.10. Clase Level 11

4.3. Diseño de Casos de Uso 11

5. Trazabilidad desde el Modelo de Diseño al Modelo de Implementación 12

6. Vista del Modelo de Implementación 13

6.1. Subsistemas 13

6.1.1. Subsistema Services 13

6.1.2. Subsistema Bussiness 13

6.1.3. Subsistema Data 13

6.1.4. Subsistema Extern Services 13

6.2. Componentes 13

6.3. Interfaces 13

6.3.1. IWPhoneAzure 13

6.3.2. IAzureServicios 13

6.3.3. IAzureDatos 13

6.3.4. API de Facebook 13

6.3.5. API de Bing 13

7. Vista del Modelo de Distribución 14

7.1. Diagrama de Distribución 14

7.2. Nodos 15

7.2.1. Nodo Windows Phone 7 15

7.2.2. Nodo Azure 15

7.2.3. Nodo Facebook 15

7.2.4. Nodo Bing 15

7.3. Conexiones 15

7.3.1. Windows Phone - Azure 15

7.3.2. Azure – Facebook 15

7.3.3. Azure – Bing 15

1. Introducción
   1. Propósito

Este documento describe la arquitectura del sistema Interpool desde distintas vistas, la de Modelo de Casos de Uso, Modelo de Diseño, Modelo de Distribución y Modelo de Implementación, también la trazabilidad del Modelo de Casos de Uso al de Diseño y del de Diseño al de Implementación. Con esto se trata de dar un conocimiento general acerca de la arquitectura del sistema.

La audiencia esperada es el equipo de desarrollo y el cliente.

* 1. Alcance

Este documento pretende ser un nexo entre lo relevado en los requerimientos y el diseño del sistema. La arquitectura aquí descripta será usada cómo guía para el diseño y luego para la implementación del sistema.

* 1. Definiciones, siglas y abreviaturas.

Todos los términos, siglas, y abreviaciones requeridas para interpretar apropiadamente este documento se encuentran en el Glosario del proyecto.

* 1. Referencias

[1] Especificación de requerimientos, RQDRQG2v6\_0.pdf

[2] Glosario, RQGLOGv8\_0.pdf

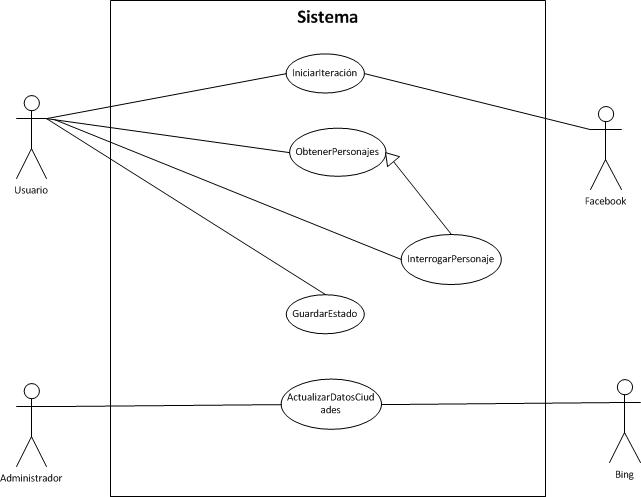
[3] Modelo de Casos de Uso, RQMODG2v8\_0.pdf.

[4] Modelo de Diseño.

* 1. Visión general

El resto de las secciones describe la arquitectura del sistema desde una vista distinta. La sección 2 la vista de Modelo de Casos de Uso, la sección 3 la trazabilidad entre el Modelo de Casos de Uso y Diseño, la sección 4 la vista del Modelo de Diseño, la sección 5 la trazabilidad desde el Modelo de Diseño y el Modelo de Implementación, la sección 6 la vista del Modelo de Implementación y la sección 7 el Modelo de Distribución.

1. Vista del Modelo de Casos de Uso
   1. Diagrama de Casos de Uso relevantes a la Arquitectura



* 1. Casos de Uso relevantes a la Arquitectura
     1. Iniciar Iteración

El usuario indica que quiere empezar un nuevo juego y se carga el juego.

