

Documento de Descripción de Arquitectura

Bryan Antonio Angarita Rodriguez Email: baangaritar@unal.edu.co

Diego Alexander Araque Molina Email: daaraquem@unal.edu.co

Hernando Jose Caviedes Barboza Email: hjcaviedesb@unal.edu.co

Stefania Arguelles Reyes Email: sarguellesr@unal.edu.co

Johan Nicolas Torres Mendoza Email: jntorresm@unal.edu.co

May 2019

1. Introduction

1.1. Nombre del Sistema de Software

UNTube

1.2. Descripción del Sistema de Software

Este es un sistema con arquitectura basada en micro servicios, cuya principal funcionalidad está relacionada con la reproducción y subida de videos. Este sistema también permite comentar y calificar los vídeos de otros usuarios, así también generará una lista de recomendaciones basada en los gustos de cada usuario y las calificaciones de los usuarios a los videos.

1.3. Concerns & Stakeholders

1.3.1. Stakeholders

- Usuarios finales: Son aquellas personas que se registran en la aplicación con el propósito de compartir su contenido (videos) y poder tener una retroalimentación de ellos
- Equipo de desarrollo Es el equipo encargado de hacer la codificación de los requerimientos y la arquitectura para la aplicación deseada
- Arquitecto Es la persona encargada de definir la arquitectura del sistema (aplicación)

1.3.2. Concerns

- Usuario

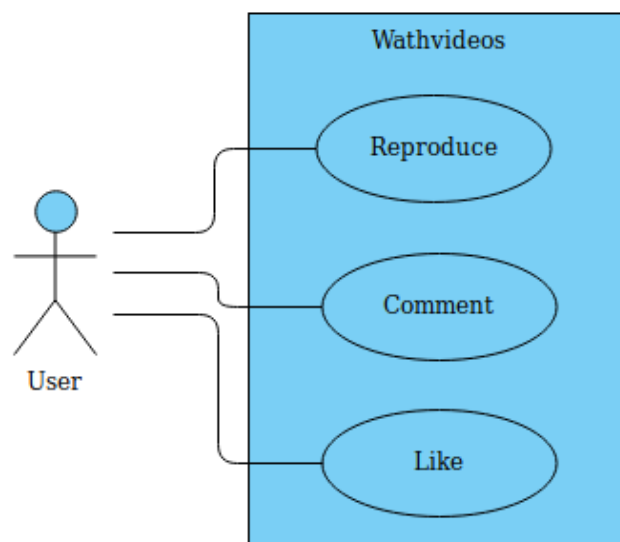
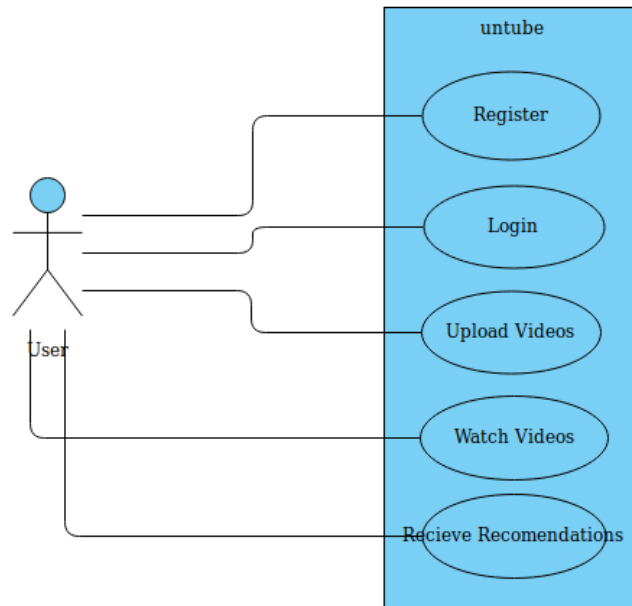
- Usabilidad
- Efectivo
- Fluidez
- Rendimiento
- Acomodación a futuros requerimientos
- Desarrolladores
 - Lenguaje de programación
 - Base de datos
 - Sistema criptográfico
 - Diseño detallado
 - Framework
- Arquitecto
 - Tiempos de entrega
 - Metodología de trabajo
 - Definición de requerimientos
 - Consistencia de la arquitectura
 - Definición de los modelos
 - Completitud del sistema

