



SYNERGY-GB

Documento de Arquitectura de Software

Notificaciones Push Enriquecidas

Table of Contents

Introducción	3
Representación Arquitectónica	3
Vista de Escenarios	4
Vista Lógica	6
Vista de Procesos	7
Vista de Desarrollo	7
Vista de Despliegue o Implantación	8
Tamaño y Desempeño	8
Calidad	9
Referencias	10

Introducción

Este documento provee una visión general de la arquitectura del sistema de notificaciones push enriquecidas móvil multiplataforma, usando distintas vistas arquitectónicas para representar distintos aspectos del sistema, brindando al lector una visión global y comprensible del diseño general del mismo.

El modelado de las vistas permitirá crear un modelo del sistema mucho más completo y constituye una base sólida para su desarrollo. En este documento se incluyen la vista de escenarios representada en el Diagrama de Casos de Uso, la vista lógica con el Diagrama de Clases, la vista de desarrollo ilustrada por el Diagrama de componentes y la vista de despliegue representada por el Diagrama de Despliegue.

Representación Arquitectónica

La Arquitectura de Software básicamente es considerada como una vista del sistema donde se incluyen sus componentes principales y la interacción y coordinación entre ellos, necesarios para cumplir con los requerimientos específicos del sistema.

En este documento se utiliza el Modelo 4+1 Vistas de Kruchten, el cual organiza la descripción de la arquitectura de software en cinco vistas (Kruchten, 1995), donde se exponen las decisiones de diseño y una quinta vista para ilustrar y validar dichas decisiones. Estas vistas se componen por la de escenarios, lógica, procesos, desarrollo e implantación. Es importante destacar que éstas son complementadas con elementos de UML que proporcionen más información a nivel gráfico. Las vistas son las siguientes:

- La vista de escenarios, también llamada casos de uso, contiene los casos de uso críticos del sistema, representados mediante el diagrama de casos de uso. Es de gran importancia ya que a través de esta vista se puede entender mejor la funcionalidad del sistema y funciona como un indicador que ayuda al diseñador a descubrir los elementos de la arquitectura durante su diseño (Kruchten, 1995).
- La vista lógica comprende las abstracciones fundamentales del sistema a partir del dominio del problema, trata principalmente los requerimientos funcionales del sistema y forma la estructura del mismo, incluye el diagrama de clases (Kruchten, 1995).
- La vista de procesos describe el diseño de concurrencia y aspectos de sincronización y muestra algunos de los requisitos no funcionales, como son desempeño, rendimiento, seguridad, fiabilidad, entre otros (Kruchten, 1995).
- La vista de desarrollo o implantación expone las características y requisitos que debe cumplir el sistema a nivel técnico (Kruchten, 1995).

- La vista de implementación engloba la distribución de paquetes de la aplicación y de manera general la distribución y funcionamiento de las distintas capas del sistema. Describe el mapeo del software en el hardware (Kruchten, 1995).

Vista de Escenarios

Para ilustrar la vista de escenarios, se incluye el diagrama de casos de uso definitivo.

