

---

# PROYECTO ARQUITECTURA Y DISEÑO DE SISTEMAS

---

Entrega 4



## **Integrantes**

MASSETTI, Matías

MOYANO, Juan Francisco

SUPERI, Agustina Micaela

## Introducción

---

El objetivo principal de este proyecto consiste en implementar un sistema con la finalidad de buscar sinónimos de términos, a través de un servicio externo “BigHugeLabs”, los cuales son ingresados mediante una aplicación móvil en el sistema operativo Android.

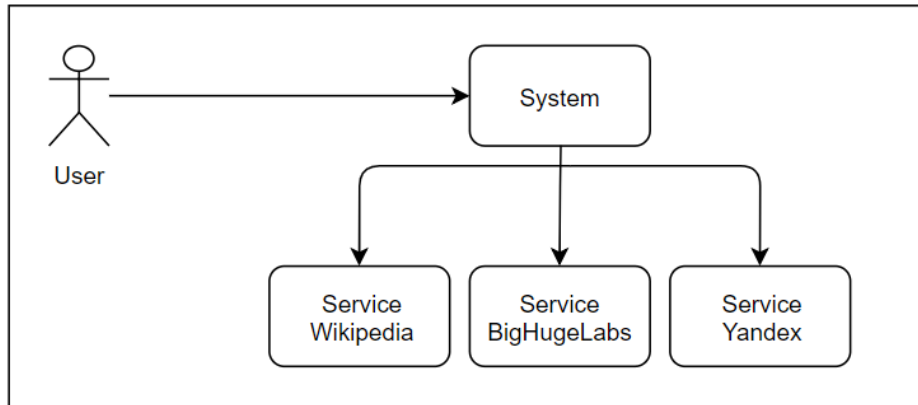
Como requisito final, nuevas funcionalidades fueron añadidas al sistema, además de poder buscar sinónimos de término, también es posible obtener su traducción, a través del servicio “Yandex”, y su definición, a través del servicio “Wikipedia”.

En el siguiente informe, dicho funcionamiento podrá ser visualizado, para mayor entendimiento, a través de los siguientes diagramas:

- ✓ **Diagrama de Contexto:** representa las entidades externas con las que interactúa el sistema.
- ✓ **Diagrama de Descomposición:** describe la organización del código. Muestra cómo las responsabilidades del sistema son particionadas
- ✓ **Diagrama de Uso:** muestra cómo los módulos dependen unos de otros.
- ✓ **Diagrama de Asignación:**

