

CONNECT! & WHERE IS MY FRIEND?

DESCRIPCIÓN DE LA ARQUITECTURA

VERSIÓN 1.8

Historia de revisiones

Fecha	Versión	Descripción	Autor
01/09/2013	1.0	Creación del documento	Dario Britos
08/09/2013	1.1	Vista del modelo de CU	Dario Britos
9/15/2013	1.2	Diagramas de comunicación, retoques en núcleo connect y agregado del núcleo WIMF	Dario Britos, Cristiano Cohelo, Martin Gaudioso, Gabriel Barbato, Gimena Bernadet
9/19/2013	1.3	Descripción casos de uso, descripción componentes, ajuste de diagramas	Dario Britos
10/21/2013	1.4	Actualización y agregado de trazabilidad al modelo de implementación.	Darío Britos
28/10/2013	1.5	Agregado sección 1.8 y modificaciones varias	Dario Britos
4/11/2013	1.6	Modificaciones minimas	Dario Britos
24/11/2013	1.7	Retoque final y entrega	Dario Britos
24/11/2013	1.8	Look & Feel	Leonardo Clavijo

CONTENIDO

1.1.INTRODUCCIÓN	4
1.2.PROPÓSITO	4
1.3.ALCANCE	4
1.4.DEFINICIONES, SIGLAS Y ABREVIATURAS.	4
1.5.REFERENCIAS	4
1.6.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	6
2.2.1.Registrar usuario (CN).....	6
2.2.2.Configurar redes sociales (CN).....	6
2.2.3.Ingresar red social (CN).....	6
2.2.4.Ingreso de permisos (CN).....	6
2.2.5.Generar y Publicar Datos mediante un código QR (CN).....	6
2.2.6.Aceptar pedido de conexión (CN).....	6
2.2.7.Sincronizar contactos (CN).....	6
2.2.8.Leer Datos mediante código QR (CN).....	6
2.2.9.Listar usuarios (WIMF).....	6
2.2.10.Envío de solicitud (WIMF).....	6
2.2.11.Aceptar solicitud (WIMF).....	6
3.TRAZABILIDAD DESDE EL MODELO DE CASOS DE USO AL MODELO DE DISEÑO CONNECT!.....	7
3.1.1.Registrar usuario (CN).....	7
3.1.2.Configurar redes sociales (CN).....	7
3.1.3.Ingresar red social (CN).....	7
3.1.4.Ingreso de permisos (CN).....	7
3.1.5.Generar y Publicar Datos mediante un código QR (CN).....	8
3.1.6.Aceptar pedido de conexión (CN).....	8
3.1.7.Sincronizar contactos (CN).....	8
3.1.8.Leer Datos mediante código QR (CN).....	8
4.TRAZABILIDAD DESDE EL MODELO DE CASOS DE USO AL MODELO DE DISEÑO WIMF?.....	9
4.1.1.Listar usuarios.....	9
4.1.2.Envío de solicitud.....	9
4.1.3.Aceptar solicitud.....	9
5.VISTA DEL MODELO DE DISEÑO.....	10
5.1.DESCOMPOSICIÓN EN SUBSISTEMAS	10
5.1.1.Connect UI.....	11
5.1.2.WIMF UI.....	11
5.1.3.Connect Núcleo.....	11
5.1.4.WIMF Núcleo.....	12
5.1.5.Data Access Controller.....	13
5.1.6.Connect Logica (SR).....	13
5.1.7.WIMF Logica (SR).....	13
5.1.8.Seguridad (SR).....	13
5.1.9.Acceso Datos (SR).....	13
5.1.10.Connect Buzón (SR).....	13
5.1.11.Redes Sociales (SR).....	13
5.1.12.WIMF Buzón (SR).....	13
5.1.13.WIMF Back tracking.....	13
5.2.DISEÑO DE CLASES NUCLEO CONNECT	14

5.3.DISEÑO DE CLASES NUCLEO WIMF	15
5.4.DISEÑO DE CLASES BACKEND	16
6.TRAZABILIDAD DESDE EL MODELO DE DISEÑO AL MODELO DE IMPLEMENTACIÓN.....	17
6.1.CONNECT!	17
6.1.1.Bussines Logic.....	17
6.1.2.Security Controller.....	17
6.1.3.IO Controller.....	17
6.1.4.Mailbox Controller.....	17
6.1.5.Social Networks Controller.....	17
6.1.6.Server Access Controller.....	17
6.1.7.Languag Controller.....	17
6.2.WHERE IS MY FRIEND?	18
6.2.1.Bussines Logic Controller.....	18
6.2.2.Security Controller.....	18
6.2.3.Tracking Controller.....	18
6.2.4.MailBoxController.....	18
6.2.5.ServerAccessController.....	18
6.2.6.Languag Controller.....	18
7.VISTA DEL MODELO DE DISTRIBUCIÓN.....	19
7.1.DIAGRAMA DE DISTRIBUCIÓN	19
7.2.NODOS	19
7.2.1.Backend.....	19
7.2.2.Mobile WINDOWS 8.....	19
7.2.3.Mobile WINDOWS PHONE.....	19
7.2.4.Mobile ANDROID.....	19
7.2.5.Mobile IOS.....	19
7.3.CONEXIONES	19

1.1. INTRODUCCIÓN

Este documento tiene como fin, la descripción de la arquitectura de los sistemas Connect! Y Where is my Friend?.

1.2. PROPÓSITO

El propósito de este documento es proporcionar una apreciación global y comprensible de la arquitectura del sistema usando diferentes puntos de vista para mostrar distintos aspectos del sistema. Capturar y llegar a las decisiones de arquitectura críticas que han sido hechas en el sistema.

1.3. ALCANCE

Este documento aplica a la estructura base del sistema, interacciones y responsabilidades de sus componentes.

Esta principalmente dirigido a diseñadores e implementadores, así como al cliente, ya que en este proyecto se hizo énfasis en la descripción de la arquitectura.

