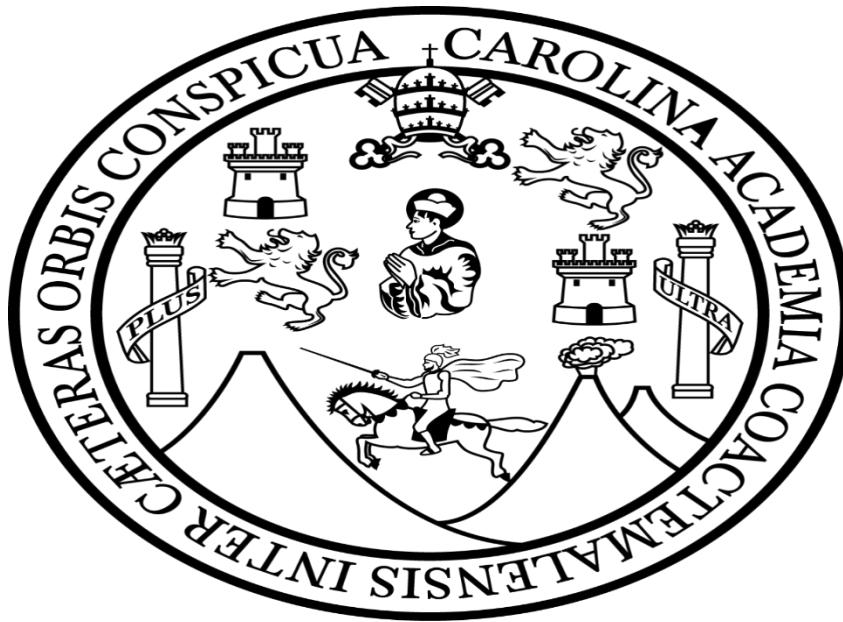


Universidad de San Carlos de Guatemala
Facultad de ingeniería
Escuela de ciencias y sistemas
Análisis y diseño de sistemas 2