## Diagrama de Contexto

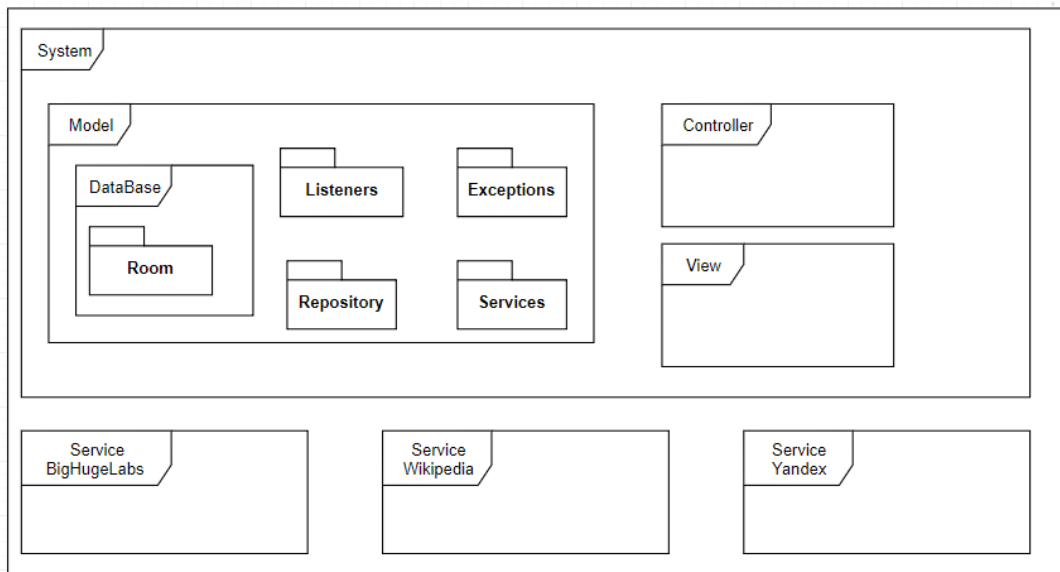
---



- ❖ **User:** usuario del sistema que utiliza la aplicación.
- ❖ **System:** contiene la lógica de la aplicación.
- ❖ **Service BigHugeLabs:** encargado de proveer sinónimos del termino que se desea buscar cuando el sistema lo solicita.
- ❖ **Service Wikipedia:** encargado de proveer la definición del término que se desea buscar cuando el sistema lo solicita.
- ❖ **Service Yandex:** encargado de proveer la traducción del término que se desea buscar cuando el sistema lo solicita.

El usuario del sistema, cualquier persona que desee buscar sinónimos de un término, su traducción o definición a través de una aplicación móvil. El sistema en caso de ser necesario utiliza un servicio externo que provee la información requerida cuando no es encontrada internamente.

## Diagrama de Descomposición



El sistema siguiente está basado en un patrón MVC (Model–View–Controller):

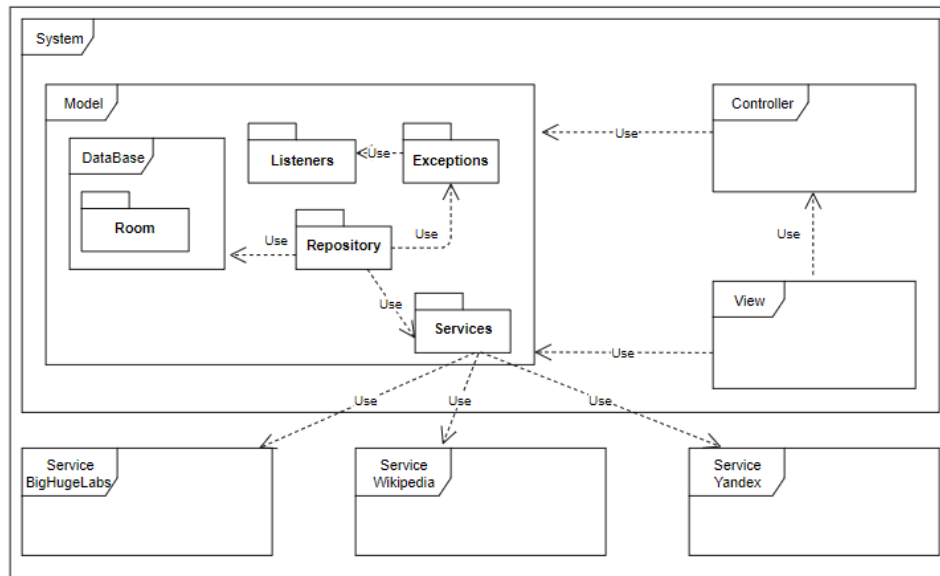
- El **modelo** que contendrá el estado de la aplicación, para ser consultado. Internamente, tiene dos formas de obtener la información requerida: a través de una base de datos, para términos ya buscados, o a través de servicios externos, es decir:

- **BigHugeLabs:** para obtener sus sinónimos.
- **Yandex:** para obtener su traducción.
- **Wikipedia:** para obtener su definición.