1.4. DEFINICIONES, SIGLAS Y ABREVIATURAS.

WIMF	Where is my Friend?
4 Plataformas	Android, Iphone, Windows Phone, Windows 8
CN	Connect!
SR	Servidor

1.5. REFERENCIAS

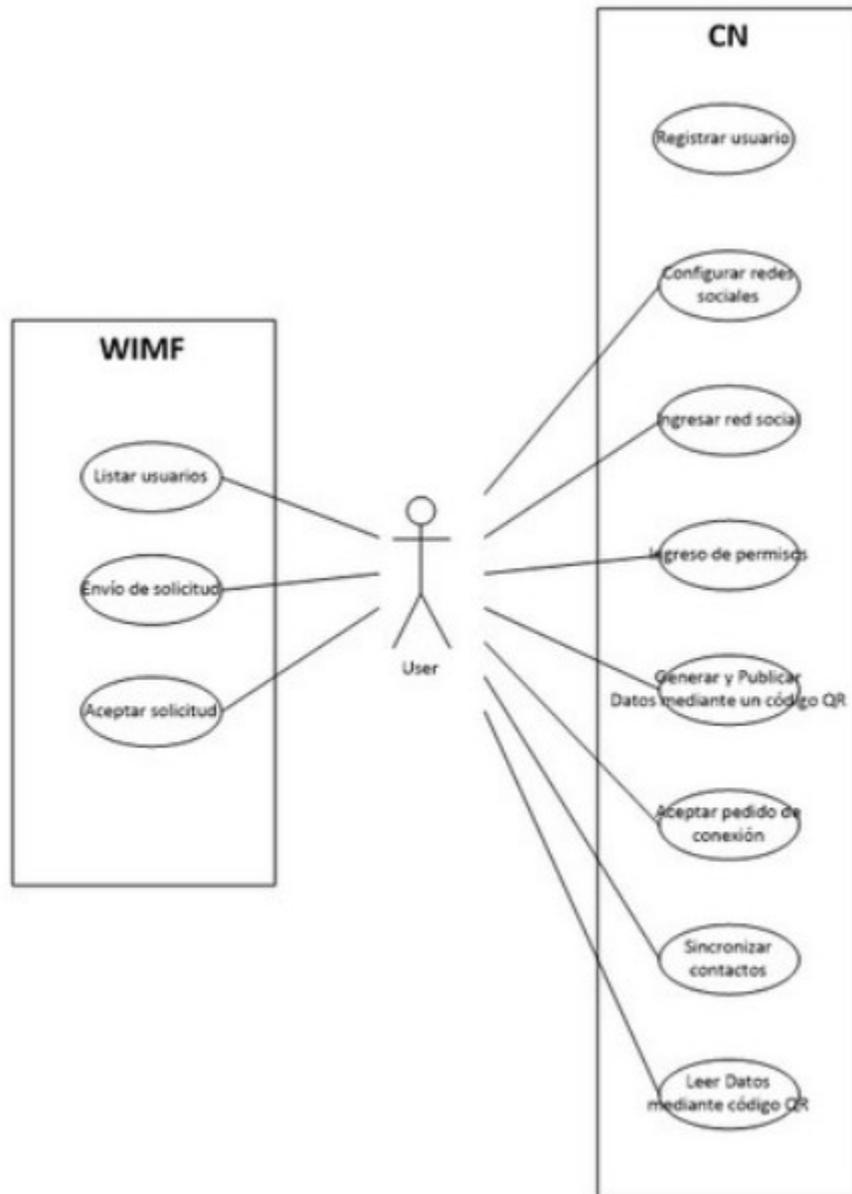
No.

1.6. VISIÓN GENERAL

El documento apunta a una visión de la arquitectura y el diseño que hacen a ambas aplicaciones, con su servidor común y sus núcleos comunes a las cuatro plataformas.

2. VISTA DEL MODELO DE CASOS DE USO

2.1. DIAGRAMA DE CASOS DE USO RELEVANTES A LA ARQUITECTURA



2.2. CASOS DE USO RELEVANTES A LA ARQUITECTURA

2.2.1. REGISTRAR USUARIO (CN)

Realiza el registro de un usuario en el sistema. La identidad del usuario también será reconocida por el sistema Where is my Friend?.

2.2.2. CONFIGURAR REDES SOCIALES (CN)

El usuario configurara las redes sociales que desea compartir por defecto al generar una conexión.

2.2.3. INGRESAR RED SOCIAL (CN)

El usuario asocia una cuenta de Twitter o de LinkedIn a su cuenta de Connect, para posteriormente publicar conexiones en dichas redes.

2.2.4. INGRESO DE PERMISOS (CN)

El usuario brindara a la aplicación Connect! Los permisos necesarios para que se puedan tomar acciones en nombre del usuario.

2.2.5. GENERAR Y PUBLICAR DATOS MEDIANTE UN CÓDIGO QR (CN)

El usuario genera un código QR el cual contiene la información necesaria para crear una conexión en connect y en las redes sociales.

2.2.6. ACEPTAR PEDIDO DE CONEXIÓN (CN)

Se acepta una conexión con otro usuario de Connect!, la conexión contiene una lista de redes sociales en las cuales se efectuara, el usuario tiene la posibilidad de recortar la lista y seleccionar solo algunas o ninguna red social.

2.2.7. SINCRONIZAR CONTACTOS (CN)

El usuario tiene la posibilidad de generar una conexión Connect! con sus contactos de las diferentes redes sociales.

2.2.8. LEER DATOS MEDIANTE CÓDIGO QR (CN)

Se le brinda al usuario la posibilidad de leer un QR generado por otro usuario, a modo de realizar la conexión entre los mismos.

2.2.9. LISTAR USUARIOS (WIMF)

Se listan todos los contactos que el usuario tiene en Connect! A modo de enviar solicitudes de tracking.

2.2.10. ENVÍO DE SOLICITUD (WIMF)

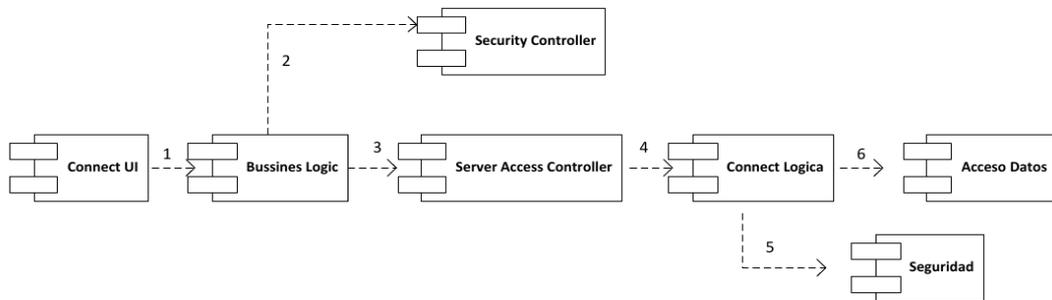
Se envía una solicitud a uno de los contactos del usuario, para poder obtener su posición durante un tiempo estipulado.

2.2.11. ACEPTAR SOLICITUD (WIMF)

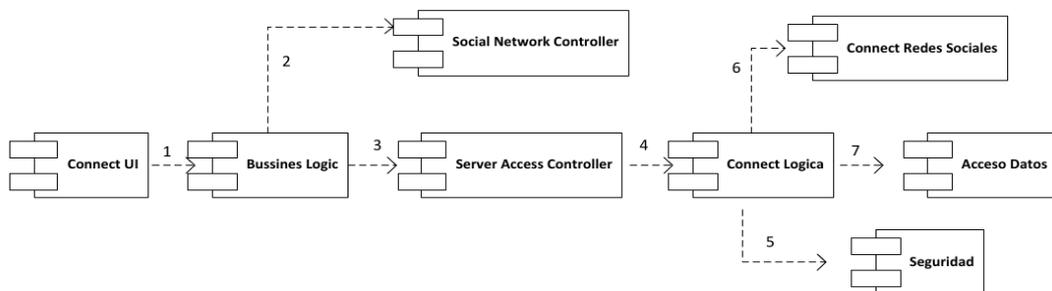
Se acepta una solicitud de tracking, enviada por un contacto del usuario, de esta manera el contacto que envía la solicitud tendrá acceso a la posición del usuario por un tiempo determinado.

