

Reporte Técnico

Proyecto de Desarrollo de Aplicaciones Web

Rafael Antonio Comonfort Viveros
A01324276

29/11/17

Contents

1	Introducción	2
2	Definición de Arquitectura	2
3	Servicios Web	3
4	Lenguajes Utilizados	3
5	Base de datos	4
6	Tecnologías Cliente	4

1 Introducción

El presente documento, ilustra y descompone las características del proyecto elaborado para la materia de Desarrollo de Aplicaciones Web. Se dará una visión general acerca de la arquitectura, los servicios, las herramientas, sistemas y lenguajes utilizados para completarlo.

Mi proyecto consiste en un catálogo de videojuegos, donde uno puede registrarse y guardar los diversos juegos, estudios desarrolladores, compañías editoras y géneros de juegos. El usuario puede agregar, editar y eliminar estos objetos. Cada uno cuenta con distintos campos que pueden ser llenados.

2 Definición de Arquitectura

El sistema se divide en *back-end* y *front-end*. Lo primero se refiere a la funcionalidad que el usuario no puede ver, pero controla en realidad los procesos de una aplicación Web. El segundo es acerca de la apariencia visual, así como a todo lo requerido para que el *back-end* funcione con la información del usuario.

En nuestro caso, se utilizaron los servicios de **Google Cloud** como *host*, base de datos y otorgador de dominio. Esta plataforma tiene la enorme ventaja de escalamiento para servicios Web, es decir, uno paga por lo que necesita, a medida que el sistema vaya creciendo. En el proyecto, utilizamos el servicio de forma gratuita, opción que conllevó a un límite de versiones y de almacenamiento de datos.

Se utilizó la librería **JSON Web Token** (JWT) para dar una capa de seguridad al servicio. Solo usuarios autorizados pueden ingresar a él y solo el usuario que agregó una entidad puede verla y manejarla. Para controlar todas las operaciones permitidas (borrar, ver, listar, eliminar, actualizar y crear) se hizo uso del *Rest Endpoints API* de Google, permitiendo compatibilidad entre aplicaciones móviles y el servicio Web.

3 Servicios Web

Como se ha mencionado anteriormente, el sistema permite agregar, modificar y eliminar: videojuegos, desarrolladores, editores y géneros. Además, estos objetos son exclusivos de los usuarios que los crearon. Desde luego, esto significa que puedes iniciar sesión con tu usuario y contraseña o en su defecto, registrar un nuevo usuario. Sin embargo, note que sólo un usuario existente puede crear otro, esto por motivos de simplicidad. En un servicio Web real, uno no permitiría esta clase de decisiones de diseño por las evidentes riesgos de seguridad que implican.

4 Lenguajes Utilizados

Para el *back-end* y la *endpoints api*, se utilizó el lenguaje de programación Python. La arquitectura de Google Cloud solicita que se defina un *modelo*, es decir, un objeto que represente lo que se va a guardar en la base de datos. Estos fueron definidos para los 4 objetos en cuestión, cada uno con diferentes campos. Las funciones de inserción, listado, entre otras, requieren tener un tipo de mensaje de entrada y salida definidos. Se hicieron objetos *Message* para actualizar, agregar y listar cada modelo. Por último, se hicieron objetos *Handler* para todas las funciones del sistema que requerían su propias URL y archivos HTML.

Para el *front-end*, se utilizó HTML para crear la distribución visual, las formas para obtener datos del usuario y las secciones para mostrarlos en las páginas Web. Así mismo, fue a través de JavaScript que se proveía funcionalidad a estas, principalmente, al hacer una request con JQuery a los *endpoints api* correspondientes.

Finalmente, se utilizó el lenguaje de programación Java en la creación de la aplicación cliente para Android. Afortunadamente, Google provee un comando en su consola que traduce todo el código Python de los endpoints, a Java. De esta forma, podíamos hacer las configuraciones necesarias y cambios menores dentro de **Android Studio**, respetando los archivos generados por dicho comando. Al modificar las URL para que coincidan con las de nuestro sitio desplegado ya en los servidores de Google Cloud, se podía construir una

aplicación en Android que permitiese hacer las mismas llamadas a funciones, que haría el navegador Web con JavaScript.

5 Base de datos

Como se ha mencionado previamente, se utilizaron los servicios de Google Cloud para almacenar todo lo relacionado con la aplicación Web desarrollada. Específicamente, hablamos de 2 componentes en particular:

- **Blobstore:** Permite guardar objetos de datos, llamados *blobs*, que son mucho más grandes que los objetos en el servicio de Datastore. Son particularmente útiles para guardar video o imágenes. En efecto, las imágenes de los juegos, los logos de desarrolladores y otros, son almacenados como "blobs".
- **Datastore:** Se trata de una base de datos NoSQL construida para escalar automáticamente, tener alto desempeño y facilitar el desarrollo de aplicaciones. Los modelos definidos en el proyecto, son almacenados de esta forma.

6 Tecnologías Cliente

Hubo 2 tecnologías utilizadas para actuar como clientes de la aplicación. La primera, naturalmente, se trata del cliente Web, el cual se accede a través de un navegador de Internet. Su comunicación con la endpoints api se constituye a través de los archivos HTML y JavaScript generados. La segunda, se trata de la aplicación para dispositivos Android. Con los archivos Java generados a partir del back-end Python, se pueden crear *Actividades* de la aplicación que sirvan el mismo propósito que el front-end web. El prototipo creado para el proyecto permite iniciar sesión y posteriormente, agregar un nuevo videojuego a la base de datos. Estos procesos responden a los métodos endpoint de *users_login* y *videogame_add* respectivamente. Los cambios se ven reflejados inmediatamente en el cliente Web.