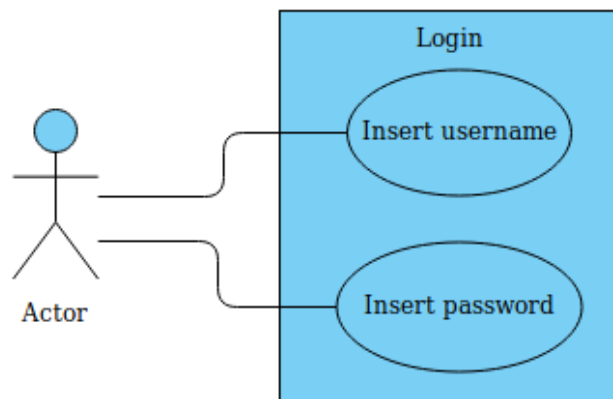
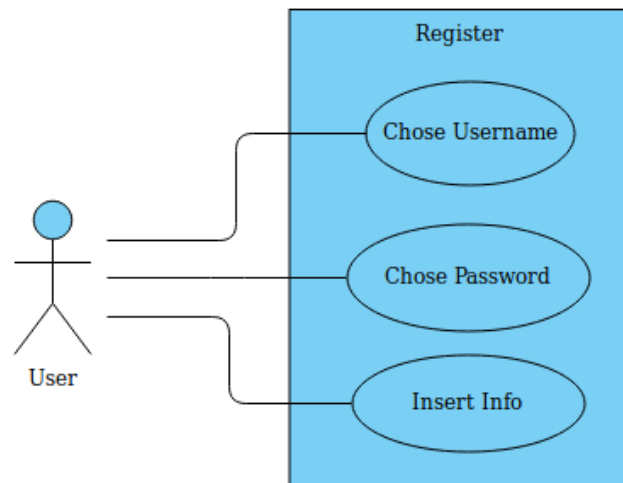
1.4. Requisitos Funcionales

- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.
- El sistema permitirá a un usuario registrado subir vídeos a la plataforma.
- El sistema permitirá a los usuarios registrados ver los vídeos subidos por los demás usuarios.
- El software generará una lista con una serie de vídeos recomendados, esta lista será realizada a partir de los gustos de cada usuario

1.5. Requisitos NO Funcionales

- Lenguajes de programación, Frameworks y Motores de Bases de Datos
 - Ruby -¿Ruby On Rails - MySQL
 - Javascript -¿NodeJS - Mongo DB
 - Go - MongoDB
 - Python -¿Django - Postgres
 - Java -¿Spring Boot - MySQL





2. Estilos y Patrones Arquitectónicos

2.0.1. Estilo

El estilo utilizado para la construcción de la aplicación es en el estilo basado en microservicios.