3. TRAZABILIDAD DESDE EL MODELO DE CASOS DE USO AL MODELO DE DISEÑO CONNECT!

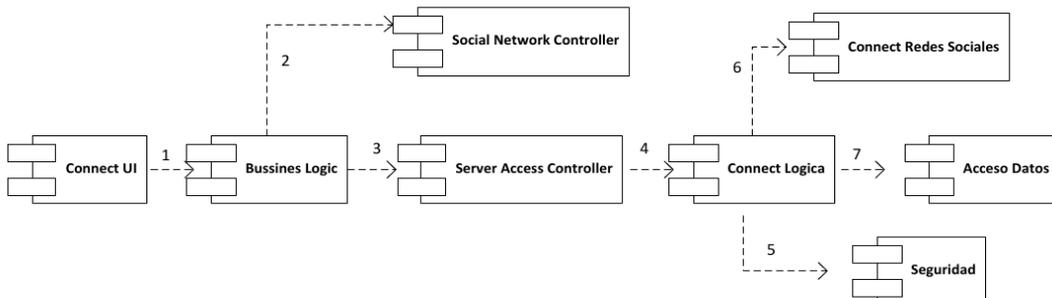
3.1.1. REGISTRAR USUARIO (CN)



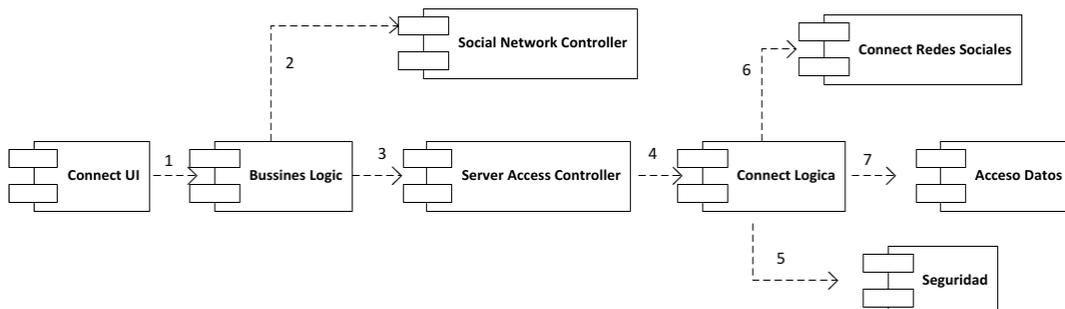
3.1.2. CONFIGURAR REDES SOCIALES (CN)



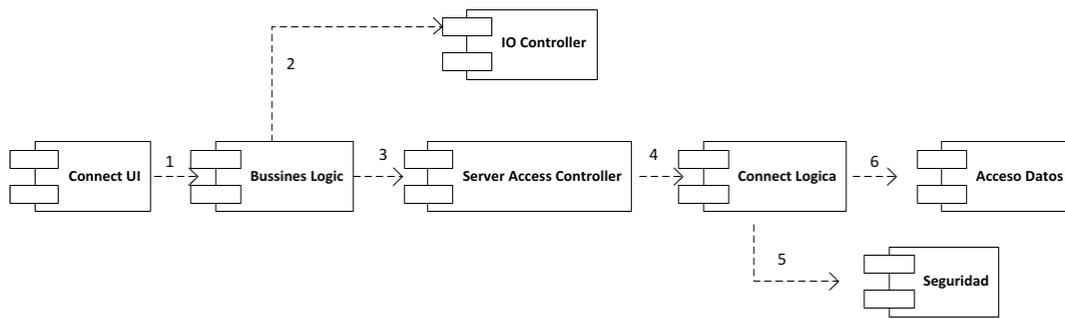
3.1.3. INGRESAR RED SOCIAL (CN)



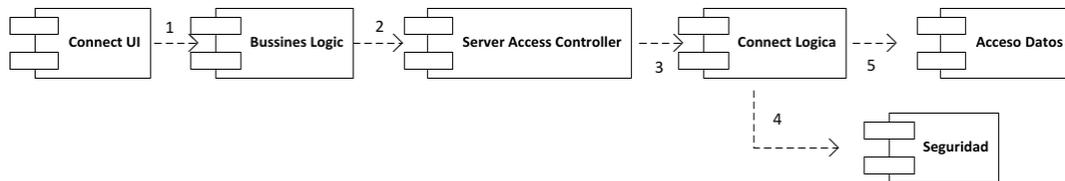
3.1.4. INGRESO DE PERMISOS (CN)



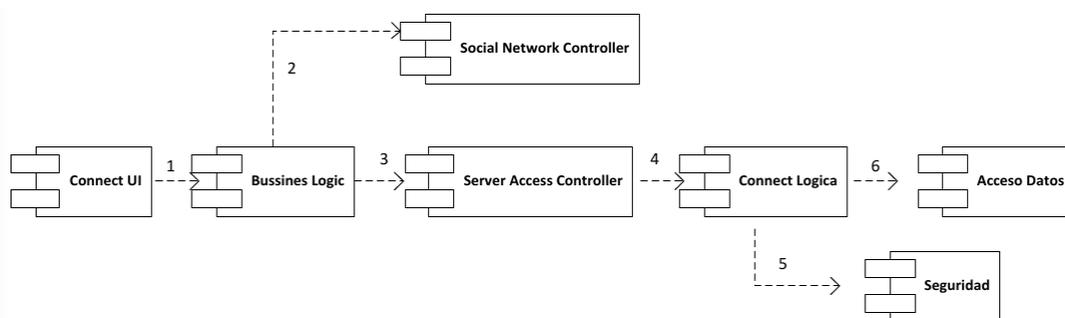
3.1.5. GENERAR Y PUBLICAR DATOS MEDIANTE UN CÓDIGO QR (CN)



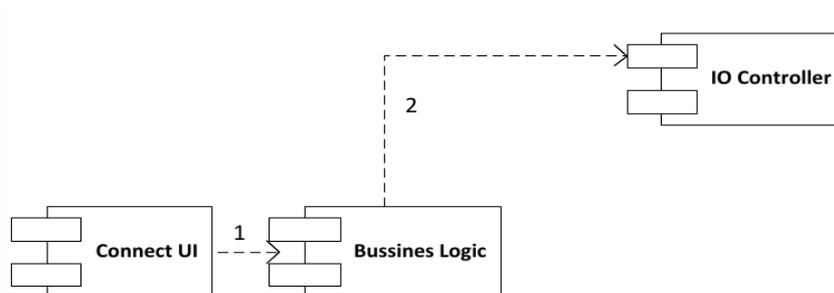
3.1.6. ACEPTAR PEDIDO DE CONEXIÓN (CN)