* + 1. Interrogar personaje

El usuario elige uno de los 3 personajes disponibles para interrogar, y este le da una pista con datos del sospechoso y/o característica de la próxima ciudad a donde deberá ir a buscar más pistas.

* + 1. Guardar Estado

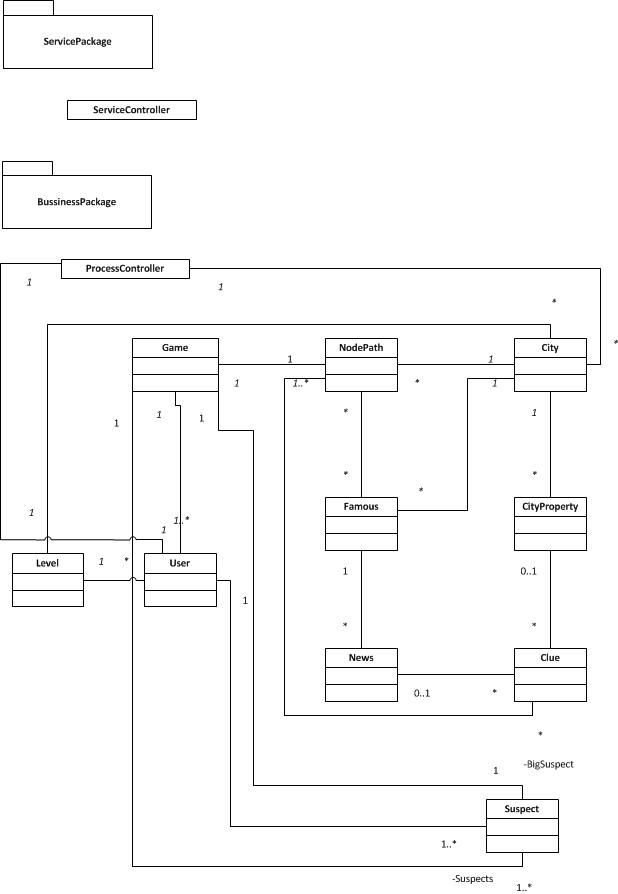
Cuando el usuario decide salir del juego en medio de una iteración para hacer otra actividad, se guarda el estado del mismo.

* + 1. Actualizar datos ciudades

El administrador del sistema ejecuta esta opción, el sistema actualiza en su base de datos las pistas referentes a la ciudad con información del momento.

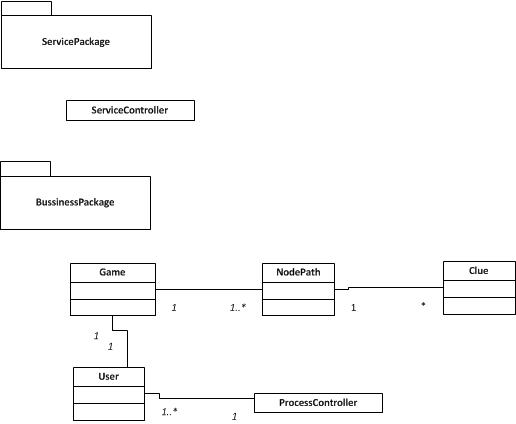
1. Trazabilidad desde el Modelo de Casos de Uso al Modelo de Diseño
   1. Iniciar Iteración

El usuario indica que quiere empezar un nuevo juego. Desde el **Windows Phone** se solicita un servicio al servidor que está en **Azure**. La **capa de servicios** de Azure, encargada de brindar servicios al Windows Phone, procesa este servicio. Solicita un servicio en la **capa de procesamiento** para traer cierta cantidad de sospechosos (variante según el nivel de juego) junto con sus preferencias y algún otro dato, como por ejemplo su foto. La capa de servicios solicita otro servicio a la capa de procesamiento que es generar el camino por donde irá el ladrón junto con los famosos y pistas para cada ciudad. La capa de procesamiento genera el camino que hará el ladrón, y en ese momento también se generan las pistas para cada ciudad en que va a estar el ladrón. Para generar esas pistas, esta capa se comunica con la **capa de datos**, ya que los datos referentes a las pistas (datos para las pistas referentes a las ciudades y datos relacionados con los famosos) van a estar actualizados ahí. Todos estos datos quedarán en memoria y serán pasados al Windows Phone a medida que este los vaya pidiendo.