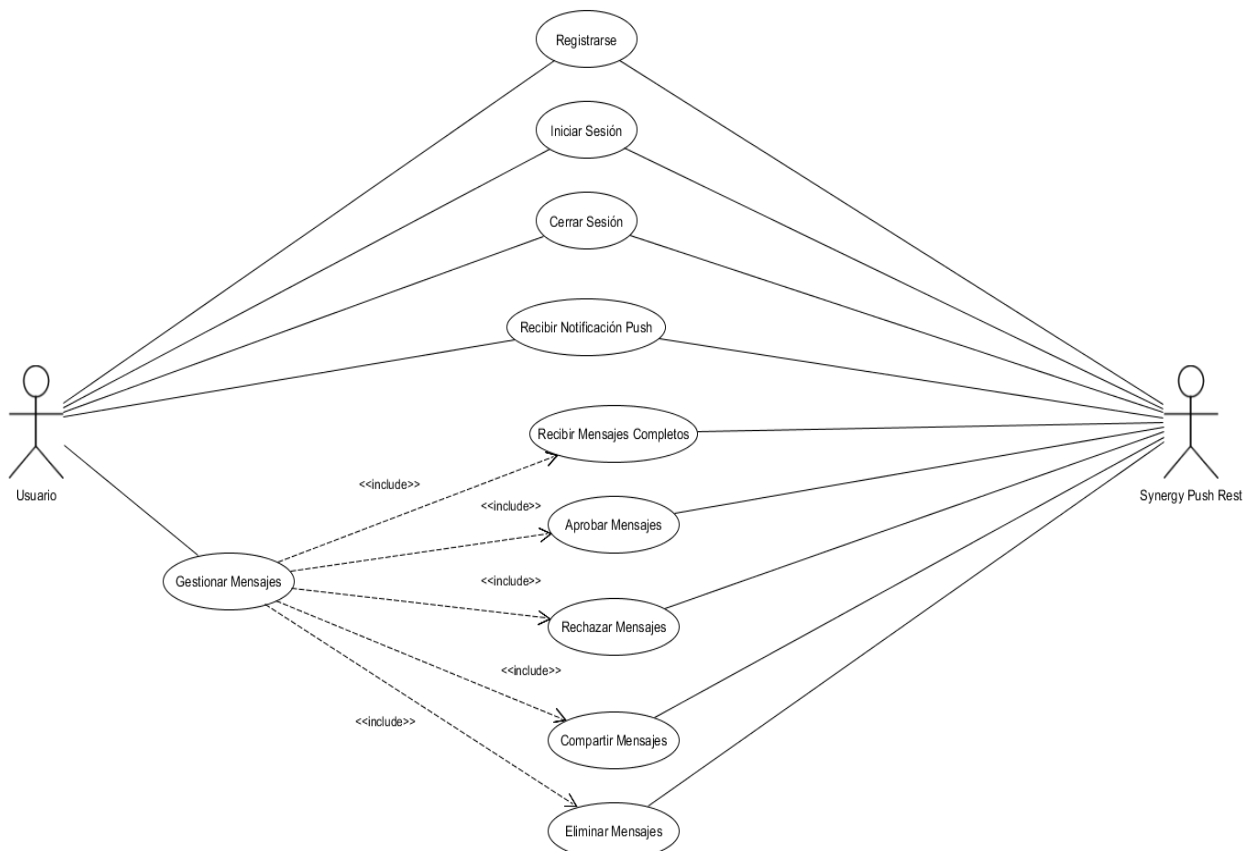


Figura 1. Diagrama de Casos de Uso tentativo. Elaboración Propia.

A continuación se presenta, la Tabla 1, donde se describen los casos de uso del sistema.

Número	Nombre del Caso de Uso	Descripción
CA_1	Iniciar Sesión	Permite a cualquier persona registrada, entrar en sesión personalizada para realizar las operaciones deseadas.
CA_2	Cerrar Sesión	Permite al usuario que posee una sesión activa, cerrarla una vez terminadas las operaciones que estuviera realizando o en cualquier momento que desee, posterior a haber iniciado sesión.
CA_3	Registrarse	Permite al usuario registrarse en el sistema.
CA_4	Recibir Notificación Push	Permite al usuario recibir notificaciones Push de Synergy Push Services.
CA_5	Gestionar Mensajes	Permite al usuario decidir qué hacer con las notificaciones recibidas.
CA_6	Recibir Mensajes	Permite al usuario recibir todas las notificaciones que se le manden a través de Synergy Push. Se hace de manera automática.
CA_7	Aprobar Mensajes	Permite al usuario aprobar notificaciones que requieran aprobación del usuario como alertas de seguridad y notificación de operaciones.
CA_8	Rechazar Mensajes	Permite al usuario rechazar notificaciones que requieran aprobación del usuario como alertas de seguridad y notificación de operaciones.
CA_9	Compartir Mensajes	Permite al usuario compartir cualquier notificación.
CA_10	Eliminar Mensajes	Permite al usuario eliminar cualquier notificación que tenga en su buzón.