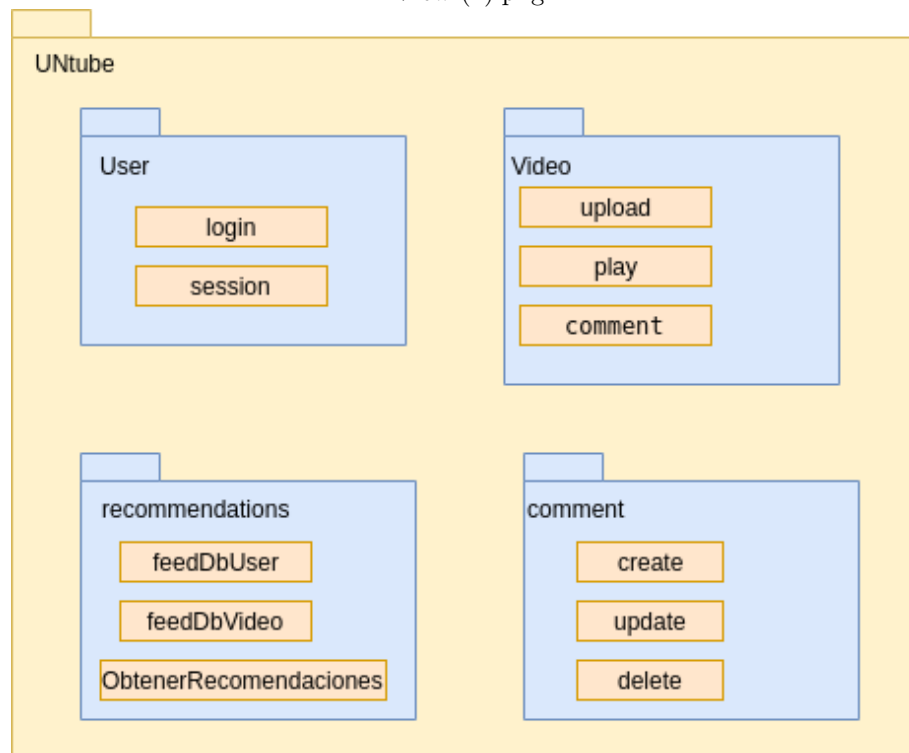
2.0.2. Patrón

Los microservicios creados en el sistema siguen el patrón de microservicios API-REST.

2.1. Vistas Arquitectónicas

2.2. Vista de Descomposición

View (1).png



2.2.1. Elementos

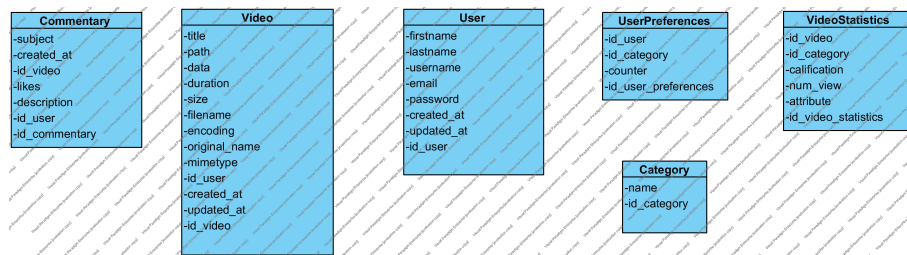
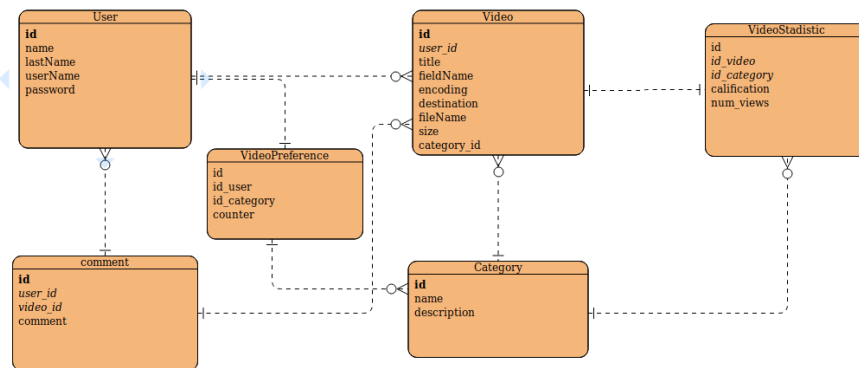
- Módulos
- Submódulos