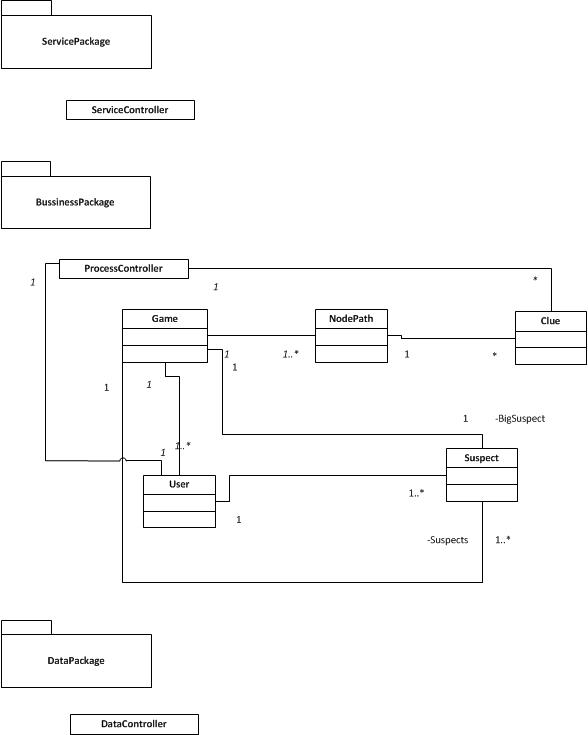
* 1. Interrogar personaje

El usuario indica el personaje a interrogar. La **capa de negocios** del **Windows Phone** se comunica con la **capa de servicios** del servidor en **Azure**. La capa de procesamiento solicita a la **capa de procesamiento** la pista (que había sido generada previamente en el caso de uso Iniciar Iteración). Esta pista está en memoria y se pasan estos a datos al Windows Phone.



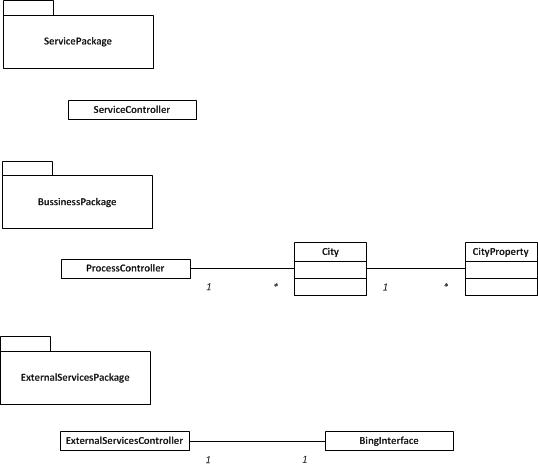
* 1. Guardar Estado

Cuando el usuario elige salir del juego en ejecución, se capta este evento y la **capa de negocios** del **Windows Phone** envía la solicitud de guardar el estado al **servidor en Azure.** La **capa de servicios** recibe la solicitud de guardar estado desde la **capa de procesamiento**. En Azure, la capa de procesamiento, solicita guardar los datos a la **capa de datos** que está en el servidor.



* 1. Actualizar datos ciudades

El administrador (o un proceso automático), elige actualizar los datos de las ciudades. La **capa de servicios externos**, se comunica con el **servidor de noticias de Bing**, y a partir de las noticias obtenidas ahí se genera las pistas de una ciudad dada. Estos datos son pasados a la **capa de datos** y guardados ahí. Este proceso es llevado a cabo para cada ciudad que está en la base de datos.



1. Vista del Modelo de Diseño
   1. Descomposición en Subsistemas
      1. Subsistema Servicios

El subsistema de servicios está en la capa de Servicios de Azure. Es el subsistema encargado de la comunicación con Windows Phone. Cómo clase principal está la clase ServiceController que implementa la interfaz de comunicación con Phone.

* + 1. Subsistema Negocios

El subsistema de negocios implementa la lógica y funciones del juego. La clase principal de este subsistema es ProcessController, que implementa la interfaz de las funciones solicitadas por el subsistema de servicios. También están las clases que describen las instancias propias del juego cómo por ejemplo User, Game, City, etc.