Tabla 1. Descripción de Casos de Uso. Elaboración Propia.

Vista Lógica

A continuación se presenta el Diagrama de Clases.

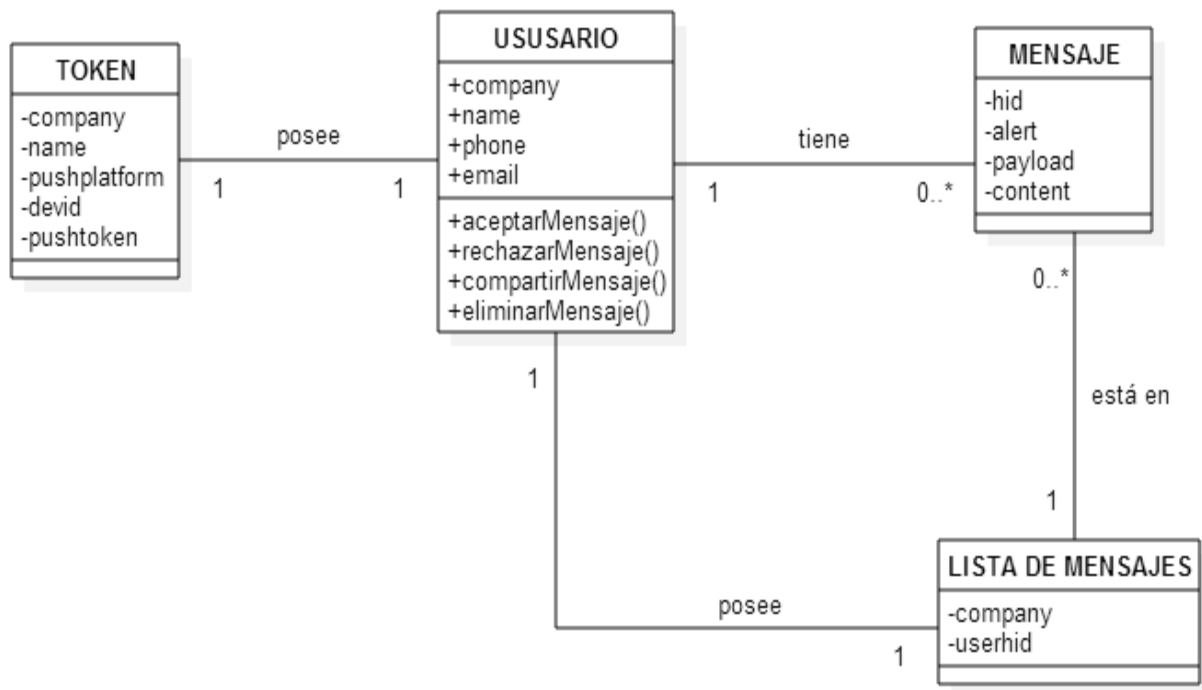


Figura 2. Diagrama de Clases tentativo. Elaboración Propia.

Vista de Procesos

La Vista de Procesos no fue desarrollada, dado que la aplicación no resuelve aspectos de concurrencia de usuarios y de datos. Los mismos son resueltos por la arquitectura Synergy Push dentro de los módulos que ofrece.

Vista de Desarrollo

La vista de desarrollo se realizó a través de un diagrama de componentes, donde se muestra la relación entre los mismos. Cabe destacar que solo los componentes más importantes están presentes en el diagrama de modo que se facilite su comprensión.

A continuación, en la figura 4, se muestra el diagrama de componentes (resumido) del sistema.

