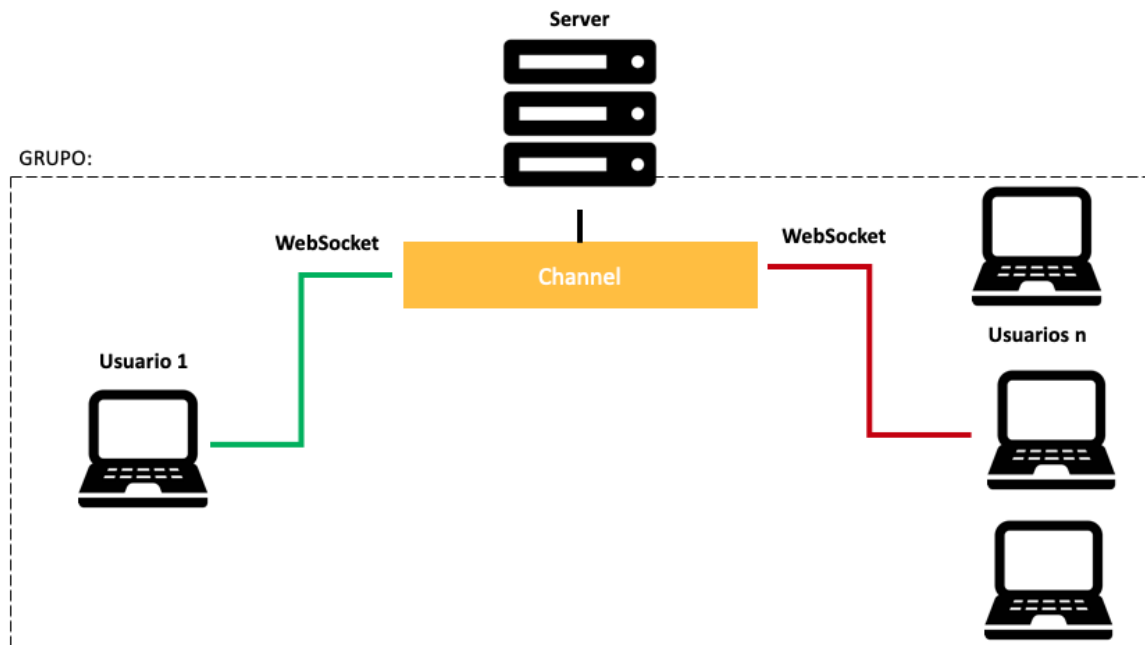


ARQUITECTURA PARA CONSTRUIR CHAT

CON
PYTHON



Arquitectura de software:

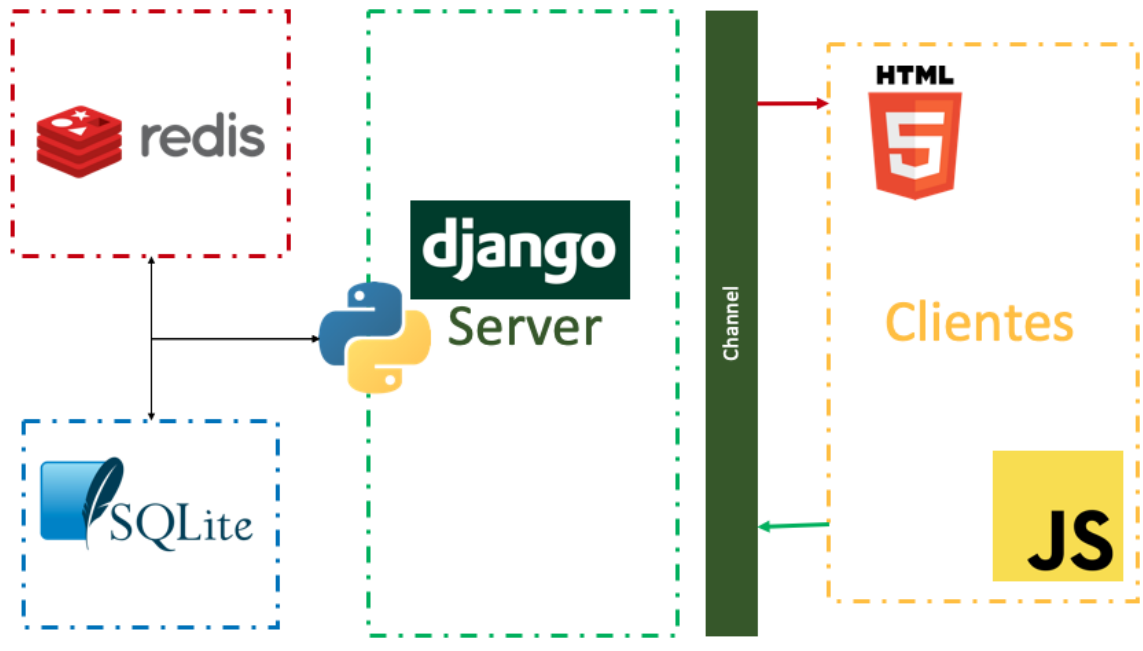


Tecnologías de software necesarias:

Python 3
Django 2.10
Channel
Channel_redis
Websocket JS
HTML5

Redis SQLite

Diagrama de tecnologías sobre la arquitectura:



Esta arquitectura permite conectar a n cantidad de usuarios a un grupo determinado el cual está dividido por un canal (channel a nivel de python) mediante el cual se realiza una verificación del mismo y se redirecciona con otros usuarios al mismo canal de comunicación. WebSockets, una de las tecnologías de comunicación bidireccional más utilizada, se implementa en este proyecto con la finalidad de enviar un Socket al servidor Django para que este le asigne un canal de comunicación nuevo o que tenga congruencia con el que el usuario ingresa en la ruta <http://localhost:8000/chat>.

Instalacion de tecnologías:

Teniendo python, pip y django instalados en nuestro sistema hacemos lo siguiente:

Channel:

Abra un CMD y ejecute el siguiente comando:

```
user$ pip install -U channel
```

esto instalará channel en su versión de python 3 y Django (para este proyecto es la versión 2.1)

Adicional, instale channel_redis para conectarse a redis:

```
user$ pip install channel_redis
```

Redis:

Puede descargarlo desde su sitio oficial: <https://redis.io/download>

O en versiones de Unix/Linux puedes utilizar el gestor de paquetes preinstalado

SQLite:

Ya viene por defecto bajo la instalación de Django con pip en todas sus versiones.

Nota:

Para ejecutar este proyecto es necesario tener preinstaladas las tecnologías antes mencionadas. Solo es ingresar a nuestro BASH y digitar:

```
user$ python manage.py migrate
```

```
user$ python manage.py runserver
```

abre <http://localhost:8000/chat>. En tu navegador de preferencia y que sea compatible con versiones actuales de javascript (ECMA5 o 6).

Referencias:

[PYTHON SOCKETS - CHAT TUTORIAL](#)

[Channels](#)