* + 1. Subsistema Datos

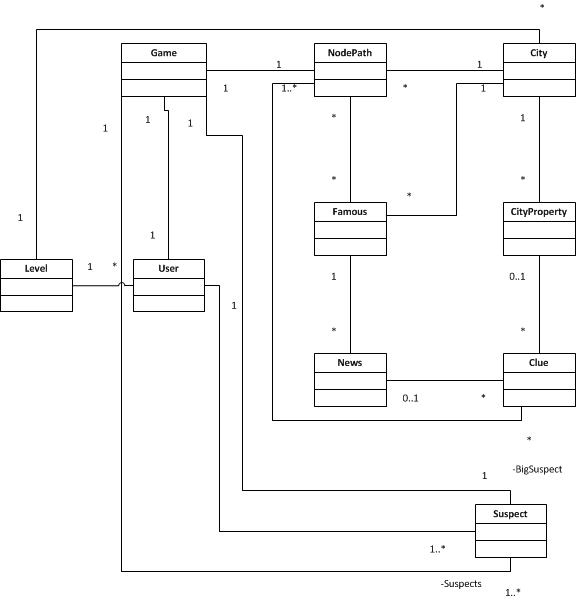
El subsistema de datos es encargado de hacer la persistencia del juego a la base de datos. La clase principal de este es DataController, responsable de las funciones que guardan los datos del juego.

* + 1. Subsistema Servicios Externos

Subsistema que tiene como propósito y responsabilidad la comunicación del servidor con los servicios de Facebook y Bing. La clase principal que la implementa es ExternServicesController.

* 1. Diseño de Clases

A continuación se muestra el diagrama de clases de las clases pertenecientes al paquete de negocios de Azure.



* + 1. Clase User

Esta clase representa a los usuarios del sistema. Cuando un usuario juega por primera vez se crea en el sistema una instancia de esta clase y la misma guarda el id de Facebook del usuario.

* + 1. Clase Game

Esta clase representa el juego actual del jugador. Participa en la resolución de la mayoría de las funcionalidades principales del juego.

* + 1. Clase NodePath

La clase NodePath representa el estar en una ciudad en un juego dado. Esta clase tiene relacionado la ciudad en la que está, los famosos que se pueden interrogar en esa ciudad, y las pistas que estos pueden dar.

* + 1. Clase City

Esta clase representa una ciudad posible a estar o viajar en el juego. Tiene relacionada a ella a los famosos que pertenecen a esa ciudad.

* + 1. Clase CityProperty

La clase CityProperty contiene una propiedad dinámica o estática de la ciudad, la cual sirve para generar las pistas para viajar a esa ciudad.

* + 1. Clase Famous

Representa un famoso de una ciudad dada, que da una pista en el juego.

* + 1. Clase News

Contiene una noticia ya parseada para ser usada en una pista, relacionada a un famoso, que sirve para dar gracia al juego.

* + 1. Clase Clue

Esta clase representa a las pistas que dan los famosos en un juego dado. Para un juego se guardan las 3 pistas por cada ciudad por donde va a ir el sospechoso.

* + 1. Clase Suspect

Representa todos los sospechosos del juego. Uno de ellos tiene el rol de gran sospechoso. Los datos de cada instancia son cargadas desde Facebook, de la lista de amigos del usuario.

* + 1. Clase Level

Esta clase representa el nivel actual del jugador. Este nivel se utiliza para dar dificultad al juego, seleccionando ciudades más difíciles (menos conocidas) y dando pistas estáticas entre los últimos informantes.

* 1. Diseño de Casos de Uso

Este punto esta pendiente.

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

La implementación del sistema será divida en 2 proyectos separados, uno para implementar la funcionalidad del juego en el Windows Phone 7 y el otro para implementar la funcionalidad del juego que va a correr en el servidor Azure en la nube.

1. Vista del Modelo de Implementación

En esta sección se describe la estructura general del modelo de implementación y la descomposición del software en módulos.

* 1. Subsistemas
     1. Subsistema Services

Este subsistema básicamente maneja la comunicación con Windows Phone 7. La clase de implementación ServicesController implementa la interfaz IServices.