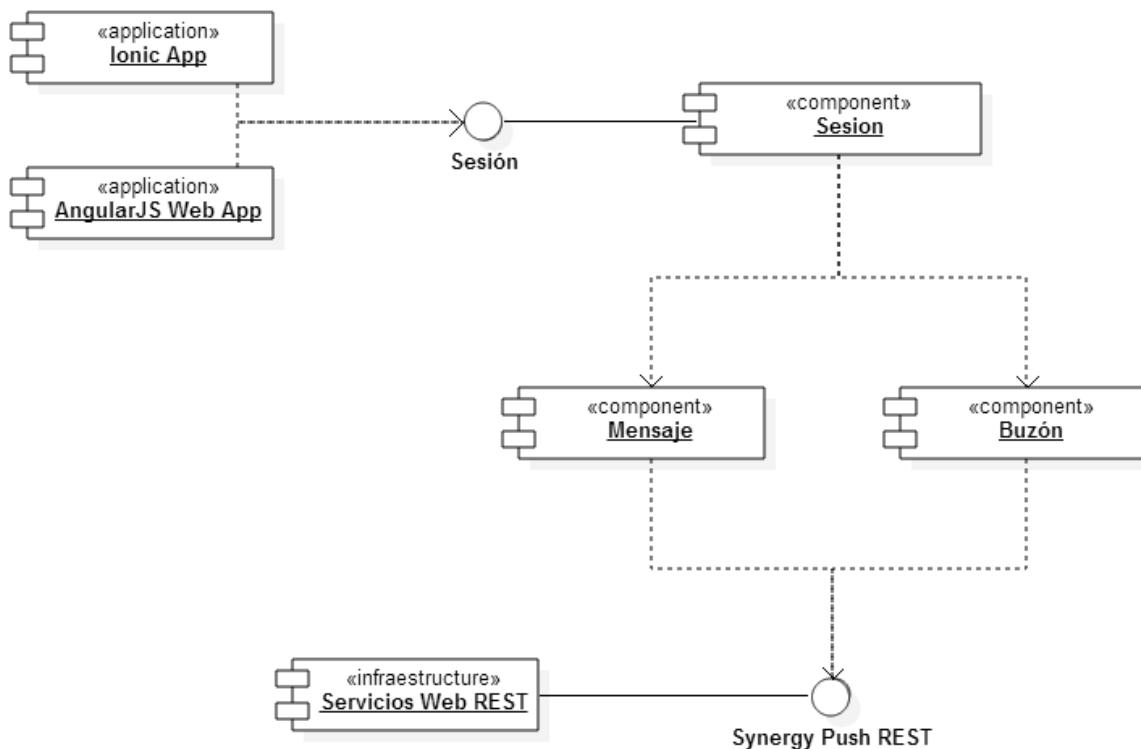


Figura 3. Diagrama de Componentes. Elaboración Propia.

Vista de Despliegue o Implantación

La vista de implantación se ve reflejada en la figura 5, mostrada a continuación.