3.1.7. SINCRONIZAR CONTACTOS (CN)

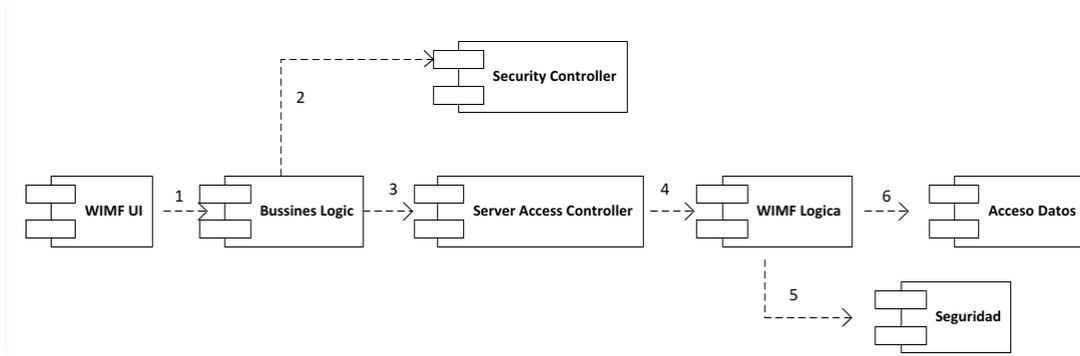


3.1.8. LEER DATOS MEDIANTE CÓDIGO QR (CN)

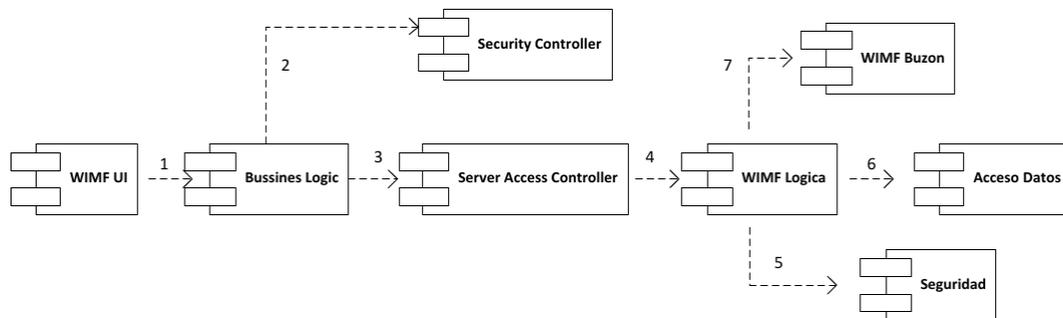


4. TRAZABILIDAD DESDE EL MODELO DE CASOS DE USO AL MODELO DE DISEÑO WIMF?

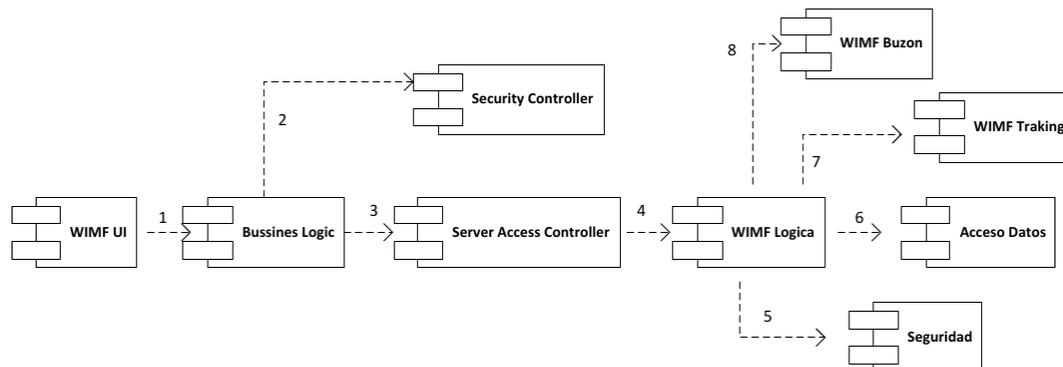
4.1.1. LISTAR USUARIOS



4.1.2. ENVÍO DE SOLICITUD

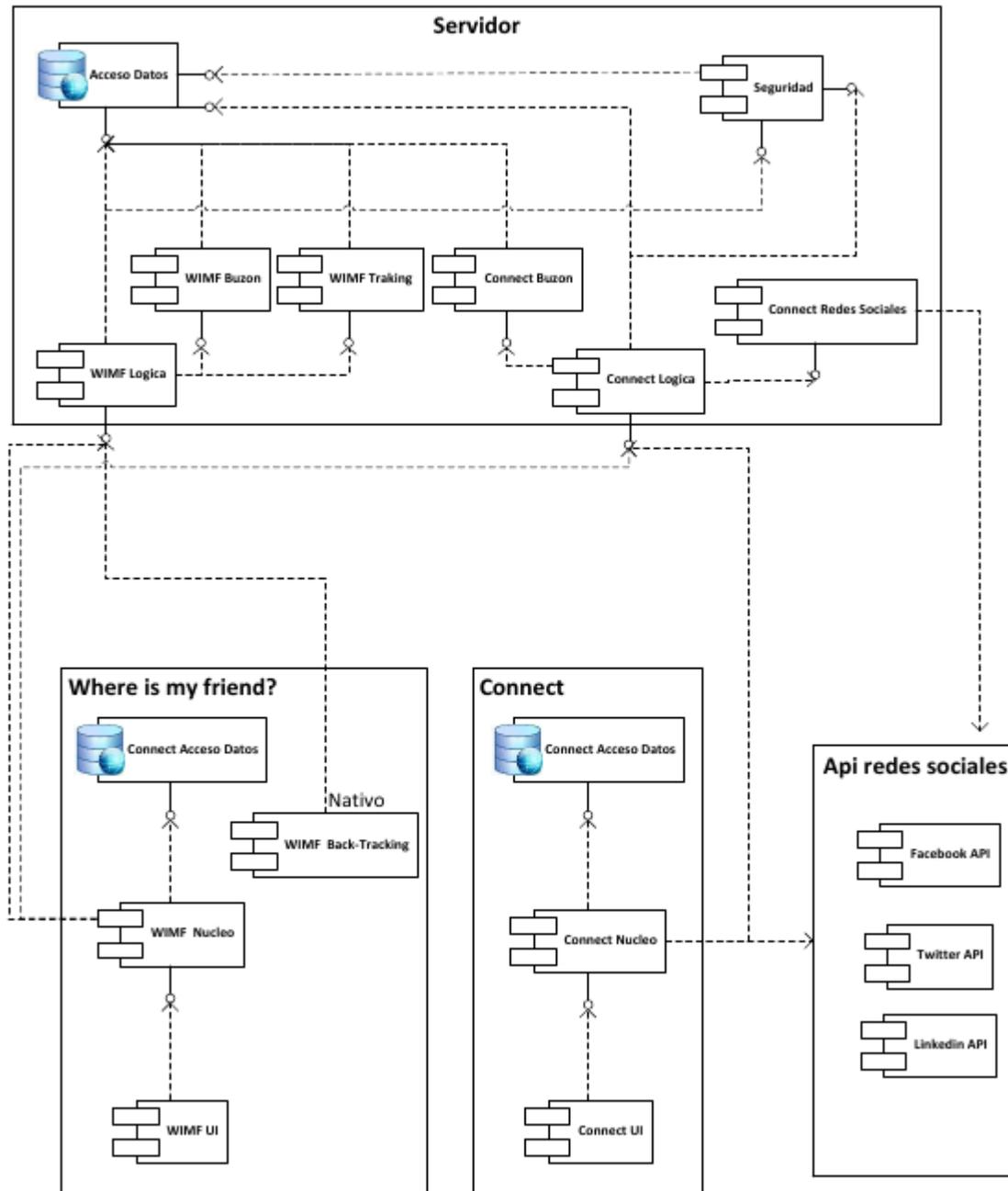


4.1.3. ACEPTAR SOLICITUD



5. VISTA DEL MODELO DE DISEÑO

5.1. DESCOMPOSICIÓN EN SUBSISTEMAS



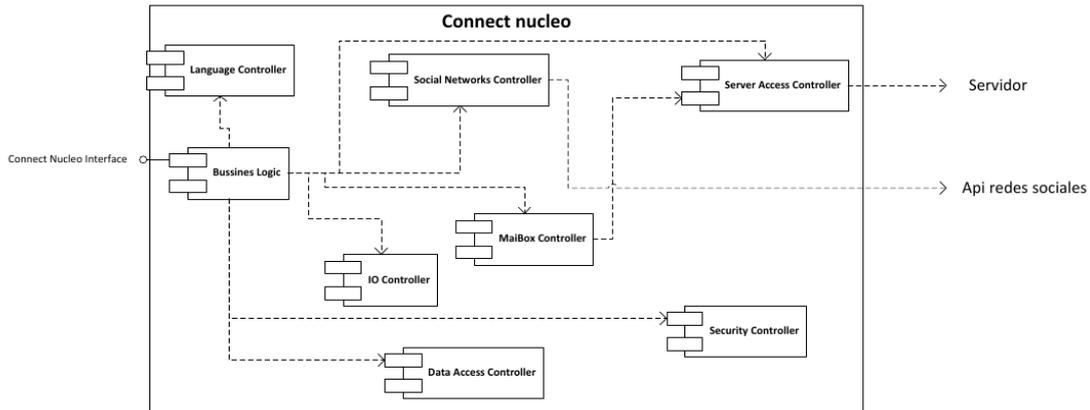
5.1.1. CONNECT UI

Componente que implementa la interface gráfica del sistema Connect!.

5.1.2. WIMF UI

Componente que implementa la interface gráfica del sistema Where is My Friend?.

5.1.3. CONNECT NÚCLEO



5.1.3.1. BUSINESS LOGIC

Componente que se encarga de implementar los casos de uso, interactuando con los restantes componentes de este subsistema.

5.1.3.2. SECURITY CONTROLLER

Componente encargado de las tareas de seguridad, las cuales incluyen registro y autenticación de usuarios, así como todo caso de uso que requiera la modificación de la identidad del usuario. Será quien mantenga los datos del usuario logueado así como su token de autenticación.

5.1.3.3. IO CONTROLLER