Documentación AYDrive Fase2

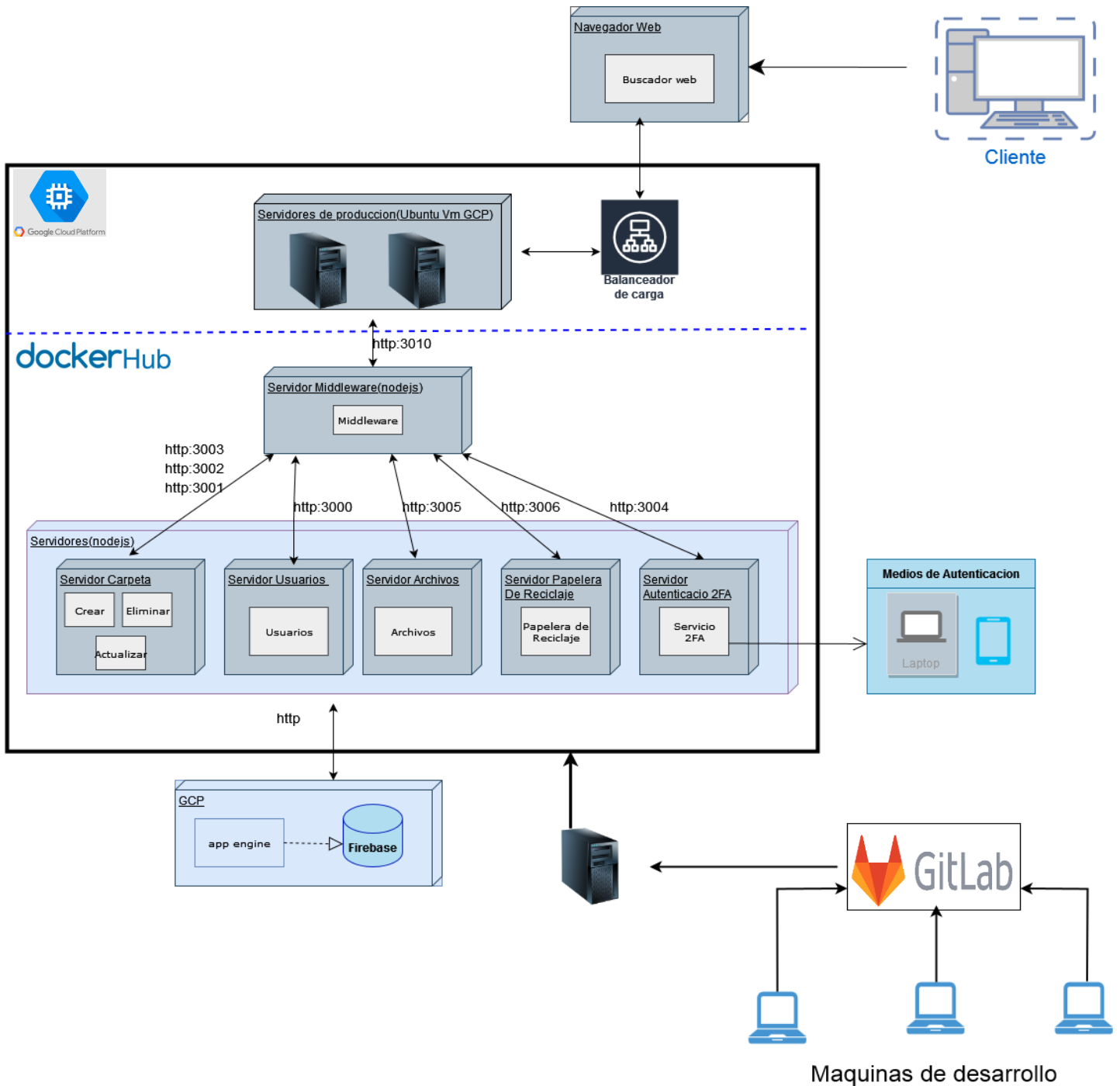


Juan Carlos I Alonzo Colocho
Davis Francisco Edward Enríquez
Selvin Lisandro Aragón Pérez
Edson Armando Guix Manuel
Kelvin Manfredy Vasquez Gomez