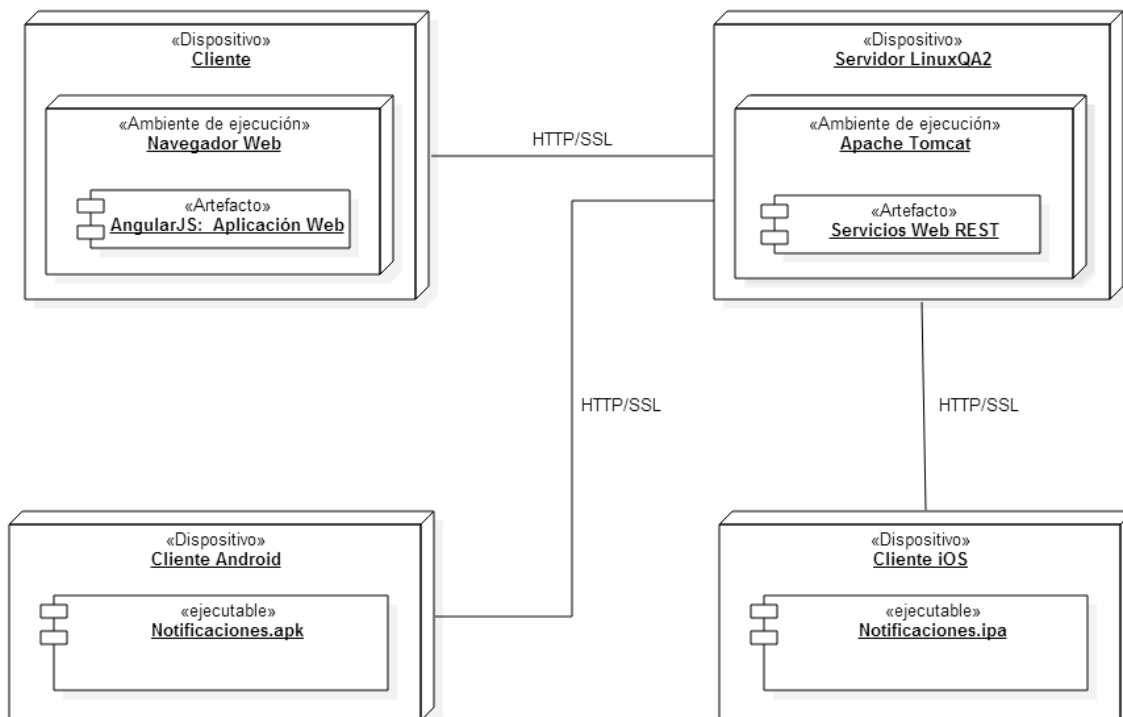


Figura 4. Diagrama de Despliegue. Elaboración Propia.

Es importante aclarar que la comunicación se realiza a través de HTTP debido a que es un prototipo, posteriormente cada una de estas se comunicará a través de HTTPS/SSL. La información que se comparte entre los nodos se encuentra en formato JSON.

Tamaño y Desempeño

- La aplicación móvil multiplataforma corre en dispositivos iOS y Android.
- Solo se permite un dispositivo por usuario y un usuario por dispositivo.
- El recurso necesario para la utilización del sistema web es un explorador o navegador de internet.

Calidad

- La arquitectura MVC (Modelo-Vista-Controlador) separa la lógica de negocio (el modelo) y la interfaz (la vista) por lo que se consigue un mantenimiento más sencillo del sistema, por ejemplo, si el sistema debe ejecutarse tanto en un navegador estándar como en un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva (y funcionará en general para cualquier dispositivo con un navegador web), sin necesidad de modificar el controlador y el modelo (Para el caso web). Incluso para el caso de la aplicación móvil aplica lo mismo, gracias a que AngularJS permite trabajar con el patrón MVC del lado del cliente en una aplicación diseñada con Ionic (En este caso con crear una sola vista, se podrá ejecutar tanto en el sistema operativo iOS como en el sistema operativo Android).
- El sistema es desarrollado como SaaS (Software as a Service), por lo que la aplicación podrá funcionar en cualquier plataforma, garantizando una alta portabilidad.
- El desarrollo de las aplicaciones está basado en módulos lo que permite ser escalable ya que permite la incorporación nuevas características para ampliar sus capacidades de manera fácil. El sistema está diseñado con esta característica para que se puedan agregar nuevos componentes sin tener que afectar lo ya existente ya que se siguieron las mejores prácticas de diseño.
- La aplicación móvil podrá desenvolverse (por ser multiplataforma) tanto en iOS como en Android y la aplicación web sólo requerirá de un navegador web.

- -

Referencias

Kruchten, P. (Noviembre de 1995). Architectural Blueprints - The "4+1" View Model of Software Architecture. *IEEE Software Magazine*, 42-50. Recuperado el 23 de Septiembre de 2013, de <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1viewarchitecture.pdf>

David Prieto. Abril de 2015. Investigación de la Arquitectura Notificaciones Push Enriquecidas.