Componente que abstrae las funcionalidades de "publicar conexión" y "obtener conexión". En nuestro caso, implementará dichas funcionalidades utilizando una librería de códigos QR.

5.1.3.4. MAILBOX CONTROLLER

Componente encargado de la obtención de notificaciones del servidor. Este componente implementa un polling de consulta, que accede al servidor para obtener las nuevas notificaciones. Cuando una notificación es leída por el usuario, este componente se encarga de eliminarla por completo, en el móvil y en el servidor.

5.1.3.5. SOCIAL NETWORKS CONTROLLER

Componente que abstrae el acceso a las redes sociales, en el móvil se encargará de la obtención de permisos para que nuestra aplicación pueda actuar en nombre del usuario, y además del envío de solicitudes de amistad/conexión/seguimiento.

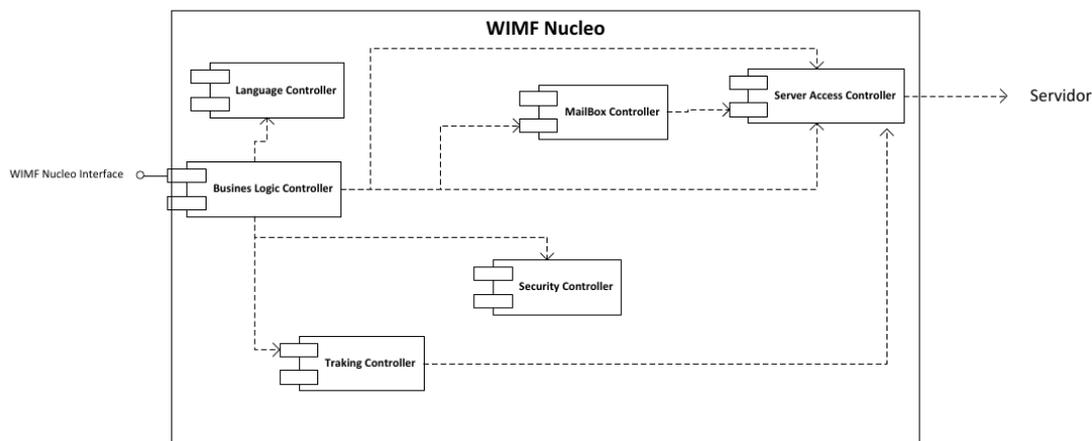
5.1.3.6. SERVER ACCESS CONTROLLER

Componente que encapsula el acceso a nuestro servidor de aplicaciones.

5.1.3.7. LANGUAGE CONTROLLER

Componente encargado de la configuración de diferentes lenguajes, para cada texto desplegado en la UI, mantiene una traducción para cada lenguaje entre los configurables.

5.1.4. WIMF NÚCLEO



5.1.4.1. BUSSINES LOGIC CONTROLLER

Encapsula todas las funcionalidades del núcleo para poder ser utilizada por la UI, como también realiza varios casos de uso o delega a otros componentes.

