**Desarrollo de software basado en componentes.**

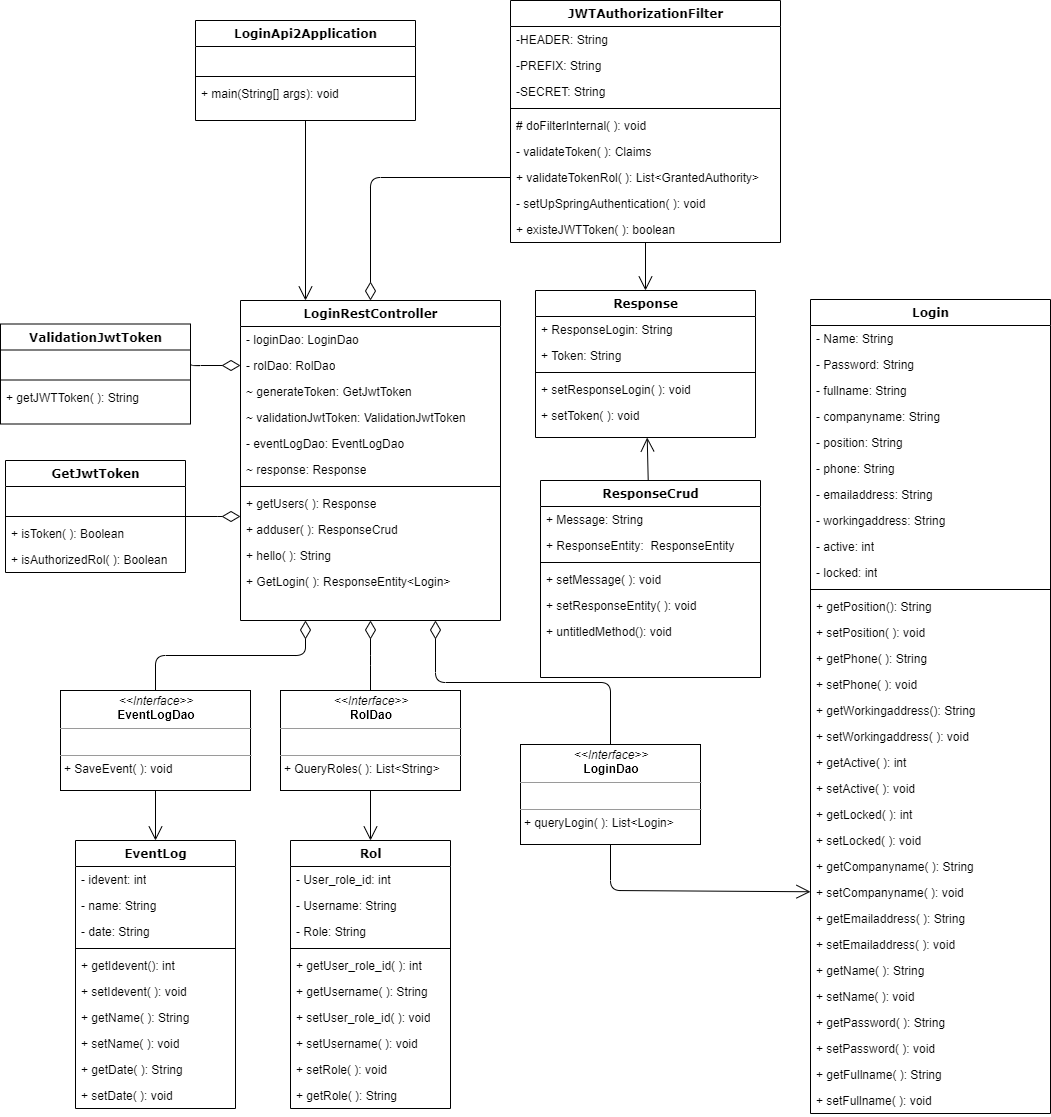
**Taller bimestral (Backend).**

**Integrantes:**

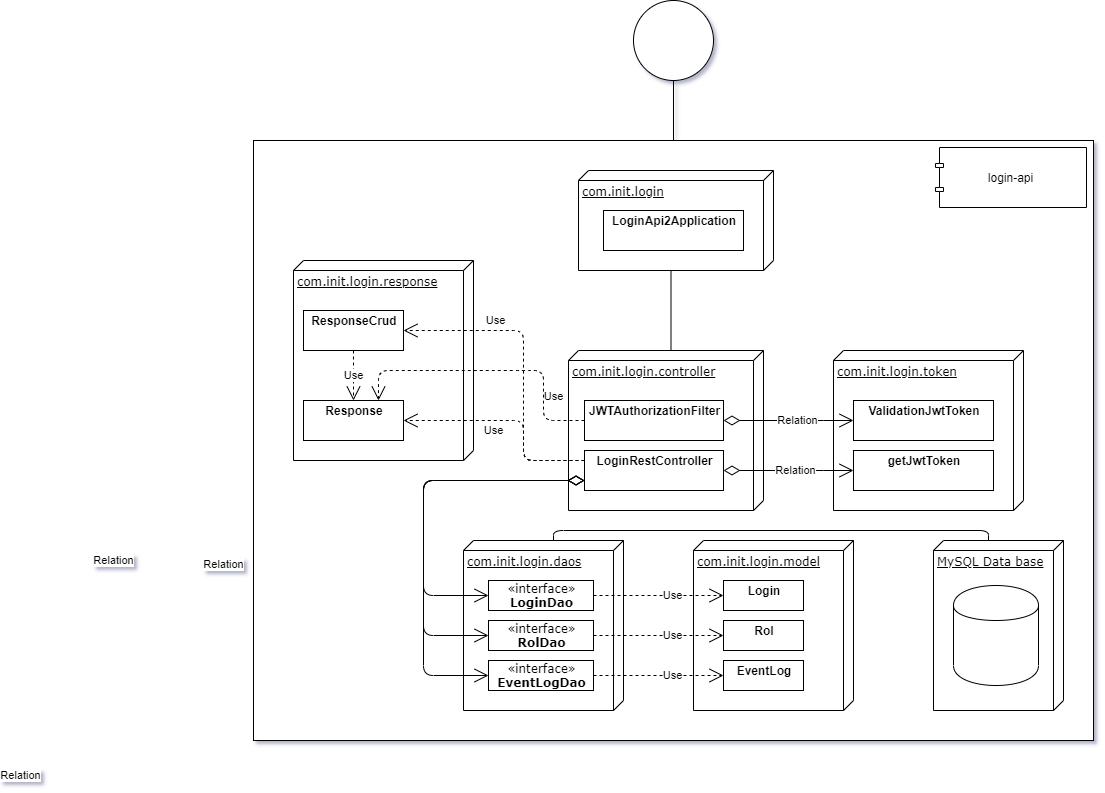
* **Duvan Arlen Corrales**
* **Martin Julián Osorio**

**Especificación de componente**

1. **Nombre de componente:** LoginAPI
2. **También conocido como:** User management component
3. **Propiedades**
   1. **Tipo:** El componente desarrollado es un API implementado mediante clases, las cuales están construidas representado el comportamiento del sistema; de esta forma se enmarca dentro de un tipo comportamental.
   2. **Subtipo:** Unidad arquitectural que presenta sus funcionalidades enmarcadas en clases, las cuales están relacionadas entre sí por medio de mecanismos como herencia, composición, agregación e implementación de interfaces. El componente se desplegará en un servidor web que soporte Java y proveerá su interface mediante mensajes en formato JSON.
   3. **Nivel:** Unidad arquitecturalexpresada en forma de API Rest.
4. **Propósito y contexto:** El API prestará el control de acceso de usuarios en el sistema que sea implementado, para esto cuenta con su propia base de datos MySQL embebida, lo que permite la fácil reutilización en múltiples sistemas diferentes, su instalación debe realizarse en un servidor web que soporte Java y mediante mensajes tipo JSON interactuará con los demás componentes del sistema en el cual sea implementado.
5. **Problema:** Controlar el acceso a cualquier sistema utilizando el mecanismo de identificación “usuario-contraseña” mediante un componente reutilizable, el cual debe ser construido en los lenguajes de programación Java o PHP, dicho control debe realizar una auditoría de cada acceso al sistema, bloquear al usuario en caso de que supere el límite de intentos fallidos de acceso además de segmentar los usuarios mediante roles que tengan asociados permisos que definirán sus posibles acciones dentro del sistema.
6. **Aplicabilidad:** El uso del componente login-api trae consigo como ventaja el desacople de la funcionalidad de control de acceso a cualquier sistema, esto siempre y cuando se despliegue en una plataforma que soporte Java y se comunique mediante mensajes tipo JSON, el API tiene una funcionalidad estándar de control de usuarios y permisos con un modelo de datos ya establecido, de esta forma podría considerarse como desventaja que el sistema que desee emplearlo debe adaptar su modelo de autorización de usuarios al que tiene el api.
7. **Descripción**
   1. **Estructura de la solución:** ElAPIcreado está compuesto de clases, su estructura está expresada mediante el siguiente modelo:



* 1. **Estrategia de solución:** El API está segmentado por paquetes que a su vez contienen clases, a continuación se presenta el modelo de componentes con su interfaz proveída:

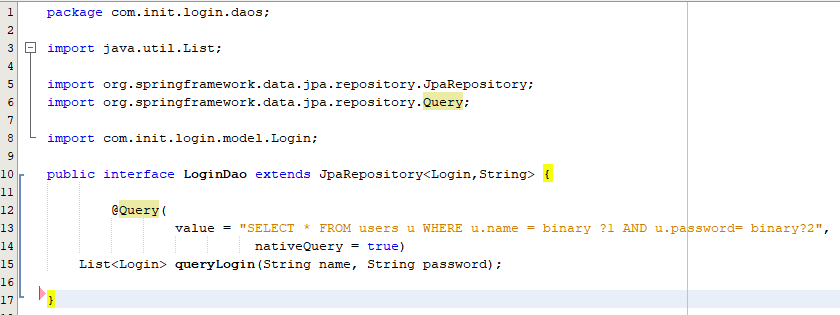


* 1. **Implementación:** El proyecto generará un archivo .jar que será instalado en un servidor web que soporte java (Mínimo debe contar con Java versión 8), Maven o Gradle y Spring boot, luego de instalarlo debe ser levantado como un servicio web; de esta forma el API queda expuesta por medio de sus URIs definidas para cada funcionalidad a prestar.

1. **Interfaces:** Las interfaces provistas bajo el estándar RESTful por el API login-api son las siguientes:

* **Login:** Haciendo uso del método GET, <http://localhost:8080/getlogin>
* **Agregar usuario:** Haciendo uso del método POST, <http://localhost:8080/adduser>
* **Cambiar contraseña:** Haciendo uso del método PUT, <http://localhost:8080/changepassword>
* **Generar listado de usuarios:** Haciendouso delmétodo GET, <http://localhost:8080/queryuser>

1. **Motivación:** La presentada en el documento es un API que gestiona la data de los usuarios de forma segura, brindando una centralización de datos que puede usarse en múltiples aplicaciones sin que están conozcan directamente la información personal del usuario. Es un componente reutilizable y totalmente desacoplado que puede utilizarse de forma natural (siguiendo las limitaciones expresadas anteriormente) en n sistemas que deseen controlar el perfil y funcionalidades a las que debe tener acceso un usuario.
2. **Características y subcaracterísticas de calidad abordadas:** Al ser un componente empleado para controlar acceso de usuarios a determinado sistema, dichos datos de acceso se enmascaran en tokens encriptados, los cuales contienen los perfiles (Roles y permisos asignados) de cada usuario; de igual forma el API dentro de su funcionalidad estándar bloquea de forma automática a los usuarios que alcanzan el límite de 3 intentos fallidos de acceso al sistema y limita las acciones de cada usuario identificado en el sistema mediante roles y permisos.
3. **Código de ejemplo:** A continuación se presenta un segmento de la consulta realizada en la base de datos para acceder al sistema:



1. **Variantes de componente:** El componente se debe desplegar en una plataforma que soporte Java, al ser una API RESTful se comunica mediante peticiones HTTP haciendo uso de sus verbos: GET, POST, PUT, DELETE, entre otros. Al adoptar un estándar como HTTP se logra comunicar con muchas aplicaciones lo que la hace altamente acoplable y al tener su propia base de datos embebida es un componente totalmente independiente.
2. **Componentes relacionados:** Un componente utilizado en la autenticación de usuarios y control de permisos es el API OAuth, un api basada en HTTP que se ha convertido casi en un estándar para delimitar accesos a usuarios, segmentar los datos a los que tienen acceso e incluso hacer trazas de las peticiones end-to-end identificando los actores involucrados en el flujo de información. Todo esto lo realiza de forma desacoplada y brindando una fácil implementación en variados sistemas, desde aplicaciones web, hasta aplicaciones móviles e incluso aplicaciones de IOT. Para más documentación sobre OAuth visite <https://oauth.net/2/>.