* + 1. Subsistema Bussiness

La clase de implementación ProcessController implementa las operaciones que solicita la clase ServicesController del subsistema Services. Por cada clase de negocio identificada en el diseño se crea una clase de implementación.

* + 1. Subsistema Data

La clase de implementación DataController maneja los pedidos de persistencia hechas por las demás clases en los demás subsistemas.

* + 1. Subsistema Extern Services

La clase de implementación ExternServicesController maneja la comunicación del Servidor con Facebook y Bing.

* 1. Componentes

Esta subsección queda pendiente de documentar.

* 1. Interfaces
     1. IWPhoneAzure

Esta interfaz es implementada por el servidor Azure y es usada por el Windows Phone, la cual le sirve para solicitar servicios al servidor.

* + 1. IAzureServicios

Implementada por la capa de Negocios del servidor. Sirve para solicitar servicios desde la capa de Servicios.

* + 1. IAzureDatos

Implementado por la capa de datos.

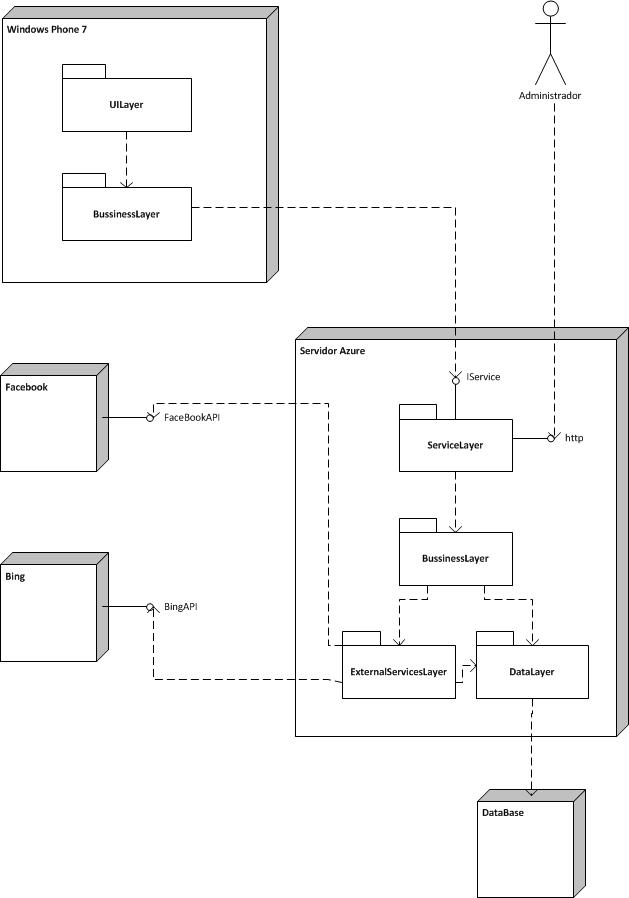
* + 1. API de Facebook

Se accederá a los datos del cliente y de sus amigos en Facebook, para usarlos en el juego, comunicándonos a través de la API proporcionada por Facebook.

* + 1. API de Bing

Se accederá a las noticias en Bing para obtener los datos de las ciudades dinámicamente y también de los famosos al generar las pistas. Esto lo haremos a través de la API proporcionada por Bing.

1. Vista del Modelo de Distribución
   1. Diagrama de Distribución



* 1. Nodos
     1. Nodo Windows Phone 7

Es el dispositivo de hardware sobre el cuál el juego será ejecutado. Un celular con el sistema Windows Phone 7 instalado.

* + 1. Nodo Azure

Nodo que está en la nube. En este están la lógica del negocio y los datos que se necesiten persistir.

* + 1. Nodo Facebook

El servidor se comunica con este nodo, para traer datos de los amigos del usuario a través de una API definida para Facebook.

* + 1. Nodo Bing

El servidor se comunica con este nodo, para traer datos de ciudades y famosos, para poder generar las pistas. Lo hace a través de una API del buscador.

* 1. Conexiones
     1. Windows Phone - Azure

A través de WCF, con WCFServices.

* + 1. Azure – Facebook

A través de la API proporcionada por Facebook.

* + 1. Azure – Bing

A través de la API proporcionada por Bing.