Se divide en los siguientes submódulos:

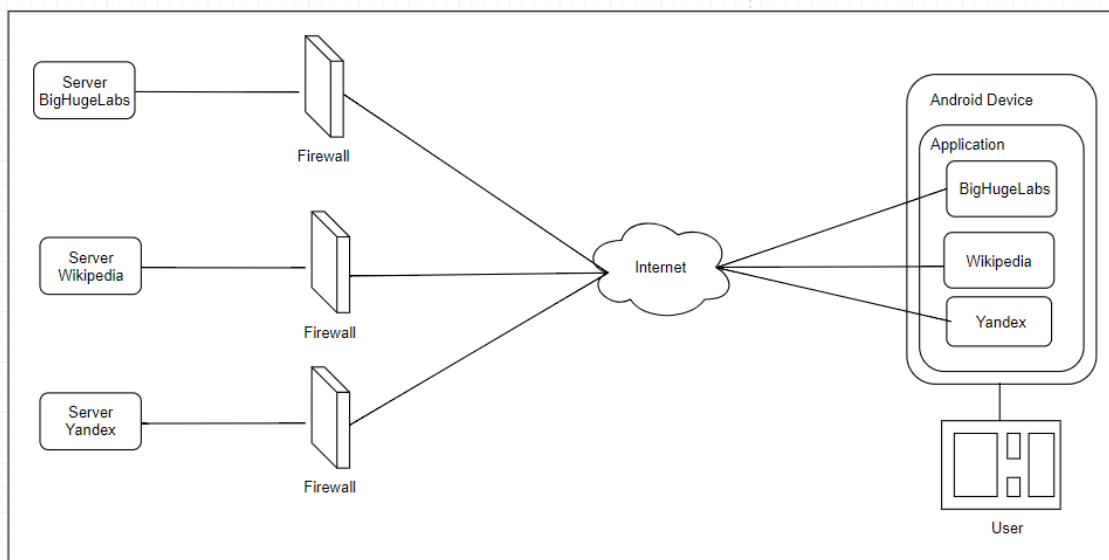
- ❖ **Repository:** tendrá la responsabilidad de la búsqueda a partir de un término.
  - ❖ **Exceptions:** es el encargado del manejo de situaciones anormales, como excepciones. Contempla los distintos tipos de excepciones disparadas con sus respectivos mensajes adecuados, y también, las medidas que deben tomarse cuando una de ellas es detectada.
  - ❖ **Listeners:** Los oyentes son los encargados de administrar los eventos significativos que se pueden producir dentro de nuestro sistema. Quienes quieran ser notificados de la ocurrencia de un cierto evento, deberán suscribirse al oyente, quién será responsable de notificar el mismo.
  - ❖ **Services:** es el encargado de la búsqueda del término en los diferentes servicios externos, si dicho termino no es encontrado internamente.
- La **vista** será el medio por el cual el usuario podrá percibir la funcionalidad de la aplicación e interactuar con ella.
  - El **controlador** es el encargado de establecer el comportamiento en sí de la aplicación, mediante la interpretación de ciertas acciones del usuario con la aplicación.

## Diagrama de Uso



El repositorio, quien es el encargado de buscar el término, deberá conocer el módulo de la base de datos. Una vez que reciba el término, optará por buscar en la base de datos si el significado del mismo ya fue buscado. En caso de éxito, lo recupera al instante de la misma. Por otro lado, si es un término “nuevo”, el significado intentará encontrarse a través de distintos servicios externos, para llevar a cabo esto, serán necesarios adapters para cada uno de ellos. Para contemplar situaciones anormales, el repositorio hace chequeos antes de buscar la palabra y después de encontrar su significado. Para esto, utiliza excepciones personalizadas y delega la responsabilidad de la rutina de comportamiento de la aplicación al mismo módulo, a través de un manejador de excepciones. Para saber cuándo se encuentra el término, la vista se encarga de crear los oyentes para el modelo. Una vez que se encuentra un término o se produce un error, la vista es notificada a través del oyente del modelo, y en base a lo ocurrido, mostrará mensajes por pantalla (tanto del error ocurrido o del significado encontrado).

## Diagrama de Asignación



El usuario ejecuta la aplicación móvil en el sistema operativo Android para buscar sinónimos de un término, su definición y su traducción. La aplicación utiliza librerías provista por servicios externos dependiendo de lo que se desea buscar, es decir, BigHugeLabs para obtener sinónimos del término, Yandex, para obtener su traducción y Wikipedia para obtener su definición, quienes se conectan a internet de forma segura a través de un Firewall para poder llevar a cabo dicha búsqueda.