2.2.2. Relaciones

- is part of

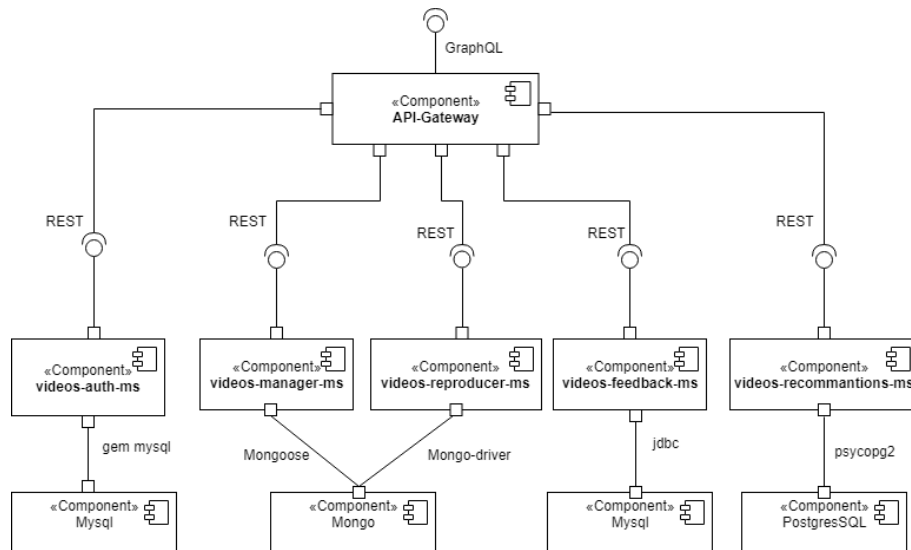
2.2.3. Propiedades

2.3. Vista de Modelo de Datos



2.3.1. Elementos

- user
- video
- category
- commentary
- userPreferences
- videoStatistics



2.4. Vista de Componentes y Conectores

2.4.1. Elementos

- Componentes
- Conectores

2.4.2. Relaciones

- Attachments

2.5. Vista de Capas

2.5.1. Elementos

- Capas
- Segmento de capas

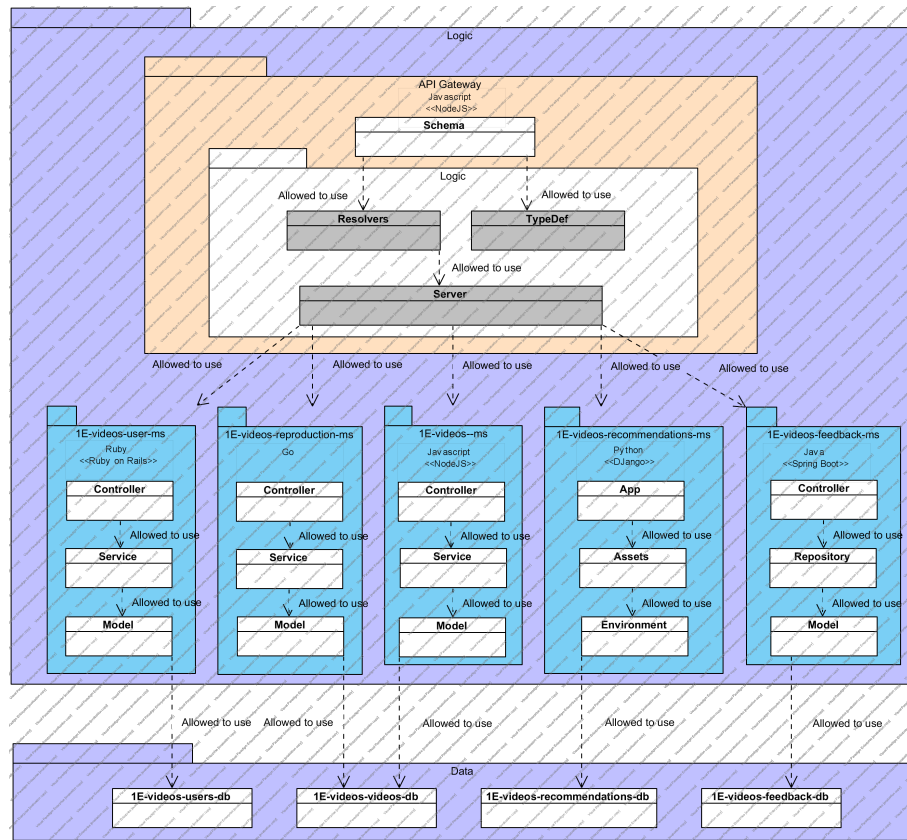
2.5.2. Relaciones

- Allowed to use

2.6. Vista de Despliegue

2.6.1. Elementos

- Nodos
- Ambientes de ejecución



2.6.2. Relaciones

- Deployed in

diagram.png