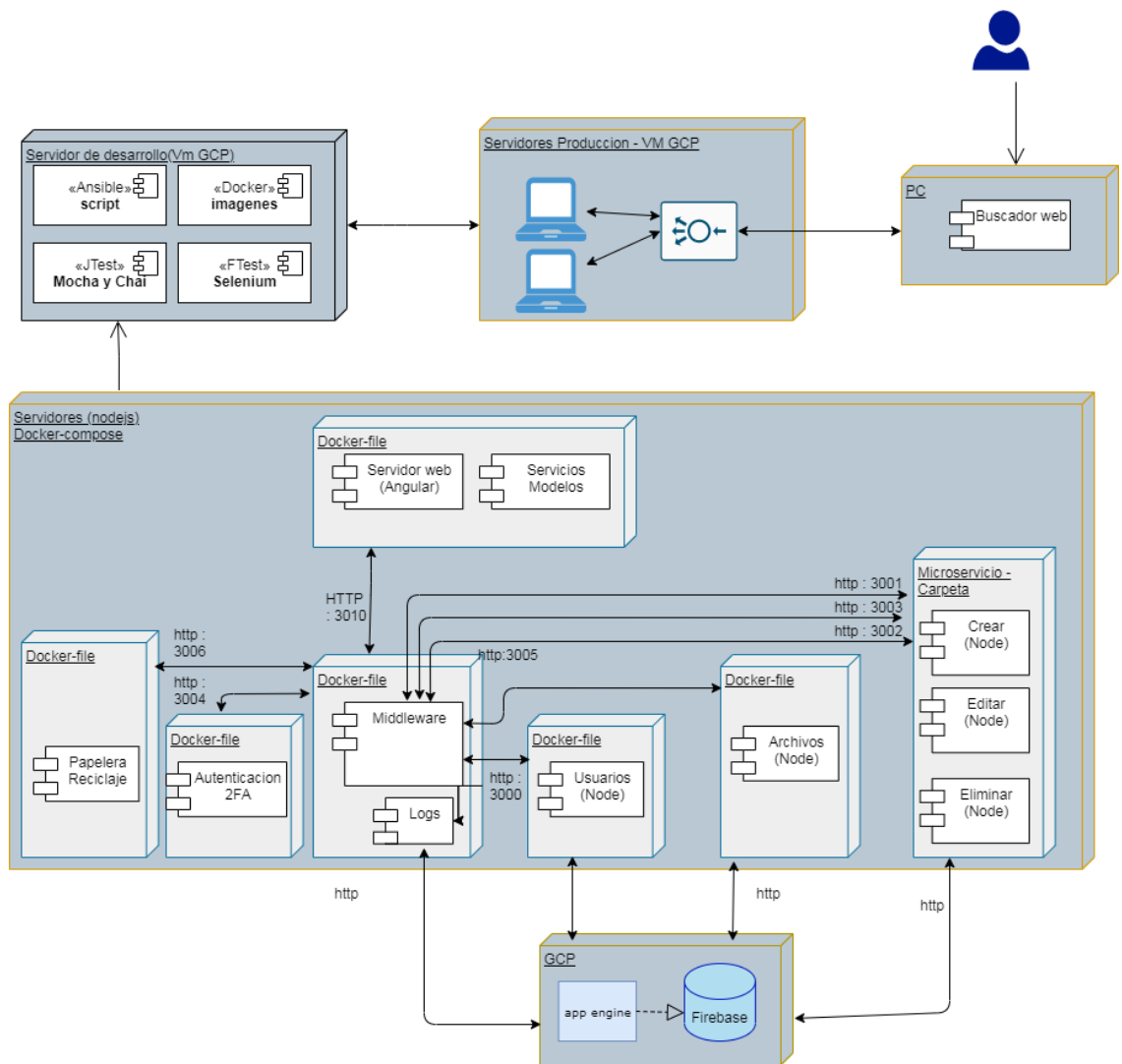
04 de octubre de 2021, Guatemala.

Diagrama de diseño arquitectónico:

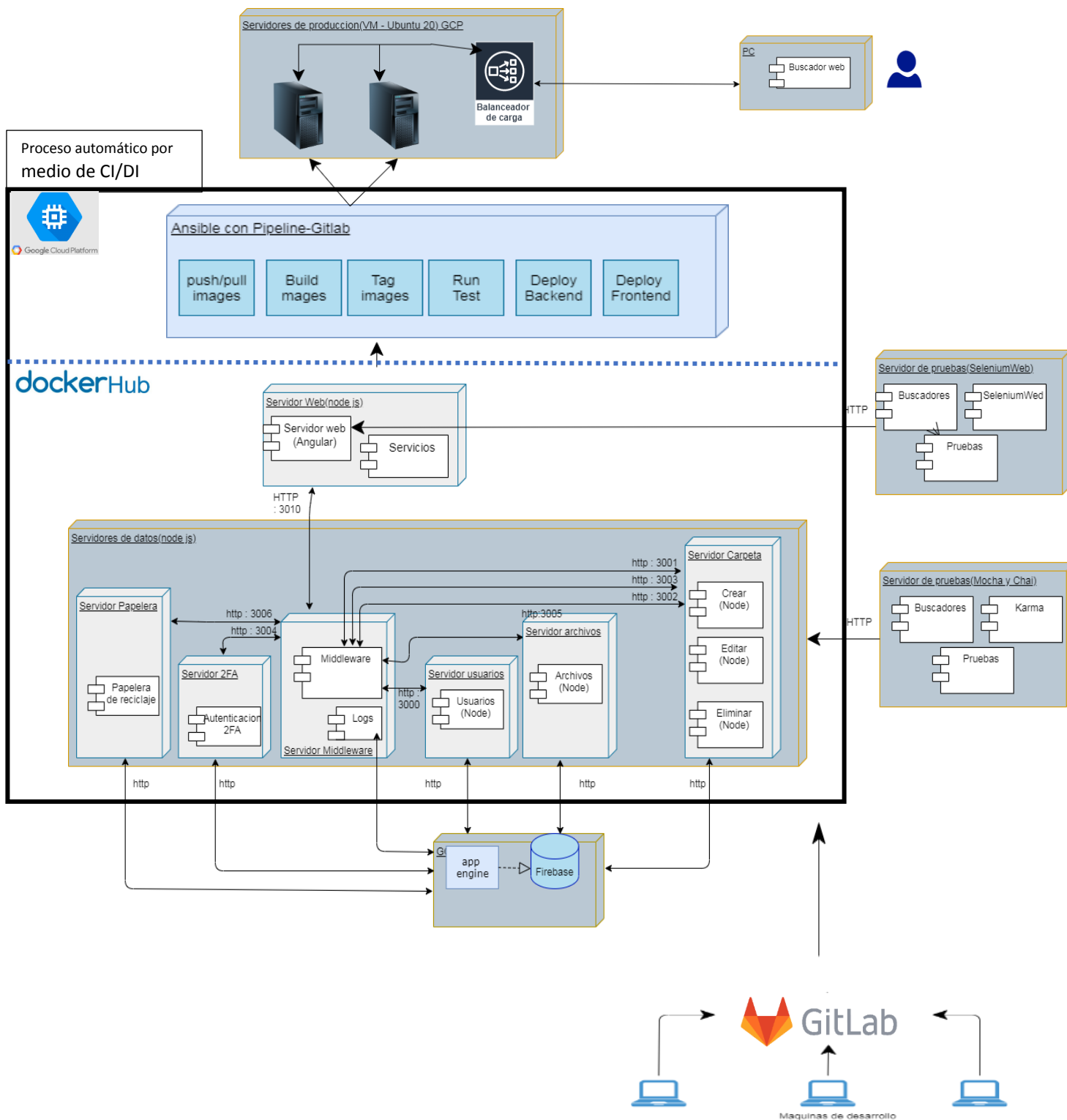
El diseño arquitectónico se interesa como debe organizarse un sistema y como debe diseñarse la estructura global de ese sistema. Este diagrama muestra la forma en que se organiza el sistema como un conjunto de componentes en comunicación.



Diagramas de despliegue:



Diagramas de ambiente de producción:



Diagramas de ambiente de pruebas:

Se utilizo Selenium para el entorno de pruebas del proyecto, ya que permite utilizar un navegador de forma local o remoto para probar los componentes del frontend, permitiendo simular a un usuario interactuando con el sistema de forma automática.

Para las pruebas unitarias se utilizará Mocha y Chai, ya que al estar trabajando toda la infraestructura del proyecto en nodejs, es conveniente utilizar mocha que es un framework de pruebas de java script.

Pruebas funcionales:

A continuación, se listan las pruebas funcionales que se realizaran:

Prueba 1 - Registrar usuario:

Se valida que el usuario pueda registrarse en la página.

Entrada: nickname, nombre, correo, fecha de nacimiento, contraseña y confirmación de la contraseña.

Salida: Se puede dar dos posibles salidas, un mensaje indicando que el usuario se ha creado con éxito o uno con el mensaje de error, el usuario no se registró.

Prueba 2 – Autenticación incorrecta:

Se valida que el usuario pueda iniciar sesión en la página web.

Entrada: nickname y contraseña.

Salida: se despliega un mensaje de error en las credenciales del usuario que quiere registrarse, no se pudo iniciar sesión.

Prueba 3 – Autenticación correcta:

Entrada: nickname y contraseña.

Salida: se despliega un mensaje de inicio de sesión exitoso.

Prueba 4 – Crear carpeta:

Se valida que el usuario pueda crear carpetas en los directorios.

