Interpool

Modelo de Implementación

Versión 11.1

Historia de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 25/09/10 | 7.0 | Creación del documento | Ignacio Infante, Marcos Sander |
| 26/09/10 | 7.1 | Revisión de SQA | Javier Madeiro |
| 24/10/10 | 11.0 | Documentación de las operaciones de cada interfaz | Marcos Sander |
| 24/10/10 | 11.1 | Revisión de documento | Alejandro García |

Contenido

[Contenido 2](#_Toc275677684)

[1. Trazabilidad desde el Modelo de Diseño 3](#_Toc275677685)

[2. Subsistemas 4](#_Toc275677686)

[2.1. Subsistema Windows Phone 7 4](#_Toc275677687)

[2.2. Subsistema Servicios - Azure 4](#_Toc275677688)

[2.3. Subsistema Negocios - Azure 4](#_Toc275677689)

[2.4. Subsistema Datos – Azure 4](#_Toc275677690)

[2.5. Subsistema Servicios Externos - Azure 4](#_Toc275677691)

[3. Interfaces 4](#_Toc275677692)

[3.1. IInterpoolWP7 4](#_Toc275677693)

[3.2. IProcessController 5](#_Toc275677694)

[3.3. IDataManager 6](#_Toc275677695)

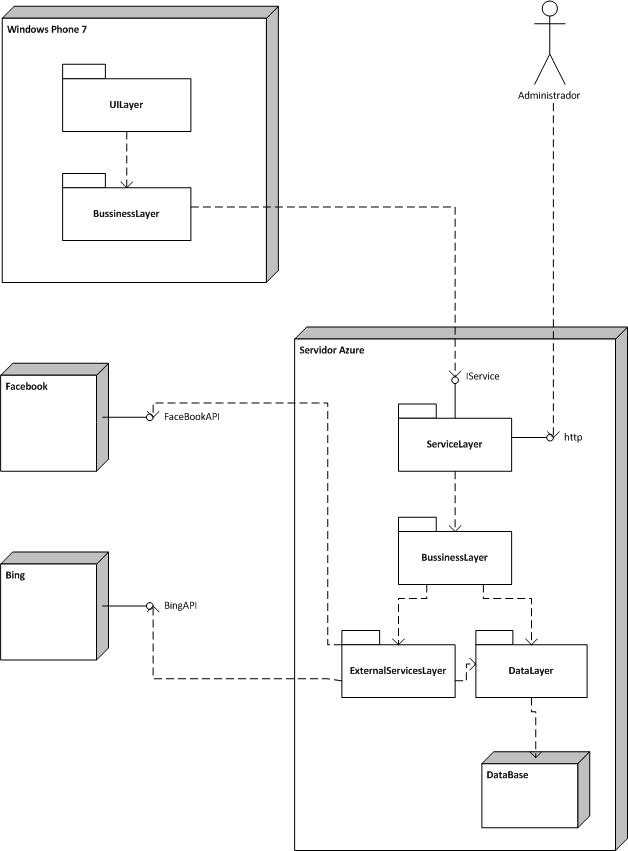
[3.4. IFacebookController 6](#_Toc275677696)

[3.5. IBingController 7](#_Toc275677697)

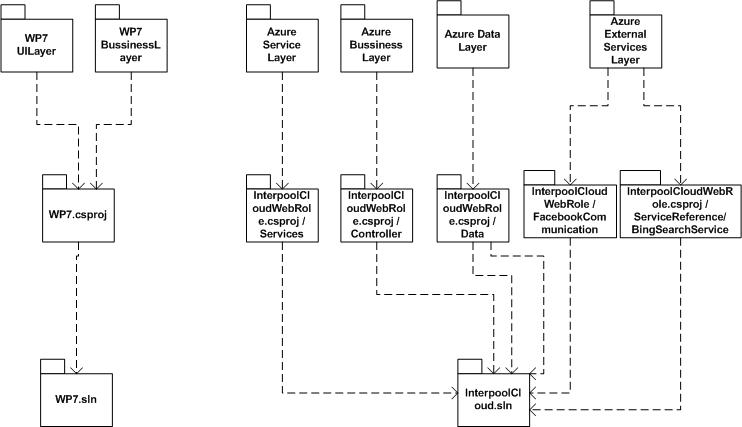
1. Trazabilidad desde el Modelo de Diseño

El diagrama que sigue describe el mapeo de los distintos elementos de la arquitectura con su implementación.

