Planos arquitectura general Formiik

Versión 1.0

|  |  |
| --- | --- |
| Autores: | David Cruz |
| Fecha de creación: | 09/02/2018 |
| Fecha de última actualización: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

# 

# 

# 

# **Seguridad Móvil**

Formiik es una plataforma que se compone de diferentes piezas divididas por responsabilidades y que se organiza de la siguiente manera:

En la parte de clientes UI, se puede acceder vía web en la página, o bien desde la aplicación Android, la cual es una aplicación nativa escrita en Java. Esta aplicación Android puede a su vez interactuar con otras aplicaciones Android (apk's externas) que al estar instaladas en el mismo dispositivo, pueden extender la funcionalidad de la aplicación Android.

Los clientes UI también se comunican hacia servicios en el backend. El backend es el cómputo en la nube de Windows Azure, que puede ser sobre todo de tres tipos: Web Sites, Web Roles o Worker Roles. Los Web Sites nos permiten publicar sitios web como la página de formiik, los web roles nos permiten desplegar servicios web que pueden consumir los clientes UI, o bien las invocaciones desde servicios centrales externos (backend de nuestros clientes, la caja roja en el diagrama). En el caso de invocación hacia los clientes se hace vía REST y cuando los clientes invocan servicios de Formiik se hacen vía SOAP. Estas conexiones se hacen a través de un SSL.

Se puede ver en el diagrama que se utiliza PowerBI embedded, que nos permite hacer reportes personalizables en el portal web.

Formiik utiliza un procesamiento asíncrono para muchas tareas. Esto nos permite ser responsivos en todo momento, ya que no consumimos recursos de los threads de invocación para resolver tareas. Para lograr esto, es que se utilizan los Worker Roles. Cuando hay alguna tarea que necesita ser procesada, se encola, y los Worker Roles se encargan de leer estas colas de trabajo para atender a los procesamientos. Se tienen worker roles específicos por cada queue en el flujo de las órdenes de trabajo.

Los repositorios que se utilizan en Formiik los llamamos Storage Centers. Estos son SQL Azure como base de datos estructurada, Table Storage para tablas no relacionadas, Blob Storage para archivos, Queues para las colas de trabajo que utilizan los worker roles, Document DB como base de datos de documentos, y Firebase como base de datos en tiempo real. Cada uno de nuestros clientes tiene sus datos en un storage center específico para así poder tener los datos separados y poder escalar sin límites. Existen además unos Table Storage seguros, a los cuales solo tiene acceso privilegiadas personas en Formiik, donde se mantienen las credenciales de acceso a los repositorios de toda la aplicación. Formiik accede a estas conexiones seguras a través de un gestor de conexiones con un acceso encriptado.

Existe un mecanismo de extensibilidad en el que podemos personalizar la lógica de asignación de órdenes con las estrategias de asignación. Estas pueden cambiar y ser cambiadas en tiempo real, sin necesidad de publicar una versión.

Por otro lado, existen invocaciones de nuestro backend al backend de nuestros clientes. Esto se realiza por web services. También es posible que hagamos validación de identidad por OAuth, con lo que podemos delegar esta tarea, así como el acceso a recursos de nuestros clientes.

En la parte derecha del diagrama se muestran las invocaciones que se pueden hacer a servicios externos, como son Google Push Notifications, Google Geocoding, SendGrid para envío de correos, assignment strategies para cuando las estrategias están en un lugar externo, y Twilio para envío de SMS.

A continuación, el diagrama general de arquitectura (Diagrama de interfaces UML):