5.1.4.2. SECURITY CONTROLLER

Encargado de encapsular lógica respecto a la seguridad, como manejo sesión de usuario o tokens de seguridad. En nuestra arquitectura, este componente mantiene la identidad del usuario logueado, así como sus tokens de redes sociales.

5.1.4.3. TRACKING CONTROLLER

Encapsula lógica necesaria para el manejo de tracking de usuarios, como también la comunicación con el modulo "WIMF Back-Tracking". Sera el encargado de realizar polling al servidor para obtener coordenadas de usuarios a los que esta trackeando, como también el de indicar al módulo WIMF Back-Tracking que debe empezar a enviar coordenadas. Sera el responsable de los casos de uso: Envío Solicitud, Aceptar Solicitud, Ver Ubicación, entre otros.

5.1.4.4. MAILBOXCONTROLLER

Encargado de obtener mensajes (ya sean notificaciones o peticiones) del servidor, mediante polling, y almacenarlos hasta que el usuario los "lea"

5.1.4.5. SERVERACCESSCONTROLLER

Encapsula toda la comunicación con el back-end

5.1.4.6. LANGUAGE CONTROLLER

Componente encargado de la configuración de diferentes lenguajes, para cada texto desplegado en la UI, mantiene una traducción para cada lenguaje entre los configurables.

5.1.5. DATA ACCESS CONTROLLER

Componente que encapsula el almacenado y la recuperación de datos.

5.1.6. CONNECT LOGICA (SR)

Componente que provee con los servicios necesarios para el funcionamiento de la aplicación Connect!. Dichos servicios estarán publicados mediante la tecnología REST. Estará implementado en C# y expuesto como Web Service.

5.1.7. WIMF LOGICA (SR)

Componente que provee con los servicios necesarios para el funcionamiento de la aplicación Where is my Friend?. Dichos servicios estarán publicados mediante la tecnología REST. Estará implementado en C# y expuesto como Web Service.

5.1.8. SEGURIDAD (SR)

Es un componente, en el servidor, que encapsula todo el trabajo de seguridad del servidor y de las aplicaciones en general. Se separó en un componente porque es común a ambas aplicaciones. Sera el encargado principalmente del manejo de tokens y de la autenticación de usuarios mediante los mismos.

5.1.9. ACCESO DATOS (SR)

Subsistema, en el servidor, que provee las funcionalidades necesarias para el almacenamiento y la recuperación de datos.

5.1.10. CONNECT BUZÓN (SR)

Componente que almacena las solicitudes y notificaciones que deben recibir los usuarios, productos de varios casos de uso. Se encargara de la persistencia de los mismos para mayor seguridad.

5.1.11. REDES SOCIALES (SR)

Componente que tiene una mínima interacción con las redes sociales para realizar el refresco de los tokens de los usuarios.

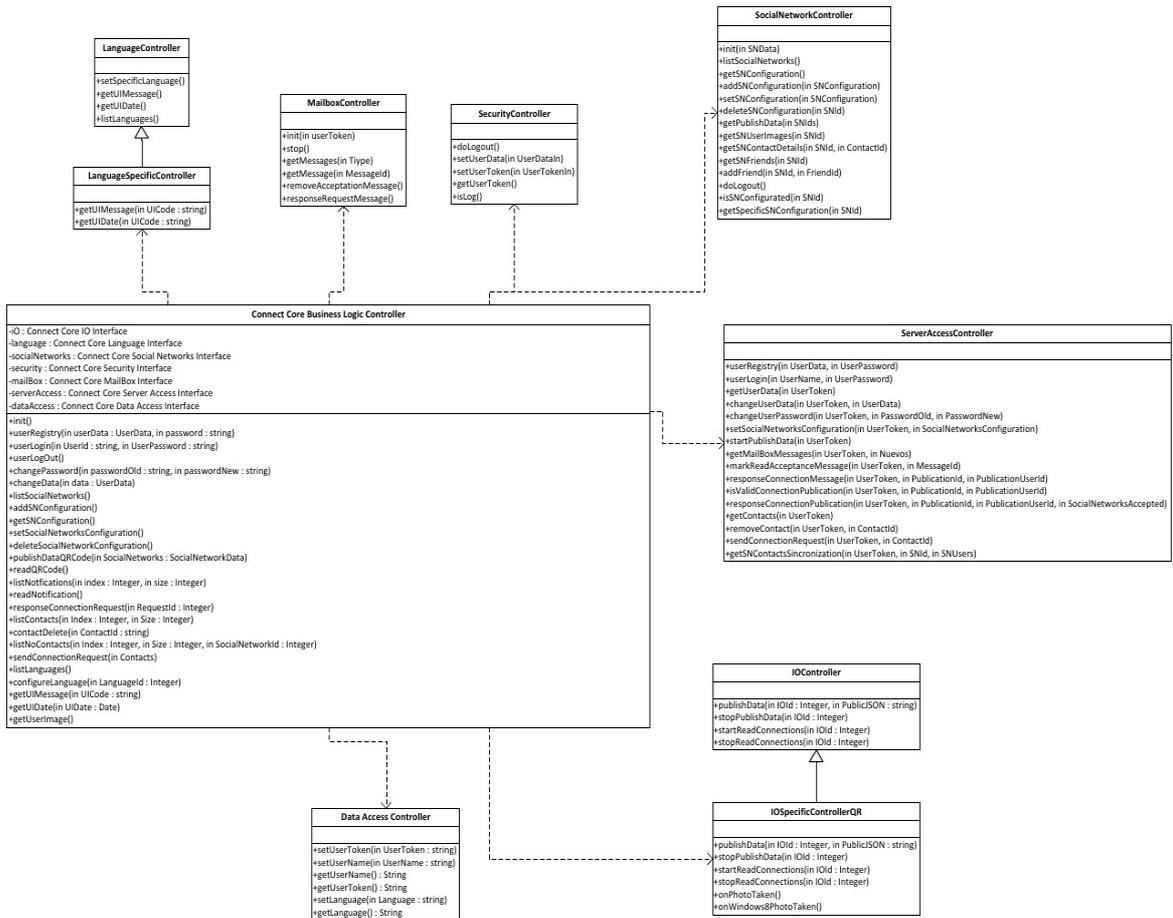
5.1.12. WIMF BUZÓN (SR)

Componente que almacena las solicitudes y notificaciones que deben recibir los usuarios, productos de varios casos de uso. Se encargara de la persistencia de los mismos para mayor seguridad.

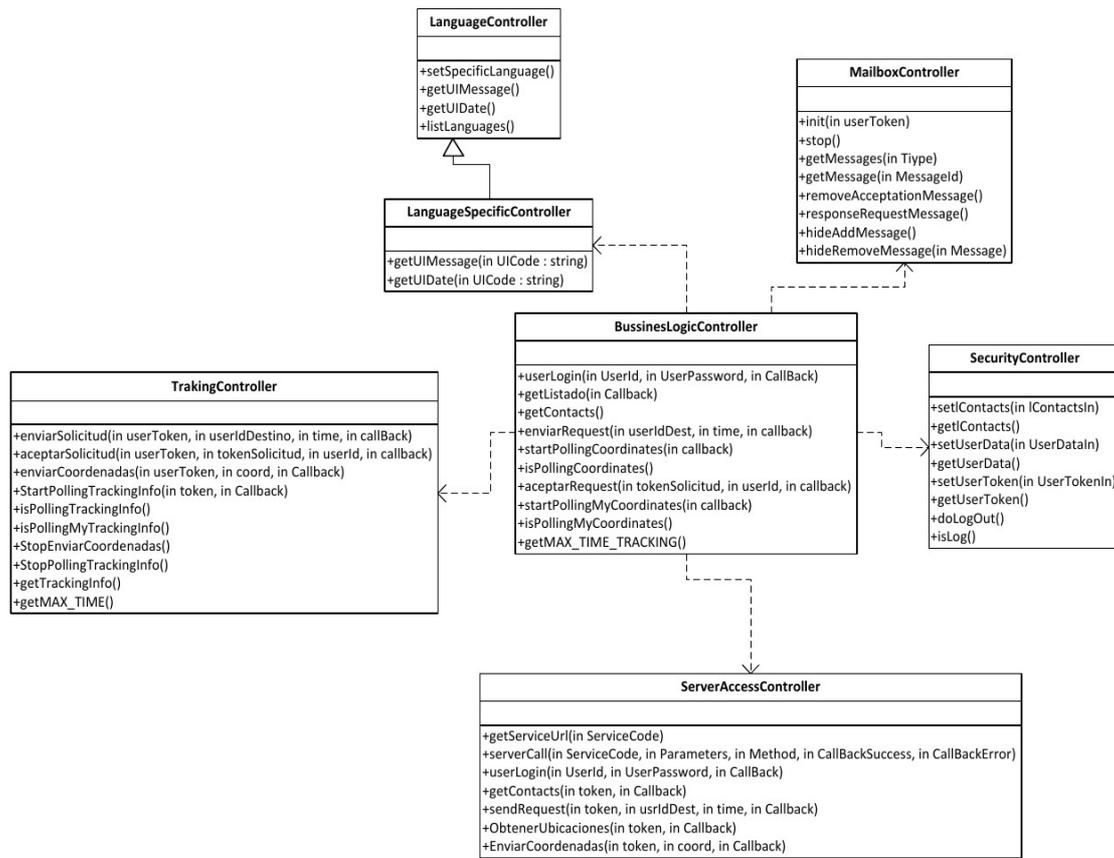
5.1.13. WIMF BACK TRACKING

Componente que se encarga de mantener los trackings activos, y de eliminarlos cuando su tiempo de vida caduca, entre otras funcionalidades. Este componente es nativo de cada plataforma, dado que debe correr en background para realizar el envío de coordenadas.

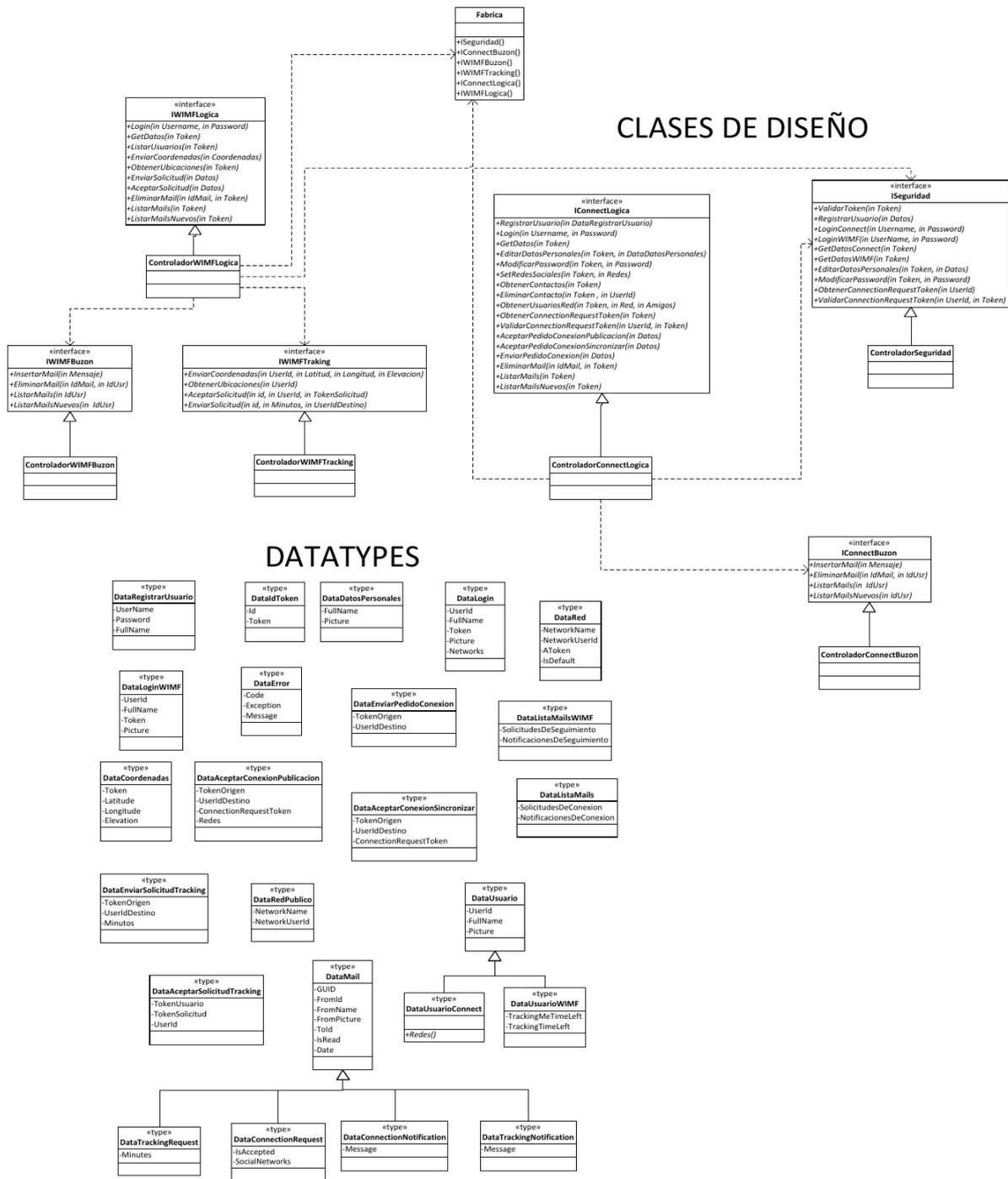
5.2. DISEÑO DE CLASES NUCLEO CONNECT



5.3. DISEÑO DE CLASES NUCLEO WIMF



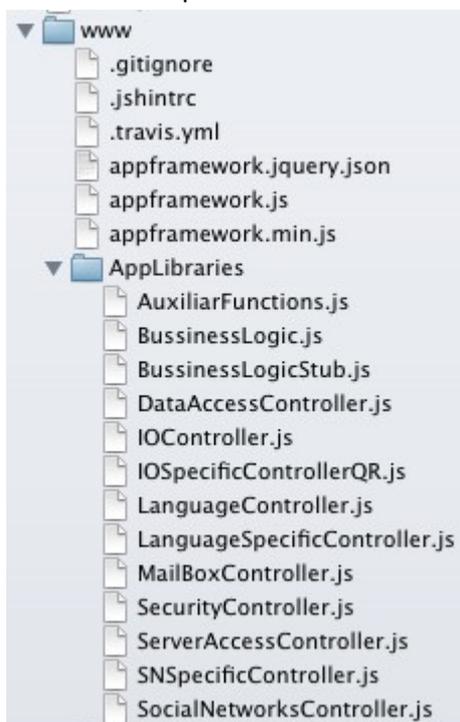
5.4. DISEÑO DE CLASES BACKEND



6. TRAZABILIDAD DESDE EL MODELO DE DISEÑO AL MODELO DE IMPLEMENTACIÓN

6.1. CONNECT!

Se muestra una imagen, del proyecto levantado en XCode, donde se observa la carpeta "www" la cual contiene el código común a las 4 plataformas, dentro de ella se encuentra una subcarpeta "AppLibraries" donde residen los archivos de implementación de las clases de la arquitectura.



6.1.1. BUSSINES LOGIC

BussinesLogic.js

6.1.2. SECURITY CONTROLLER

SecurityController.js

6.1.3. IO CONTROLLER

IOController.js

6.1.4. MAILBOX CONTROLLER

MailBoxController.js

6.1.5. SOCIAL NETWORKS CONTROLLER

SocialNetworksController.js

6.1.6. SERVER ACCESS CONTROLLER

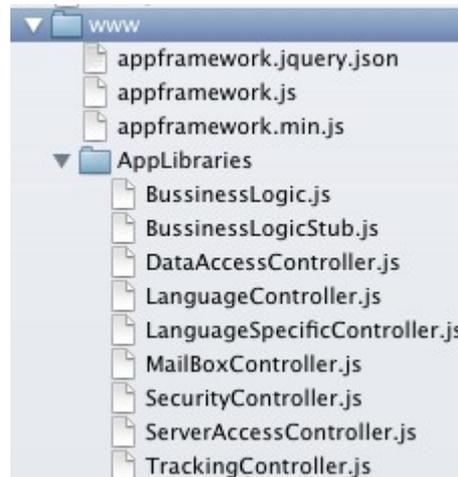
ServerAccessController.js

6.1.7. LANGUAGE CONTROLLER

LanguageController.js

6.2. WHERE IS MY FRIEND?

Se muestra una imagen, del proyecto levantado en XCode, donde se observa la carpeta "www" la cual contiene el código común a las 4 plataformas, dentro de ella se encuentra una subcarpeta "AppLibraries" donde residen los archivos de implementación de las clases de la arquitectura.



6.2.1. BUSSINES LOGIC CONTROLLER

BussinesLogic.js

6.2.2. SECURITY CONTROLLER

SecurityController.js

6.2.3. TRACKING CONTROLLER

TrackingController.js

6.2.4. MAILBOXCONTROLLER

MailboxController.js

6.2.5. SERVERACCESSCONTROLLER

ServerAccessController.js

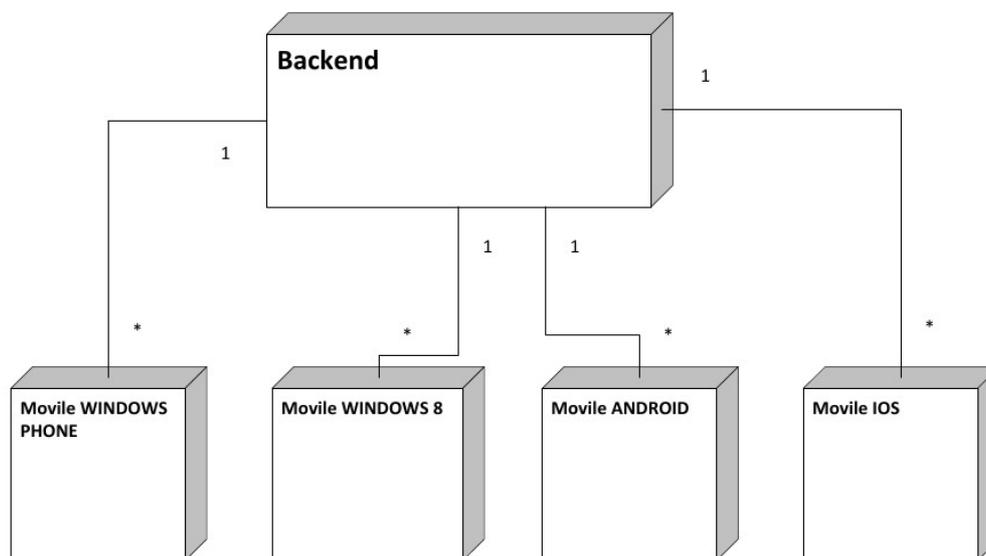
6.2.6. LANGUAGE CONTROLLER

LanguageController.js

7. VISTA DEL MODELO DE DISTRIBUCIÓN

7.1. DIAGRAMA DE DISTRIBUCIÓN

Nuestro diagrama de distribución aplica para ambas aplicaciones: Connect! y Where is my Friend?



7.2. NODOS

7.2.1. BACKEND

Es el servidor central de nuestra aplicación, residente en la nube de **Microsoft Azure**. Se encarga del manejo de datos, y de la comunicación entre los móviles, para ambas aplicaciones CN y WIMF. Gestiona los usuarios de las aplicaciones y las conexiones entre los mismos. Mantiene los trackings activos y brinda una interface REST para cada aplicación.

7.2.2. MOBILE WINDOWS 8

Aplicación de escritorio ejecutable en ordenadores y tablets con sistema operativo Windows 8.

7.2.3. MOBILE WINDOWS PHONE

Aplicación Mobile ejecutable en celulares con sistema operativo WINDOWS PHONE 8.

7.2.4. MOBILE ANDROID

Aplicación Mobile ejecutable en celulares con sistema operativo ANDROID

7.2.5. MOBILE IOS

Aplicación Mobile ejecutable en celulares con sistema operativo IOS

7.3. CONEXIONES

Todas las conexiones son mediante la internet, con el protocolo REST.