Los elementos de la arquitectura se representan en este diagrama:



Y en el siguiente diagrama se muestra la correspondencia con los elementos de implementación:



1. Subsistemas
   1. Subsistema Windows Phone 7

Este subsistema está implementado en la solución WP7.sln, en el proyecto WP7.csproj. Se implementan la interfaz de usuario y la lógica que se ejecutará en el Windows Phone 7.

* 1. Subsistema Servicios - Azure

Este subsistema está implementado en la solución InterpoolCloud.sln, el proyecto InterpoolCloudWebRole.csproj, en la carpeta Services. Se implementa la conexión con el Windows Phone 7.

* 1. Subsistema Negocios - Azure

Este subsistema está implementado en la solución InterpoolCloud.sln, el proyecto InterpoolCloudWebRole.csproj, en la carpeta Controller. Se implementa la lógica de los servicios ofrecidos al Windows Phone.

* 1. Subsistema Datos – Azure

Este subsistema está implementado en la solución InterpoolCloud.sln, el proyecto InterpoolCloudWebRole.csproj, en la carpeta Data. Se implementa los servicios de persistencia en la base de datos y la conexión con la misma.

* 1. Subsistema Servicios Externos - Azure

Este subsistema está implementado en la solución InterpoolCloud.sln, el proyecto InterpoolCloudWebRole.csproj, en la carpeta FacebookCommunication y la carpeta ServiceReference/BingSearchService. Se implementa la comunicación con Facebook y Bing y los servicios necesarios para la aplicación que usan Facebook y Bing.

1. Interfaces
   1. IInterpoolWP7

Esta interfaz que está definida en el archivo IInterpoolWP7.cs, es utilizada por la capa de negocios del Windows Phone 7 para acceder a los servicios de Azure. La implementa el archivo InterpoolWP7.cs.

Se lista a continuación las operaciones que hacen parte de esta interfaz:

* void StartGame(string userIdFacebook);

Esta operación recibe como parámetro al identificador de facebook del usuario (userIdFacebook) y su resultado es iniciar un nuevo juego en el servidor.

* DataCity GetCurrentCity(string userIdFacebook);

Recibe el identificador de facebook del usuario y retorna la ciudad en que se está en el juego actual.

* List<DataCity> GetPossibleCities(string userIdFacebook);

Recibe como parámetro de entrada al identificador de facebook del usuario y retorna las ciudades posibles distinta de la actual que se podía haber viajado desde la ciudad anterior.

* DataFamous GetCurrentFamous(string userIdFacebook, int numClue);

Recibe cómo parámetro de entrada el identificador del usuario de facebook y el número de pista que se quiere obtener (por ciudad son 3 pistas y el número de pista es un número del 1 al 3) y se retorna los datos del famoso que dará esa pista.

* List<DataFacebookUser> FilterSuspects(string userIdFacebook, DataFacebookUser fbud);

Esta operación recibe cómo parametros de entrada al id del usuario de facebook y también a los filtros instanciados y retorna una lista de todos los sospechosos, que para el juego actual del usuario cumplen que sus características sean iguales a la de los filtros instanciados.

* void EmitOrderOfArrest(string userIdFacebook, string userIdFacebookSuspect);

Dado un usuario de facebook identificado por su id de facebook y dado un sospechoso dado del juego actual del usuario, se emite una orden de arresto para dicho sospechoso.

* List<DataCity> GetCities(string userIdFacebook);

Para el usuario de facebook dado y su juego actual, se devuelven las ciudades posibles a viajar.