Entrada: nombre de la carpeta.

Salida: se debe mostrar un mensaje, con información de que la carpeta se ha creado con éxito.

Prueba 5 – Eliminar Carpeta:

Se valida que el usuario pueda eliminar una carpeta.

Entrada: en la pagina se debe seleccionar la carpeta que se desea eliminar.

Salida: mensaje indicando que la carpeta ha sido eliminada con éxito.

Pruebas unitarias:

A continuación, se listan las pruebas unitarias que se realizarán:

Prueba 1 – Listar archivos:

Se valida que la función retorne todos los archivos contenidos dentro de un directorio.

Entrada: se indica el identificador de la carpeta del cual se desean ver los archivos.

Salida: se despliega una lista de archivos que contiene la carpeta especificada.

Prueba 2 – Eliminar archivo permanentemente:

Se valida que los archivos se puedan eliminar de forma permanente del sistema.

Entrada: se debe indicar el identificador del archivo que se desea eliminar.

Salida: esta prueba debe retornar un valor verdadero para que la prueba se considere correcta.

Prueba 3 – Crear archivo:

Se valida que la función pueda crear un archivo dentro del directorio indicado.

Entrada: nombre del archivo, identificador del directorio destino.

Salida: debe retornar un valor verdadero o un estado ok para que pase la prueba.

Prueba 4 – Login:

Se valida que la función evalúe los requeridos por el componente.

Entrada: nickname y contraseña.

Salida: debe retornar un valor de verdadero para que la prueba se considere correcta.

Prueba 5 – Crear usuario:

Se valida que la función pueda registrar a un nuevo usuario.

Entrada: usuario, correo, contraseña, fecha nacimiento, nickname.

Salida: se espera un valor verdadero o un estado ok para que la prueba se tome como correcta.

Prueba 6 – Actualiza datos:

Se valida que la función pueda actualizar los datos del perfil de los usuarios.

Entrada: usuario, correo, contraseña, fecha nacimiento.

Salida: se espera un valor verdadero o un estado ok para que la prueba se tome como correcta.

Prueba 7 – Crear carpeta:

Se valida que la función pueda registrar una nueva carpeta.

Entrada: nombre, fecha creación, id propietario.

Salida: se espera un valor verdadero o un estado 200 para que la prueba se tome como correcta.

Prueba 8 – Eliminar carpeta:

Se valida que la función pueda eliminar una nueva carpeta.

Entrada: identificador de la carpeta que se desea eliminar.

Salida: se espera un valor verdadero o un estado 200 para que la prueba se tome como correcta.

Prueba 9 – Editar carpeta:

Se valida que la función pueda editar el nombre de una nueva carpeta.

Entrada: : identificador de la carpeta que se desea editar.

Salida: se espera un valor verdadero o un estado 200 para que la prueba se tome como correcta.