* DataCity Travel(string userIdFacebook, string nameNextCity);

Dado el usuario de facebook y el nombre de una ciudad posible a viajar, se realiza el viaje.

* DataClue GetClueByFamous(string userIdFacebook, int numFamous);

Esta operación recibe el id del usuario de facebook y el número de famoso a interrogar y retorna la pista que ese famoso da.

}

* 1. IProcessController

Esta interfaz está definida en el archivo IProcessController.cs, es utilizada por la capa de servicios de Azure para comunicarse con la capa de Procesamiento de Azure. Es implementada por el archivo ProcessController.cs. Las operaciones son:

* void StartGame(string userIdFacebook);
* DataCity GetCurrentCity(string userIdFacebook);
* List<DataCity> GetPossibleCities(string userIdFacebook);
* DataFamous GetCurrentFamous(string userIdFacebook, int numClue);
* List<DataFacebookUser> FilterSuspects(string userIdFacebook, DataFacebookUser fbud);
* DataCity Travel(string userIdFacebook, string nameNextCity);
* void EmitOrderOfArrest(string userIdFacebook, string userIdFacebookSuspect);
* List<DataCity> GetCities(string userId);
* DataClue GetClueByFamous(string userIdFacebook, int numFamous);

* 1. IDataManager

El archivo IDataManager.cs define a esta interfaz. Es utilizada por la capa de procesamiento de Azure y por la capa de servicios externos de Azure, para comunicarse con la capa de datos. La implementa el archivo DataManager.cs.

A continuación se describe las operaciones de esta interfaz:

* + IQueryable<City> GetCities(InterpoolContainer context);
  + IQueryable<Level> GetLevels(InterpoolContainer context);
  + IQueryable<Famous> GetFamousByCity(City city, InterpoolContainer context);
  + IQueryable<CityProperty> GetCityPropertyByCity(City city, InterpoolContainer context);
  + IQueryable<Suspect> GetSuspectByGame(Game g, InterpoolContainer context);
  + string GetLastUserIdFacebook(InterpoolContainer context);
  + string GetUserIdFacebookByLoginId(string userLoginId, InterpoolContainer context);
  + OAuthFacebook GetLastUserToken(InterpoolContainer context);
  + void StoreUser(User user, InterpoolContainer context);
  + void StoreSuspect(Suspect suspect, InterpoolContainer context);
  + void SaveChanges(InterpoolContainer context);
  + InterpoolContainer GetContainer();
  + string GetParameter(string name, InterpoolContainer context);
  + Game GetGameByUser(string userIdFaceook, InterpoolContainer conteiner);
  + List<DataFacebookUser> FilterSuspects(string userIdFacebook, DataFacebookUser fbud, InterpoolContainer container);
  1. IFacebookController

Esta interfaz es definida en el archivo IFacebookController.cs, es utilizada por la capa de Procesamiento de Azure para comunicarse con parte de la capa de servicios externos que implementan la comunicación con Facebook.

* string GetUserId(OAuthFacebook auth);

Retorna el id de facebook del usuario.

* List<string> GetFriendsId(string userId);

Retorna la lista de ids de los amigos de facebook del usuario.

* DataFacebookUser GetFriendInfo(string userId, string userFriendId);

Retorna la información del amigo del usuario, con id de facebook ‘userIdFriendId’.

* List<string> GetFriendsNames(OAuthFacebook auth, string userId);

Retorna los nombres de los amigos del usuario.

* void AddFriend(string name, string id, OAuthFacebook auth);

* void DownloadFacebookUserData(OAuthFacebook auth, Game game, InterpoolContainer context);

Guarda la información de los amigos del usuario en la base de datos.

* 1. IBingController

Esta interfaz es definida en el archivo IBingController.cs, es utilizada por la capa de Procesamiento de Azure para comunicarse con parte de la capa de servicios externos que implementan la comunicación con Bing.