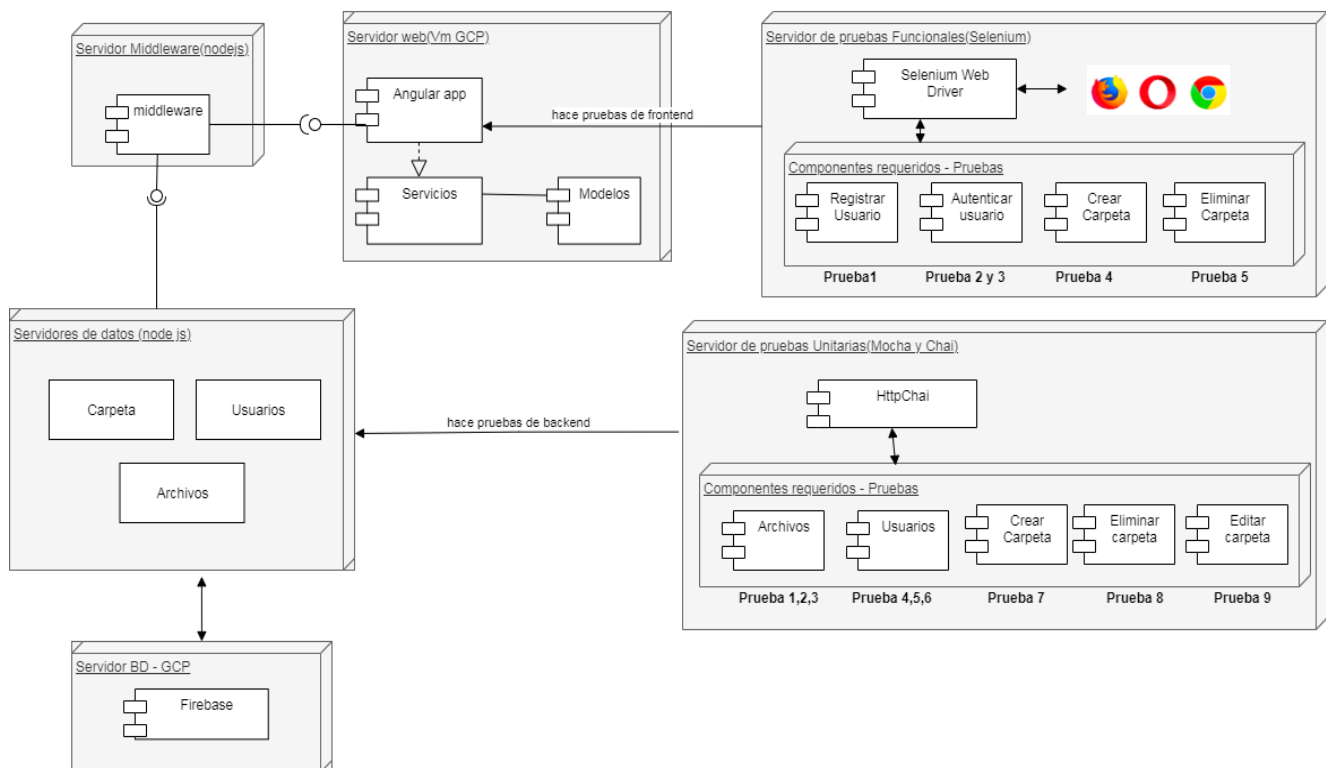


Diagrama Devops:

A continuación, se detallan los pasos realizados por todo el proceso de integración y despliegue continuos, en el diagrama los pasos serán representados con un valor numérico que indica el número de paso para una mayor comprensión.

Nota: todos los pasos que se detallaran se realizan automáticamente por medio de un script, los únicos pasos que se realizan manualmente es la fusión de las ramas.

Integración continua:

1. En la rama feature se crea una nueva funcionalidad, después de completarla se desea agregar el incremento a la rama de desarrollo(develop).
2. Con la ayuda de la herramienta de automatización gitlab-runner, cuando se detecta una petición de fusión(merge) a la rama de desarrollo(develop), se contruyen las imágenes de los contenedores donde se realizarán las pruebas.
3. Se ejecutan las pruebas unitarias y funcionales consecutivamente sobre los contenedores, si todas las pruebas pasan satisfactoriamente el flujo devops sigue su curso, caso contrario, el flujo se detiene.
4. Los contenedores se suben a un repositorio en la nube(docker hub).
5. Después de haber pasado las pruebas satisfactoriamente y los cambios subidos al repositorio de imágenes de docker, se procede a realizar la fusión de la rama de desarrollo a la rama de producción(main).

Despliegue continuo:

6. El despliegue inicia con el merge de la rama de desarrollo a la rama de producción, se descargan las imágenes de los contenedores del repositorio de imágenes.
7. Con la ayuda de la herramienta Ansible, ejecutara un script que aprovisionara la aplicación en dos máquinas de producción, teniendo cada máquina los mismos contenedores.
8. Se ejecutan los contenedores en ambas máquinas de producción.

El diagrama de devops se muestra a continuación, en la siguiente página.

Integración continua:

