

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
ANÁLISIS Y DISEÑO 1  
GRUPO 6

# MANUAL TÉCNICO

## PRÁCTICA 3 FASE 1

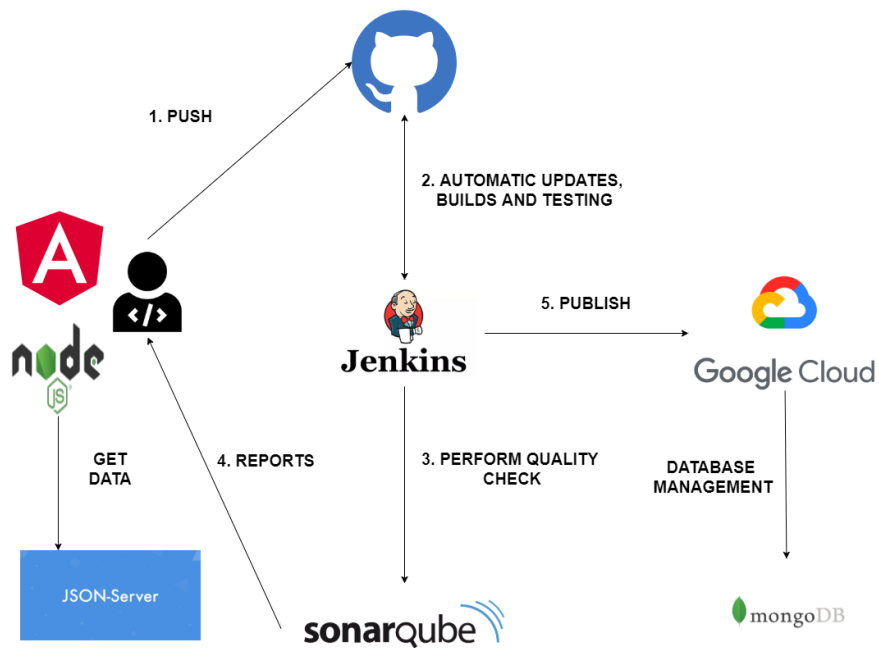
NOMBRE	CARNÉ
EDGAR ARNOLDO ALDANA ARRIOLA	201602797
LUIS LEONEL AGUILAR SÁNCHEZ	201603029
MAYNOR DAVID SALGUERO GUILLÉN	201504192
SERGIO ALEJANDRO SILVA ROLDÁN	201610657
FRANCISCO HUMBERTO LEZANA RAMOS	201503777

# CONTENIDOS

<b>CONTENIDOS</b>	<b>2</b>
<b>ARQUITECTURA</b>	<b>3</b>
VISTA GENERAL DE LA APLICACIÓN	3
DESCRIPCIÓN DE HERRAMIENTAS UTILIZADAS	3
<b>HERRAMIENTA DE PRUEBAS</b>	<b>7</b>
MOCKS	9
<b>HERRAMIENTA INTEGRACIÓN CONTÍNUA</b>	<b>11</b>
<b>BASE DE DATOS</b>	<b>13</b>
DIAGRAMA DE BASE DE DATOS	15
<b>FUNCIONES, MÉTODOS Y CONFIGURACIÓN</b>	<b>16</b>
<b>DOCUMENTACIÓN DE SONAR QUBE</b>	<b>22</b>

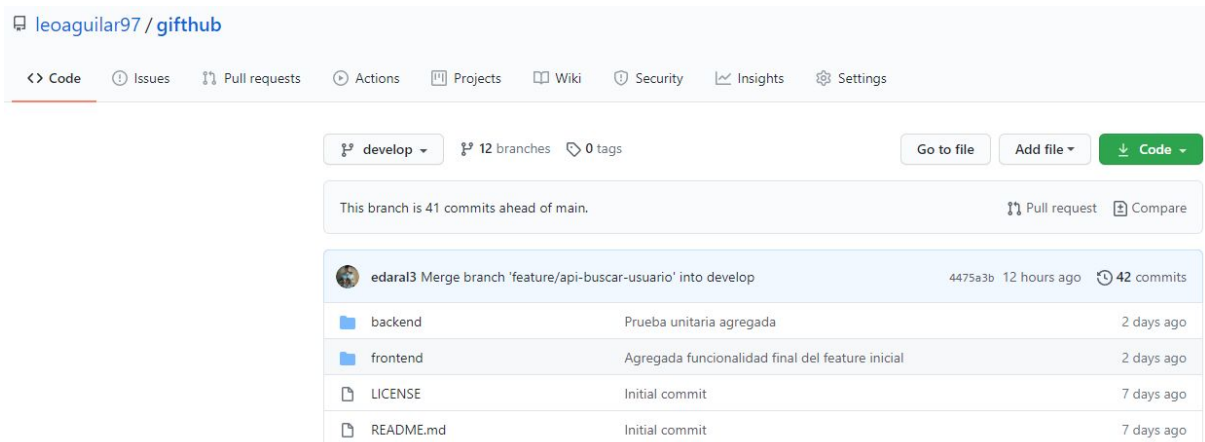
# ARQUITECTURA

## VISTA GENERAL DE LA APLICACIÓN



## DESCRIPCIÓN DE HERRAMIENTAS UTILIZADAS

- **GITHUB:** Sistema de control de versionamiento utilizado para almacenar el código del proyecto.



Link al repositorio: <https://github.com/leoagUILar97/gifthub/tree/develop>

- **ANGULAR:** Framework utilizado para el desarrollo web de la aplicación en el lenguaje TypeScript en el cual se desarrollará toda la parte de Frontend.
- **NODEJS:** Framework utilizado para desarrollar el backend debido a su concurrencia y velocidad.
- **JENKINS:** Software utilizado para la integración continua del proyecto, se utilizó un server en Google Cloud para alojarlo y se crearon dos trabajos.

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		Github Sonar	41 min - #17	5 hr 5 min - #15	16 sec	
		github-app	41 min - #35	3 days 3 hr - #7	20 sec	
		github-app-deploy	1 day 17 hr - #2	N/A	55 sec	
		github-app-frontend	8 hr 41 min - #22	40 min - #23	1 min 42 sec	

Icon: S M L

link al servidor jenkins: <http://34.66.153.213:8080/>

Dashboard
github-app

BUILD NOW
Configure
Delete Project
GitHub Hook Log
Rename

Build History
trend
find

#35

Oct 18, 2020 2:42 AM

#34

Oct 17, 2020 6:41 PM

#33

Oct 17, 2020 3:03 PM

#32

Oct 17, 2020 3:03 PM

#31

Oct 17, 2020 6:26 AM

#30

Oct 17, 2020 4:47 AM

#29

Oct 17, 2020 4:44 AM

#28

Oct 16, 2020 2:55 PM

#27

Oct 16, 2020 12:55 PM

#26

Oct 16, 2020 10:55 AM

#25

Oct 16, 2020 10:33 AM

#24

Oct 16, 2020 10:29 AM

#23

Oct 16, 2020 10:25 AM

#22

Oct 16, 2020 10:17 AM

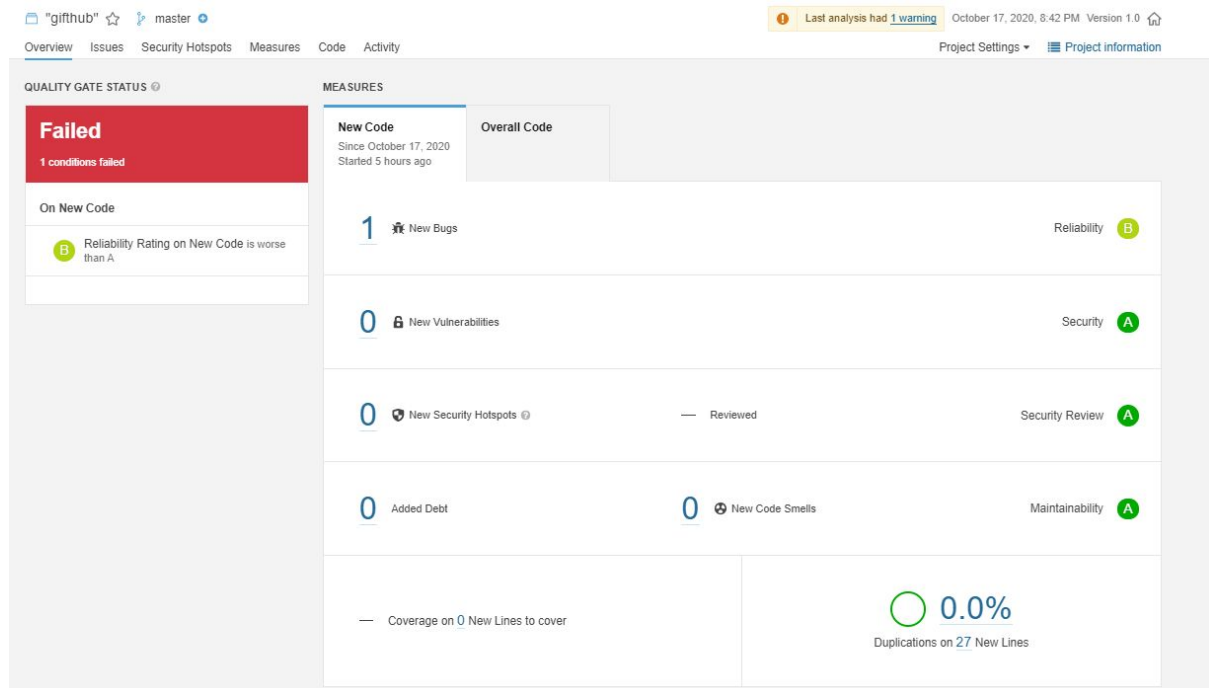
Recent Changes

Downstream Projects
github-app-deploy
github-app-frontend

Permalinks
Last build (#35), 42 min ago
Last stable build (#35), 42 min ago
Last successful build (#35), 42 min ago
Last failed build (#7), 3 days 3 hr ago
Last unsuccessful build (#7), 3 days 3 hr ago
Last completed build (#35), 42 min ago

Vista de builds recientes.

- **SONARQUBE:** Plataforma de código abierto que se usa principalmente para realizar un análisis del código fuente.



Link al server de SonarQube: <http://34.68.114.173:9000/dashboard?id=gifthub>

- **GOOGLE CLOUD:** Proveer el servicio en la nube, de manera que siempre esté accesible a través de internet. En este servicio aloja tanto la API como el frontend, utilizando una instancia de Ubuntu 18.04LTS.

GiftHub-ayd1							
Search products and resources							
VM instances							
<a href="#">CREATE INSTANCE</a> <a href="#">IMPORT VM</a> <a href="#">REFRESH</a> <a href="#">START / RESUME</a> <a href="#">STOP</a> <a href="#">SUSPEND</a>							
Filter VM instances							
<input type="checkbox"/>	Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	jenkins-2	us-central1-a			10.128.0.4 (nic0)	34.66.153.213	SSH
<input type="checkbox"/>	nodejs	us-central1-a			10.128.0.5 (nic0)	35.239.230.8	SSH
<input type="checkbox"/>	sonarqube	us-central1-a			10.128.0.8 (nic0)	34.68.114.173	SSH
<input type="checkbox"/>	sonarqube-db	us-central1-a			10.128.0.7 (nic0)	34.123.250.160	SSH

- **MONGO ATLAS:** Utilizado para almacenar los datos de usuarios, tarjetas y todo lo necesario para la construcción del proyecto.

- **Base de datos de pruebas:**

The screenshot shows the MongoDB Atlas interface for a project named 'PROJECTUSAC'. The database 'GifthubData' is selected, and the 'Collections' tab is active. The interface shows two databases: 'testing' and 'usuarios'. The 'testing' database has a size of 17.11KB and an index size of 96KB. The 'usuarios' collection has 100 documents, a size of 17.11KB, and an index size of 24KB. A table lists the collections and their details:

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
usuarios	100	17.11KB	176B	4	96KB	24KB

- **Base de datos de producción:**

The screenshot shows the MongoDB Atlas interface for the 'gifthub.usuarios' collection. The collection has a size of 17.36KB, 101 total documents, and an index size of 128KB. The interface shows the 'Find' tab selected, and a filter bar is visible. The query results show a single document with the following fields:

```
{
  "_id": ObjectId("5f8bab3fe3655be534cdedd"),
  "username": "EAWLL",
  "correo": "pede@metusVivamuseuismod.co.uk",
  "contrasena": "AOE18HZK7YO",
  "nombres": "Oren",
  "apellidos": "McCray",
  "dpi": 896344147851,
  "edad": 58
}
```

# HERRAMIENTA DE PRUEBAS

Se realizaron pruebas del primer módulo de usuarios, se utilizaron los siguientes frameworks:

- **Mocha:** gestor de pruebas. <https://www.npmjs.com/package/mocha>
- **Chai:** para realizar asserts y expects. <https://www.npmjs.com/package/chai>
- **Sinon:** para utilizar spies, stubs y mocks. <https://www.npmjs.com/package/sinon>
- **Jasmine y Karma:** para realizar pruebas en frontend. <https://jasmine.github.io/>
- **Selenium y Protractor:** para realizar pruebas e2e. <https://www.protractortest.org/#/>

Las pruebas finalizadas son las siguientes:

```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS

Validaciones en la BD
  ✓ Mostrar error cuando no se pueda conectar a la base de datos

Historia: Registrar Usuarios
  POST /
    ✓ Guardar un usuario con datos correctos
    ✓ Intentar guardar un usuario con datos incorrectos
    ✓ Error de la base de datos al intentar insertar usuarios.
  PUT /
    ✓ Actualiar un usuario existente
    ✓ Actualiar un usuario que no existente
    ✓ Intentar actualizar usuario sin enviarle datos
    ✓ Error de la base de datos al intentar actualizar usuarios.
  GET /
    ✓ Buscar un usuario existente
    ✓ Buscar un usuario que no existe
    ✓ Error de la base de datos al buscar usuario.
    ✓ Obtener usuarios
    ✓ Error de la base de datos al intentar obtener todos los usuarios.
  DELETE /
    ✓ Eliminar un usuario existente
    ✓ Intentar eliminar un usuario que no existente
    ✓ Error de la base de datos al intentar eliminar un usuarios.

Historia: Conectarse a api externa
  GET /
    ✓ Actualizar las giftcards en la base de datos
    ✓ Error al realizar la peticion a la api externa

Historia: Ver catalogo giftcards
  GET /
    ✓ Obtiene las giftcars de la base de datos
    ✓ Error de la base de datos al obtener las giftcards
    ✓ Obtiene los precios de la base de datos
    ✓ Error de la base de datos al obtener los precios
    ✓ Obtiene los precios de la base de datos
    ✓ Error de la base de datos al obtener los precios
```

## Historia: Realizar compra

POST /

- ✓ Error, no se pararon los datos requeridos para realizar el pago
- ✓ Error, las giftcards no se encontraron en la base de datos
- ✓ Las giftcards se encontraron en la base de datos
- ✓ Buscar un usuario en la base de datos
- ✓ Verifica si existen tarjetas no validas
- ✓ Guarda las transacciones fallidas si los datos de la tarjeta de credito no coinciden
- ✓ Guarda las transacciones fallidas si la tarjeta no cuenta con los fondos necesarios
- ✓ Se insertan giftcards al usuario y se añade a las transacciones del usuario
- ✓ Actualizar el usuario con las nuevas giftcards compradas
- ✓ Erroa al Actualizar el usuario con las nuevas giftcards compradas
- ✓ Generacion codigo alfanumerico
- ✓ Generacion numero enmascarado
- ✓ Error al generar numero enmascarado
- ✓ Compra de tarjetas exitosa (1285ms)

## Login

POST /

- ✓ Loguin exitoso
- ✓ Login con un usuario que no existe
- ✓ Contraseña incorrecta
- ✓ Error de la base de datos al buscar hacer login.

## Regalar GiftCards

- ✓ Buscar un usuario en la base de datos
- ✓ Retornar indefinido si el usuario no es encontrado
- ✓ Si no existe la tarjeta en el inventario del usuario que regala, devuelve un error
- ✓ Actualizar usuario devuelve datos del usuario de manera correcta
- ✓ Actualizar usuario devuelve un mensaje de error cuando hay un error modificandolo
- ✓ Regalar giftcard devuelve error si hay datos faltantes
- ✓ Si no se encuentra el usuario que regala devuelve un error 404
- ✓ Si no se encuentra el usuario al que regalan devuelve un error 404
- ✓ Si hay un error modificando las tarjetas del usuario al que regalan devuelve un error 400
- ✓ Si hay un error modificando al usuario que regala muestra un error
- ✓ Devuelve un codigo 200 si las tarjetas fueron regaladas con exito
- ✓ Devuelve error si al requerir las transacciones no se es un usuario admin
- ✓ Devuelve una lista de transacciones si el usuario es admin
- ✓ Devuelve un error 500 si existe un error en la base de datos

56 passing (2s)

## Karma v4.4.1 - connected

DEBUG

jasmine 3.5.0

Options

20 specs, 0 failures, randomized with seed 25183

Finished in 1.72s

PagarComponent

CarritoComponent

HomeComponent

ModificarComponent

RegalarGiftCardsComponent

- Debe retornar la longitud del arreglo de giftcards a regalar
- Debe retornar verdadero al agregar 1 giftcard
- Debe retornar el nombre display de una tarjeta
- Debe retornar la longitud del arreglo de giftcards
- Debe retornar la disponibilidad de una tarjeta
- should create
- Debe retornar impreso al llamar al metodo imprimirError
- Debe retornar falso al momento de regalar una tarjeta vacia

DetallesCompraComponent

ServicioComprarTarjetaService

InventarioGiftCardsComponent

- Debe retornar verdadero al agregar 1 giftcard
- Debe retornar la longitud del arreglo de giftcards
- Debe retornar la disponibilidad de una tarjeta
- Debe retornar impreso al llamar al metodo imprimirError
- should create

VistaGiftCardsComponent

- should create

NavbarComponent

RegistrarseComponent

ServicioHistorialService

LoginComponent

AdminComponent

SidebarComponent

ModificarUsuarioService

ServicioRegalarService

- should be created
- Debe regalar una tarjeta
- Debe retornar un usuario encontrado

AppComponent

ServicioVistaGiftCardsService

- Debe actualizar la bd con las giftcards
- should be created
- Debe obtener las giftcards desde nuestra base

HistorialComprasComponent



## MOCKS

Se utilizarán mocks en una variedad de pruebas debido a su alto nivel de adaptación para cada requerimiento que se debe probar. Sin embargo, para este sprint solo se realizaron 5 pruebas con mocks, de las cuales las dos formas más significativas de ellas se detallan a continuación:

La siguiente prueba corresponde a la verificación de los datos enviados como código al momento de eliminar un usuario. Si se intenta eliminar un usuario que no existe, se espera a que la API retorne un 400 bad request.

```
it('Devuelve error si se envia un codigo incorrecto', done => {

    let res = {
        send: () => {},
        status: sinon.stub().returnsThis()
    };

    const mock = sinon.mock(res);

    mock.expects("send").once().withExactArgs({
        message: "Codigo invalido"
    });

    controller.delete({ params: { codigo: "codError" } }, res);

    expect(res.status.calledOnce).to.be.true;

    expect(res.status.firstCall.calledWithExactly(400)).to.be.true;

    mock.verify();

    done();
});
```

Cuando se encuentra un error 500 en la base de datos, debido a falta de recursos para consumirla (internet, espacio, etc)... o debido a una gama diversa de razones, se espera que la api siga funcionando y regrese un error 500 junto con un mensaje exacto.

```
it("Error 500 cuando se encuentra un error en la DB - GETUSUARIOS",
done => {

    let catchStub = sandbox.stub();
    let stub = sandbox.stub(controller.Usuario,
'find').returns({
```

```

        then: sandbox.stub().callsFake(() => { return {
catch: catchStub } })),
    });

    catchStub.callsFake((cb) => {
        cb({ message: "STUB - Error" });
    });

    let res = {
        json: () => {},
        status: sinon.stub().returnsThis()
    };

    let req = {
        search: "asd"
    };

    const mock = sinon.mock(res);

    mock.expects("json").once().withExactArgs({
        message: "Error al retornar una lista de productos
de la BD"
    });

    controller.find({ query: req }, res);

    expect(stub.calledOnce).to.be.true;
    expect(res.status.calledOnce).to.be.true;

    expect(res.status.firstCall.calledWithExactly(500)).to.be.true;

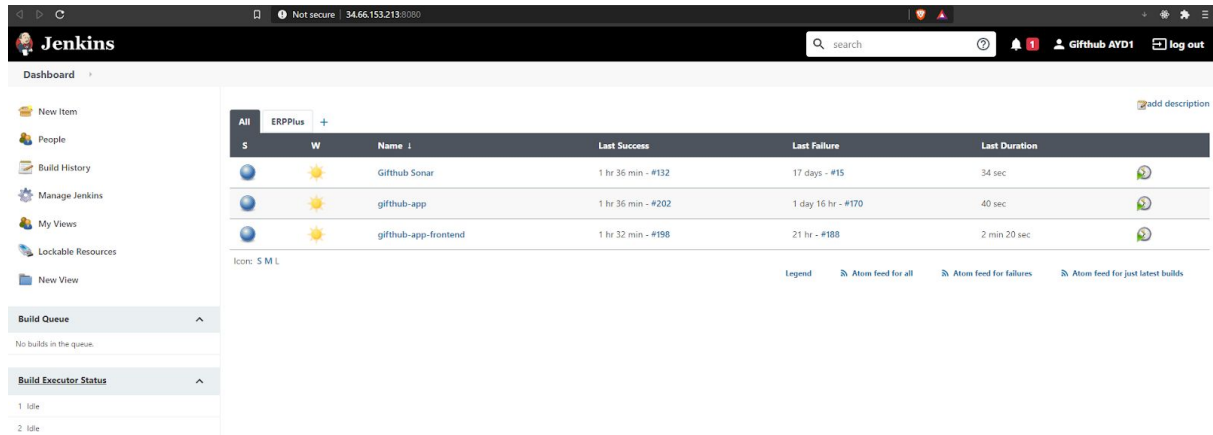
    mock.verify();

    done();
    });
});

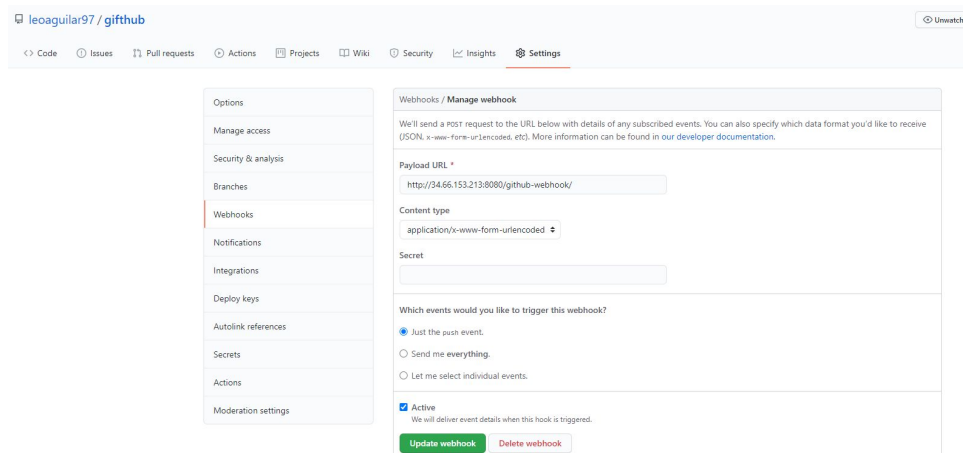
```

# HERRAMIENTA INTEGRACIÓN CONTÍNUA

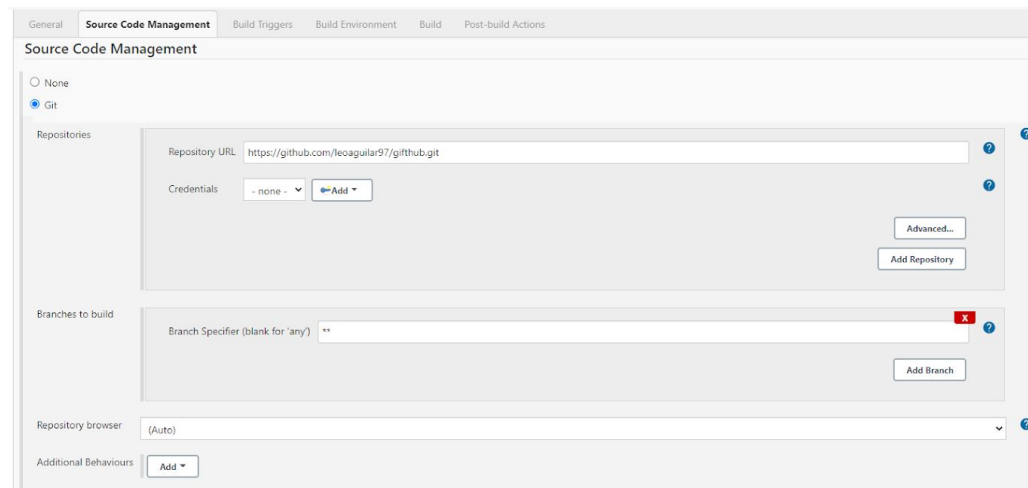
Se utilizó [Jenkins](#) cómo herramienta de integración continua, Jenkins es un servidor de automatización open-source. Además, se conectó el repositorio en Github del código que realizamos, a través de un WebHook de Github.



Vista de los proyectos de Jenkins



Webhook de Github



Source Code Management - Vista del Build Script en Jenkins

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

---

**Execute shell** X ?

Command

```
cd backend
npm install
npm test
```

[See the list of available environment variables](#)

Advanced...

---

**Execute shell** X ?

Command

```
gcloud beta compute ssh --zone "us-central1-a" "nodejs" --project "github-ayd1" <<EOF
cd ~/github
git reset --hard
git pull

cd backend
npm install - production
pm2 restart all

exit
EOF
```

[See the list of available environment variables](#)

Advanced...

Add build step ▾

## Execute Shell - Vista del Build Script en Jenkins

General Source Code Management Build Triggers Build Environment Build **Post-build Actions**

---

Notes to push to remote repositories

---

**E-mail Notification** X ?

Recipients

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

☐ Send e-mail for every unstable build

☐ Send separate e-mails to individuals who broke the build ?

## Post-build Action - Vista del Build Script en Jenkins

General Source Code Management Build Triggers Build Environment Build **Post-build Actions**

---

**Editable Email Notification** X ?

Disable Extended Email Publisher ☐ ?

Allows the user to disable the publisher, while maintaining the settings

Project From

Project Recipient List

Comma-separated list of email address that should receive notifications for this project.

Project Reply-To List

Comma-separated list of email address that should be in the Reply-To header for this project.

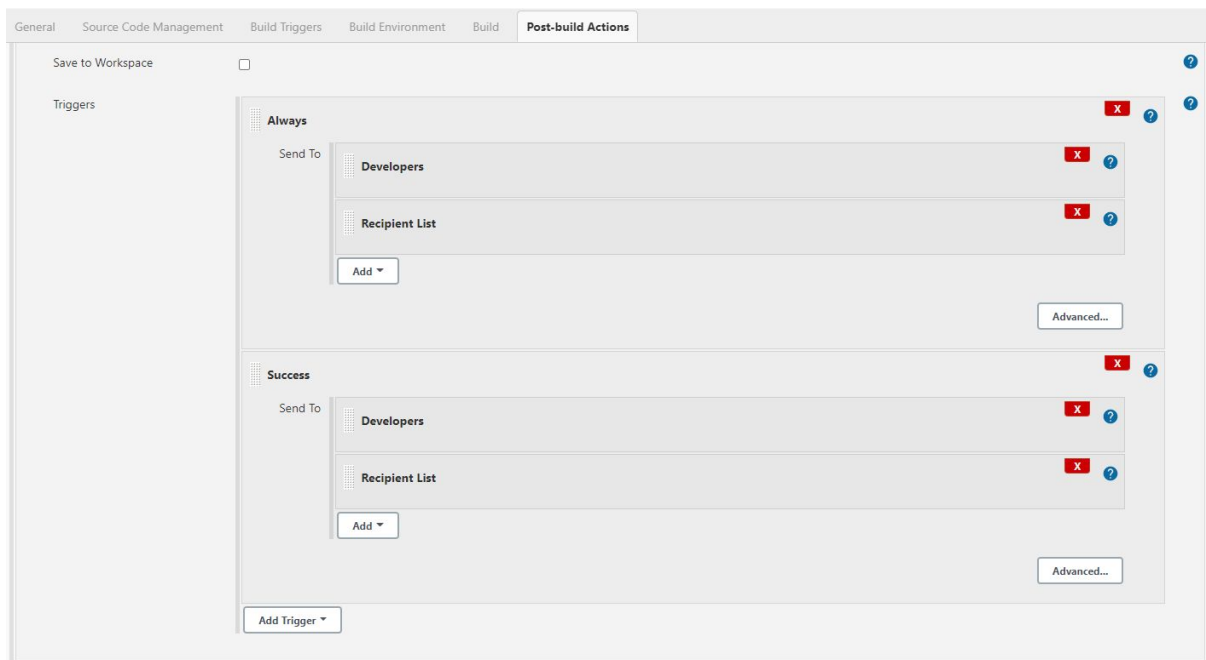
Content Type

Default Subject

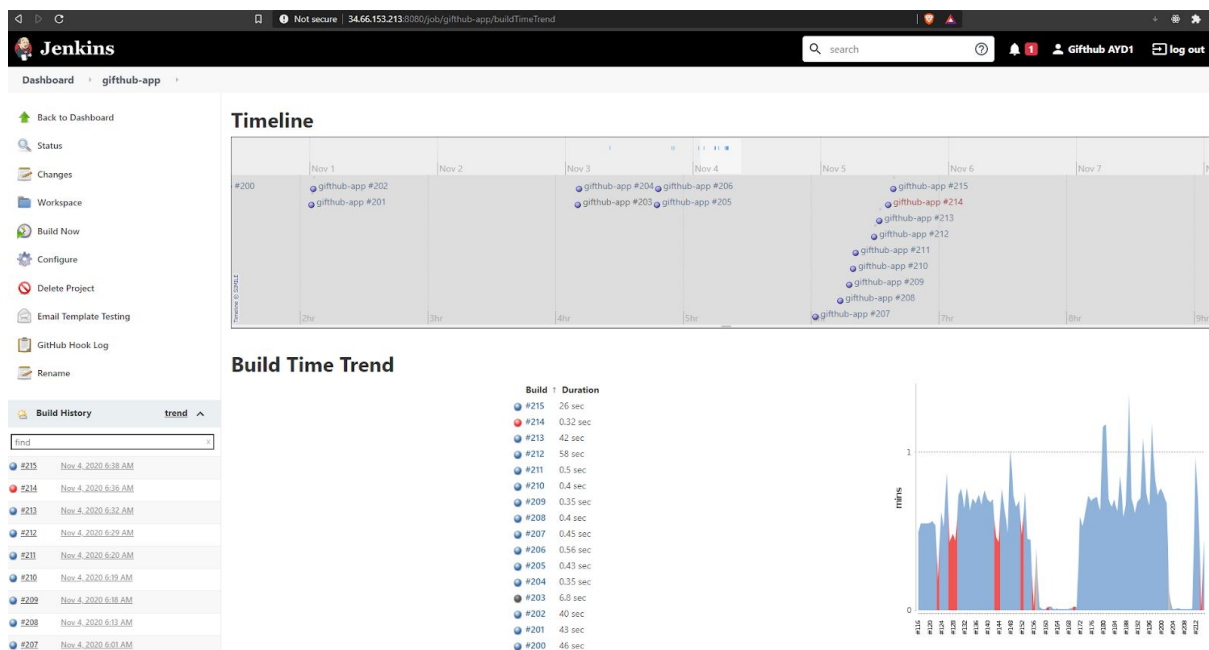
Default Content

Attachments

## Post-build Action - Correo - Vista del Build Script en Jenkins



Post-build Action - Trigger Correo - Vista del Build Script en Jenkins



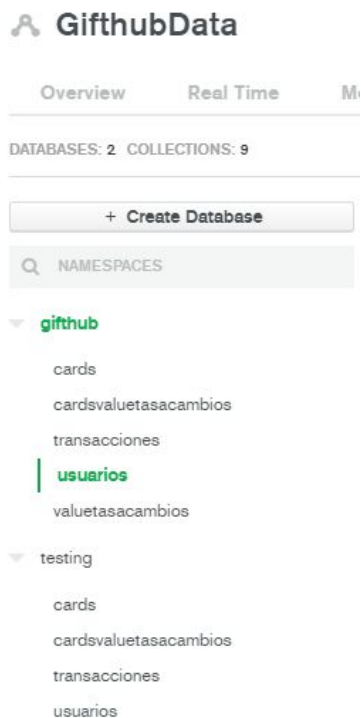
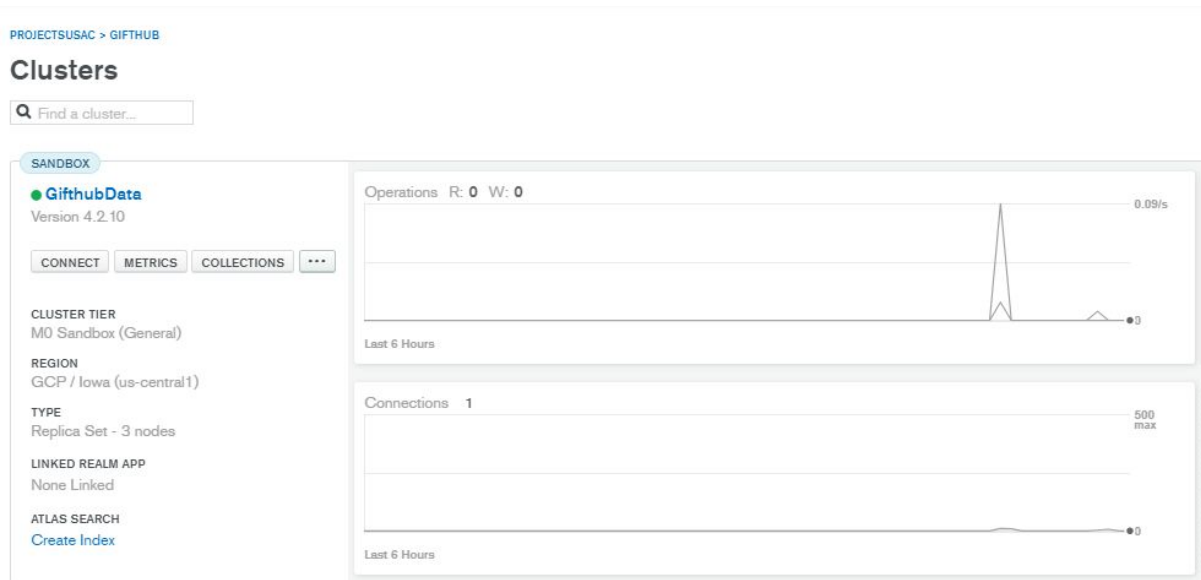
Vista del proyecto de jenkins

## BASE DE DATOS

Se utiliza una base de datos no relacional con el dbms MongoDB, utilizado en un clúster M0 en la capa gratis del proveedor de servicios en la nube Mongo Atlas (<https://www.mongodb.com/cloud/atlas>).

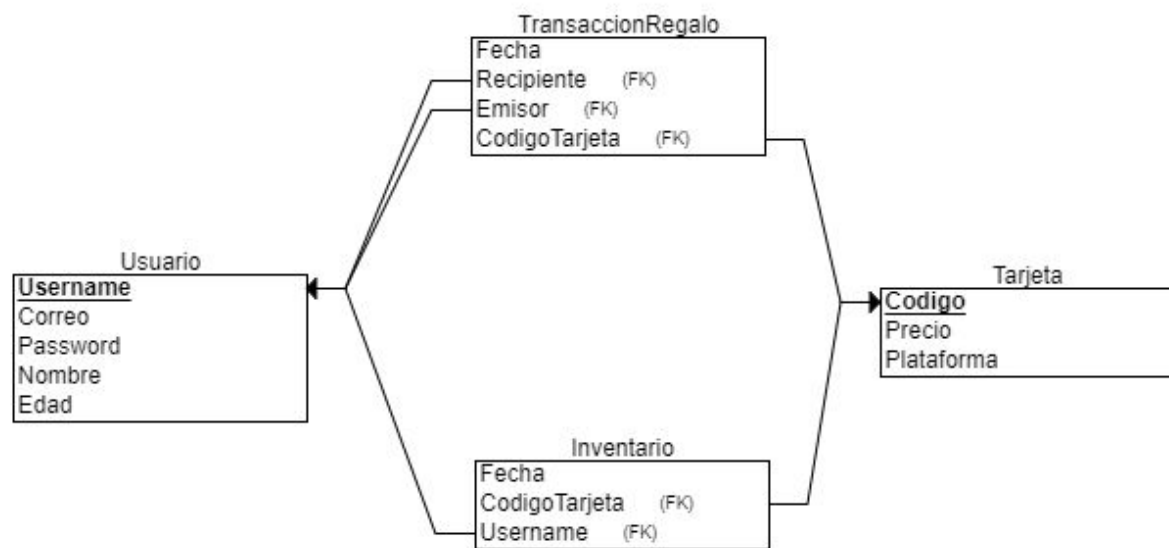
Para ello se utilizó la cuenta personal de uno de los integrantes del equipo, se configuró una nueva organización con un nuevo proyecto dentro.

En el proyecto se definieron dos bases de datos, una de producción y otra de pruebas unitarias y funcionales, las cuales cuentan con sus propias colecciones de datos independientes.



Bases de datos guardadas en MongoAtlas.

## DIAGRAMA DE BASE DE DATOS



# FUNCIONES, MÉTODOS Y CONFIGURACIÓN

## **controllers/usuario.js**

**exports.crear = (req, res):** Recibe un json con los datos del usuario, lo agrega a la base de datos y regresa archivo json con mensaje de confirmación o de error en caso de no poder agregar el usuario.

**exports.actualizar = (req, res):** Recibe un json con los datos del usuario, lo agrega a la base de datos y regresa archivo json con mensaje de confirmación o de error en caso de no poder actualizar el usuario.

**exports.buscar = (req, res):** Recibe un query con username para realizar la búsqueda del respectivo usuario y devolver sus datos en forma de json, de no encontrarlo se devuelve un mensaje de usuario no encontrado.

**exports.eliminar = (req, res):** Recibe un query con username para realizar la eliminación del respectivo usuario, de no encontrar al usuario se devuelve un mensaje de error.

**exports.getAll = (req, res):** Retorna a todos los usuarios existentes en la base de datos.

## **controllers/regalar.js**

**exports.regalar = async(req, res):** Recibe un json con dos usernames y una lista de giftcards, las cuales se restan del username1 y se suman al username2 y luego se guarda en la lista de transacciones. En caso de generarse al momento de realizar el regalo se guarda el error en una lista de transacciones.

**exports.transacciones = async function (req, res):** Esta función retorna la lista de todas las transacciones realizadas por todos los usuarios.

**async function guardarEnHistorial(values, acceptance, cb):**  
Recibe la información que es necesaria para guardar en el historial de transacciones.



**async function obtenerUsuario(username):** Devuelve el usuario que cuente con el username que recibe como parámetro.

**async function modificarInventarioTrarjetasUsuario1(usuario, tarjetas):** Quita las tarjetas que el usuario1 regalara.

**async function modificarInventarioTrarjetasUsuario2(usuario, tarjetas):** Agrega las tarjetas que el usuario1 regalo.

**async function actualizarUsuario(usuario):** actualiza el usuario en la base de datos.

### **controllers/login.js**

**exports.findOne = (req, res):** Recibe un json con un username o correo electrónico y contraseña, el username o correo electrónico es utilizado para buscar al usuario en la base de datos, mientras que la contraseña es utilizada para validar si se retorna o no la información del usuario.

### **controllers/compra**

**async function obtenerGiftcards(tarjetas):** Devuelve la información de las tarjetas que encuentran en la base de datos que se desea comprar.

**async function buscarUsuario(username):** Devuelve al usuario que coincida con el username que se pasa como parámetro.

**async function realizarTransaccion(tarjetasCredito, userData, tarjetaUsuario, monto, TarjetasGift, username, tarjetaCompra):**  
Se pasa la información de un usuario, las giftcards, la tarjeta de crédito y demás información utilizada para corroborar que la tarjeta de crédito existe, si esta existe se reduce el crédito de la tarjeta de crédito. En caso de que la tarjeta no exista o no cuente con el crédito suficiente para realizar la compra se agrega el error al historial de compras del usuario.

**async function realizarTransaccion2(tarjetas, tarjetasUsuario, card, usuario, giftcard, tarjetasGitf, monto):** Verifica que las tarjetas que se deseen comprar existan en la base de datos

**async function actualizarUsuarios(usuario, username):** Se pasa el usuario con la información nueva para actualizar al usuario que coincida con el username en la base de datos.

**async function generarAlfanumerico():** Devuelve un código alfanumérico de 8 caracteres utilizado en las giftcards.

**async function encriptar():** Recibe un número de 16 dígitos y devuelve una cadena de 16 caracteres con el siguiente formato "####XXXXXXX####".

**async function guardarEnHistorial(tarjetas, username, mensaje, estado):** se pasa la información requerida en la lista de transacciones y se guarda en la base de datos.

**exports.pago = async(req, res):** recibe un json que contiene un username, una tarjeta de crédito y una lista de tarjetas, se verifica que exista el usuario, la tarjeta de crédito (si no existe se agrega) y las tarjetas enviadas. Si la información es correcta se agrega al historial de transacciones del usuario y a la lista de transacciones realizadas por todos los usuarios, se actualiza el usuario y se guarda toda la información en la base de datos.

## **controllers/cards**

**exports.getAll = (req, res):** Devuelve la lista de giftcards que se encuentra en la base de datos.

**exports.actualizar = (req, res):** Obtiene una lista de giftcards de la api proporcionada por el auxiliar, elimina las giftcards que se encuentran actualmente en la base de datos y guarda los datos obtenidos recientemente.

**function obtenerTasaCambios(res):** Devuelve la tasa de cambios que se encuentra en la base de datos.

**function ObtenerPrecios(res):** Devuelve los precios de las giftcards de la base de datos.

### **models/usuario.js**

**module.exports = mongoose => {}:** Devuelve el modelo utilizado para el usuario.

### **models/transacciones.js**

**module.exports = mongoose => {}:** Devuelve el modelo utilizado para las transacciones.

### **models/cardValueTasaCambio.js**

**module.exports = mongoose => {}:** Devuelve el modelo utilizado para guardar los datos obtenidos de la api proporcionada por el auxiliar.

### **models/index.js**

**module.exports = db:** Realiza la configuración con la base de datos, importa los modelos a utilizar y los convierte en modelos de mongoose.

### **routes/usuario.js**

**app.use("/usuario", router):** Crea la ruta "/usuario" para realizar las operaciones de CRUD sobre usuario.

**router.post("/", usuario.crear):** Crea el post que realiza la creación de los usuarios, recibe un json con los datos del usuario.

**router.put("/:username", usuario.actualizar):** Crea el put que realizará la actualización de usuarios por medio del username, recibe un json de la información a modificar.

**router.get("/:username", usuario.buscar):** Crea un get para obtener a los usuarios por su username.

**router.get("/", usuario.buscar):** Crea un get para obtener a todos los usuarios.

**router.delete("/:username", usuario.eliminar):** Crea un delete que elimina a los usuarios por su username.

### **routes/regalar.js**

**router.post("/", regalar.regalar):** Crea la ruta para regalar, tarjetas de un usuario a otro.

**router.get("/", regalar.transacciones):** Crea la ruta que retorna la lista de transacciones que el usuario administrador puede observar.

### **routes/login.js**

**router.post("/", producto.findOne):** Crea la ruta para realizar un login en la aplicación.

### **routes/compra.js**

**router.post("/pago", compra.pago):** Crea la ruta para realizar compras de tarjetas por usuarios.

### **routes/cards.js**

**router.post("/", producto.actualizar):** Crea la ruta para actualizar los datos de la api compartida por el auxiliar en la base de datos.

**router.post("/mongo", producto.getAll):** Crea la ruta para obtener los datos de las giftcards de la base de datos.

### **config/db.config.js**

Contiene un module.exports el cual contiene dos urls, una para la base de datos que se implementara en el proyecto y la otra para realizar tests.

## **index.js**

**const dbConnect = (url, cb):** Se encarga de realizar la conexión a la base de datos.

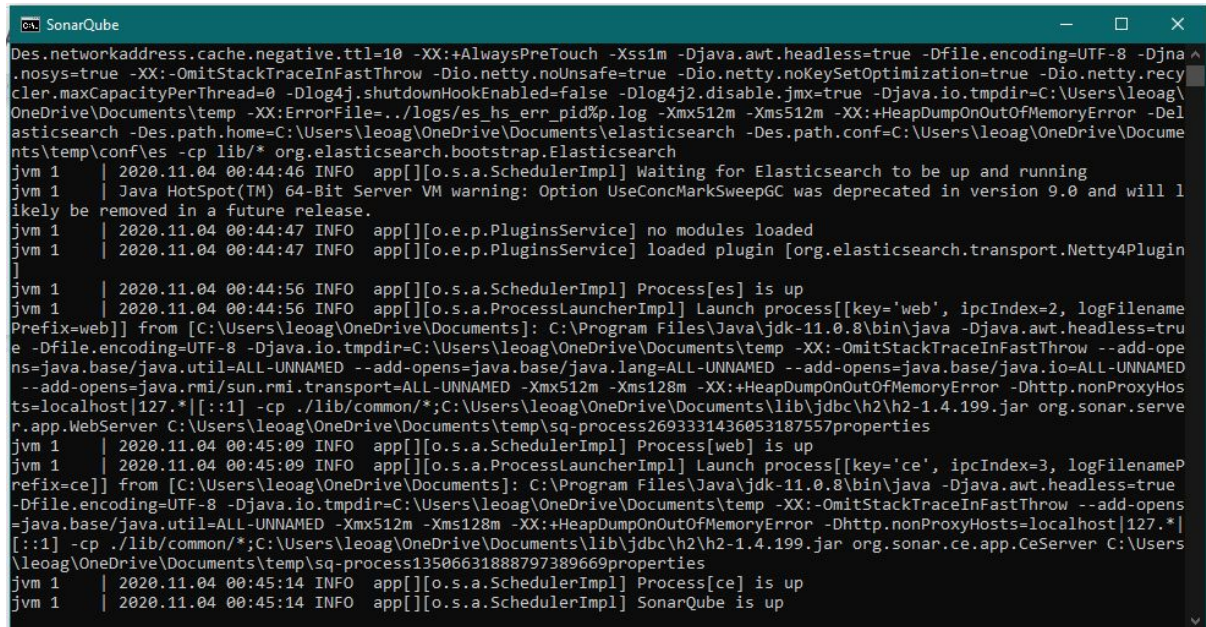
**app.get('/', function(req, res):** Crea la ruta que devuelve un mensaje confirmando que la api y la base de datos están funcionando.

**const server = app.listen(port, () => {}):** Se encarga de realizar la api en el puerto 5000.

**module.exports = { server: server, app: app }:** Exporta el server, app y la base de datos.

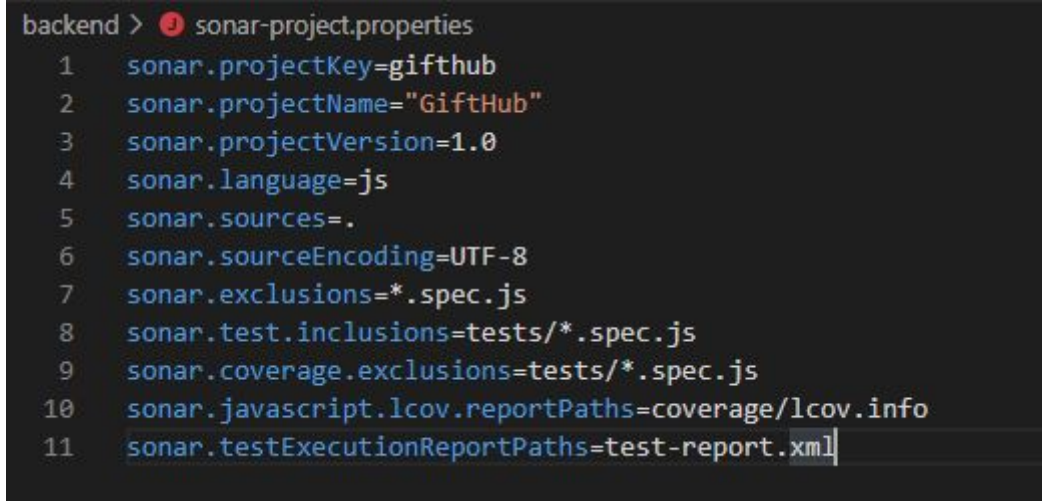
# DOCUMENTACIÓN DE SONAR QUBE

Se realizó una configuración local de SonarQube, en una computadora con sistema operativo windows.



```
Des.networkaddress.cache.negative.ttl=10 -XX:+AlwaysPreTouch -Xss1m -Djava.awt.headless=true -Dfile.encoding=UTF-8 -Djna
.nosys=true -XX:-OmitStackTraceInFastThrow -Dio.netty.noUnsafe=true -Dio.netty.noKeySetOptimization=true -Dio.netty.recy
cler.maxCapacityPerThread=0 -Dlog4j.shutdownHookEnabled=false -Dlog4j2.disable.jmx=true -Djava.io.tmpdir=C:\Users\leoag\
OneDrive\Documents\temp -XX:ErrorFile=../logs/es_hs_err_pid%p.log -Xmx512m -Xms512m -XX:+HeapDumpOnOutOfMemoryError -Del
asticsearch -Des.path.home=C:\Users\leoag\OneDrive\Documents\elasticsearch -Des.path.conf=C:\Users\leoag\OneDrive\Docume
nts\temp\conf\es -cp lib/* org.elasticsearch.bootstrap.Elasticsearch
jvm 1 | 2020.11.04 00:44:46 INFO app[[o.s.a.SchedulerImpl] Waiting for Elasticsearch to be up and running
jvm 1 | Java HotSpot(TM) 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in version 9.0 and will 1
likely be removed in a future release.
jvm 1 | 2020.11.04 00:44:47 INFO app[[o.e.p.PluginsService] no modules loaded
jvm 1 | 2020.11.04 00:44:47 INFO app[[o.e.p.PluginsService] loaded plugin [org.elasticsearch.transport.Netty4Plugin
]
jvm 1 | 2020.11.04 00:44:56 INFO app[[o.s.a.SchedulerImpl] Process[es] is up
jvm 1 | 2020.11.04 00:44:56 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='web', ipcIndex=2, logFilename
Prefix=web]] from [C:\Users\leoag\OneDrive\Documents]: C:\Program Files\Java\jdk-11.0.8\bin\java -Djava.awt.headless=tru
e -Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\leoag\OneDrive\Documents\temp -XX:-OmitStackTraceInFastThrow --add-ope
ns=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED
--add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHos
ts=localhost|127.*|[:1] -cp ./lib/common/*;C:\Users\leoag\OneDrive\Documents\lib\jdbc\h2\h2-1.4.199.jar org.sonar.serve
r.app.WebServer C:\Users\leoag\OneDrive\Documents\temp\sq-process2693331436053187557properties
jvm 1 | 2020.11.04 00:45:09 INFO app[[o.s.a.SchedulerImpl] Process[web] is up
jvm 1 | 2020.11.04 00:45:09 INFO app[[o.s.a.ProcessLauncherImpl] Launch process[[key='ce', ipcIndex=3, logFilenameP
refix=ce]] from [C:\Users\leoag\OneDrive\Documents]: C:\Program Files\Java\jdk-11.0.8\bin\java -Djava.awt.headless=true
-Dfile.encoding=UTF-8 -Djava.io.tmpdir=C:\Users\leoag\OneDrive\Documents\temp -XX:-OmitStackTraceInFastThrow --add-opens
=java.base/java.util=ALL-UNNAMED -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|
[:1] -cp ./lib/common/*;C:\Users\leoag\OneDrive\Documents\lib\jdbc\h2\h2-1.4.199.jar org.sonar.ce.app.CeServer C:\Users
\leoag\OneDrive\Documents\temp\sq-process13506631888797389669properties
jvm 1 | 2020.11.04 00:45:14 INFO app[[o.s.a.SchedulerImpl] Process[ce] is up
jvm 1 | 2020.11.04 00:45:14 INFO app[[o.s.a.SchedulerImpl] SonarQube is up
```

Inicio del servidor de Sonarqube



```
backend > sonar-project.properties
1 sonar.projectKey=gifthumb
2 sonar.projectName="GiftHub"
3 sonar.projectVersion=1.0
4 sonar.language=js
5 sonar.sources=.
6 sonar.sourceEncoding=UTF-8
7 sonar.exclusions=*.spec.js
8 sonar.test.inclusions=tests/*.spec.js
9 sonar.coverage.exclusions=tests/*.spec.js
10 sonar.javascript.lcov.reportPaths=coverage/lcov.info
11 sonar.testExecutionReportPaths=test-report.xml
```

Configuración del archivo de configuración de sonarqube

```

backend > JS sonar-project.js > ...
1  const sonarqubeScanner = require('./lib/sonarqube-scanner');
2
3  sonarqubeScanner({
4    serverUrl: 'http://localhost:9000',
5    options: {
6      'sonar.sources': './app',
7      'sonar.tests': 'tests',
8      'sonar.inclusions': '**',
9      'sonar.test.inclusions': 'tests/*.test.js',
10     'sonar.javascript.lcov.reportPaths': 'coverage/lcov.info',
11     'sonar.testExecutionReportPaths': 'coverage/test-sonar.xml'
12   }
13 }, () => {});

```

Configuración del archivo sonar-project.js para configurar el scanner de sonarqube.

```

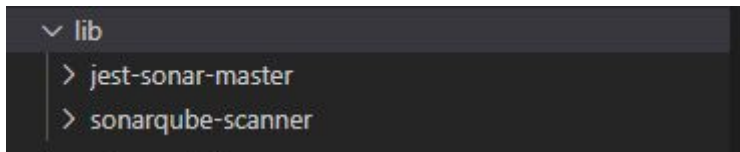
backend > {} package.json > ...
37  },
38  "jest": {
39    "testEnvironment": "node",
40    "coveragePathIgnorePatterns": [
41      "/node_modules/",
42      "/lib/"
43    ],
44    "reporters": [
45      "default",
46      [
47        "./lib/jest-sonar-master",
48        {
49          "outputDirectory": "coverage",
50          "outputName": "test-sonar.xml"
51        }
52      ]
53    ]
54  },
55  "jestSonar": {
56    "reportPath": "coverage",
57    "reportFile": "test-reporter.xml",
58    "indent": 4
59  },

```

Configuración del archivo package.json para utilizar jest como generador de coverage.

Y jestSonar para compatibilidad entre jest y sonarqube.





Nombres de librerías necesarias para correr el código de jest y sonarqube.



Comando de ejecución de sonarqube

File	% Stmts	% Branch	% Funcs	% Lines
All files	82.47	77.31	77.42	83.67
backend	75	50	40	75
index.js	75	50	40	75
backend/app/config	100	100	100	100
db.config.js	100	100	100	100
backend/app/controllers	82.04	78.76	80.52	83.58
cards.js	100	100	93.33	100
compra.js	79.67	67.35	75	80.72
login.js	93.33	100	75	93.33
reglar.js	74.75	78.38	71.43	77.66
usuario.js	95.12	100	88.24	95.12
backend/app/models	77.5	100	50	77.5
cardsValueTasaCambio.js	62.5	100	50	62.5
index.js	100	100	100	100
transacciones.js	62.5	100	50	62.5
usuario.js	62.5	100	50	62.5
backend/app/routes	100	100	100	100
cards.js	100	100	100	100
compra.js	100	100	100	100
login.js	100	100	100	100
regalar.js	100	100	100	100
usuario.js	100	100	100	100
Test Suites: 2 passed, 2 total				
Tests: 56 passed, 56 total				
Snapshots: 0 total				
Time: 6.865 s				
Ran all test suites.				

Ejecución del comando jest --coverage



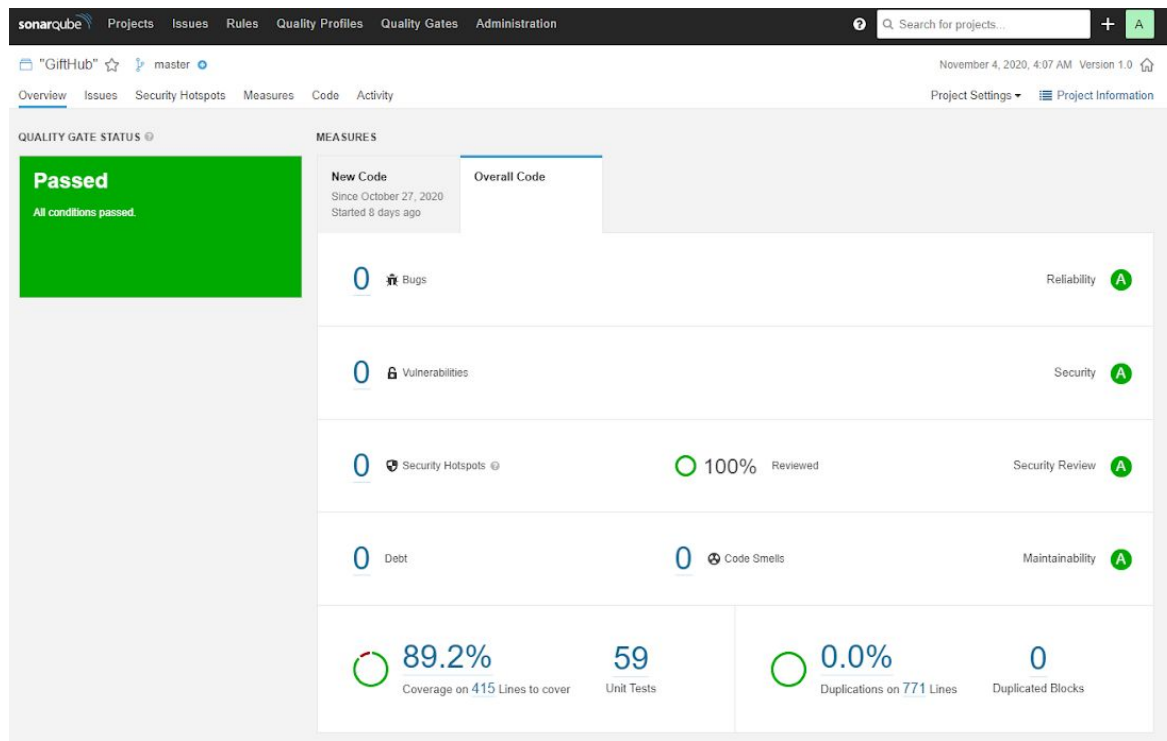
```

[00:51:09] Starting analysis...
[00:51:09] Checking if executable exists: C:\Users\leoag\.sonar\native-sonar-scanner\sonar-scanner-4.4.0.2170-windows\bin\sonar-scanner.bat
[00:51:09] Platform binaries for SonarScanner found. Using it
[00:51:09] Scanner configuration file: C:\Users\leoag\.sonar\native-sonar-scanner\sonar-scanner-4.4.0.2170-windows\bin\.\conf\sonar-scanner.properties
INFO: Project root configuration file: C:\Users\leoag\OneDrive\Documents\GitKraken\github\backend\sonar-project.properties
INFO: SonarScanner 4.4.0.2170
INFO: Java 11.0.3 AdoptOpenJDK (64-bit)
INFO: Windows 10 18.0 amd64
INFO: User cache: C:\Users\leoag\.sonar\cache
INFO: Scanner configuration file: C:\Users\leoag\.sonar\native-sonar-scanner\sonar-scanner-4.4.0.2170-windows\bin\.\conf\sonar-scanner.properties
INFO: Project root configuration file: C:\Users\leoag\OneDrive\Documents\GitKraken\github\backend\sonar-project.properties
INFO: Analyzing on SonarQube server 8.5.0
INFO: Default locale: "en_US", source code encoding: "UTF-8"
INFO: Load global settings
INFO: Load global settings (done) | time=123ms
INFO: Server id: BF41A1F2-AXvzvDx_b_E3OmdDLUHz
INFO: User cache: C:\Users\leoag\.sonar\cache
INFO: Load/download plugins
INFO: Load plugins index
INFO: Load plugins index (done) | time=91ms
INFO: Load/download plugins (done) | time=388ms
INFO: Process project properties
INFO: Process project properties (done) | time=8ms
INFO: Execute project builders
INFO: Execute project builders (done) | time=1ms
INFO: Project key: github
INFO: Base dir: C:\Users\leoag\OneDrive\Documents\GitKraken\github\backend
INFO: Working dir: C:\Users\leoag\OneDrive\Documents\GitKraken\github\backend\scannerwork
INFO: Load project settings for component key: 'github'
INFO: Load project settings for component key: 'github' (done) | time=27ms
INFO: Load quality profiles
INFO: Load quality profiles (done) | time=70ms
INFO: Load active rules
INFO: Load active rules (done) | time=1173ms
INFO: Indexing files...
INFO: Project configuration:
INFO:   Included sources: **
INFO:   Excluded sources: *.spec.js, tests/*.test.js
INFO:   Included tests: tests/*.test.js
INFO:   Excluded sources for coverage: tests/*.spec.js
INFO: Load project repositories
INFO: Load project repositories (done) | time=26ms
INFO: 17 files indexed
INFO: 0 files ignored because of inclusion/exclusion patterns
INFO: 0 files ignored because of scm ignore settings
INFO: Quality profile for js: Sonar way
INFO: ----- Run sensors on module "Github"
INFO: Load metrics repository
INFO: Load metrics repository (done) | time=22ms
INFO: Sensor CSS Rules [cssfamily]
INFO: No CSS, PHP, HTML or VueJS files are found in the project. CSS analysis is skipped.

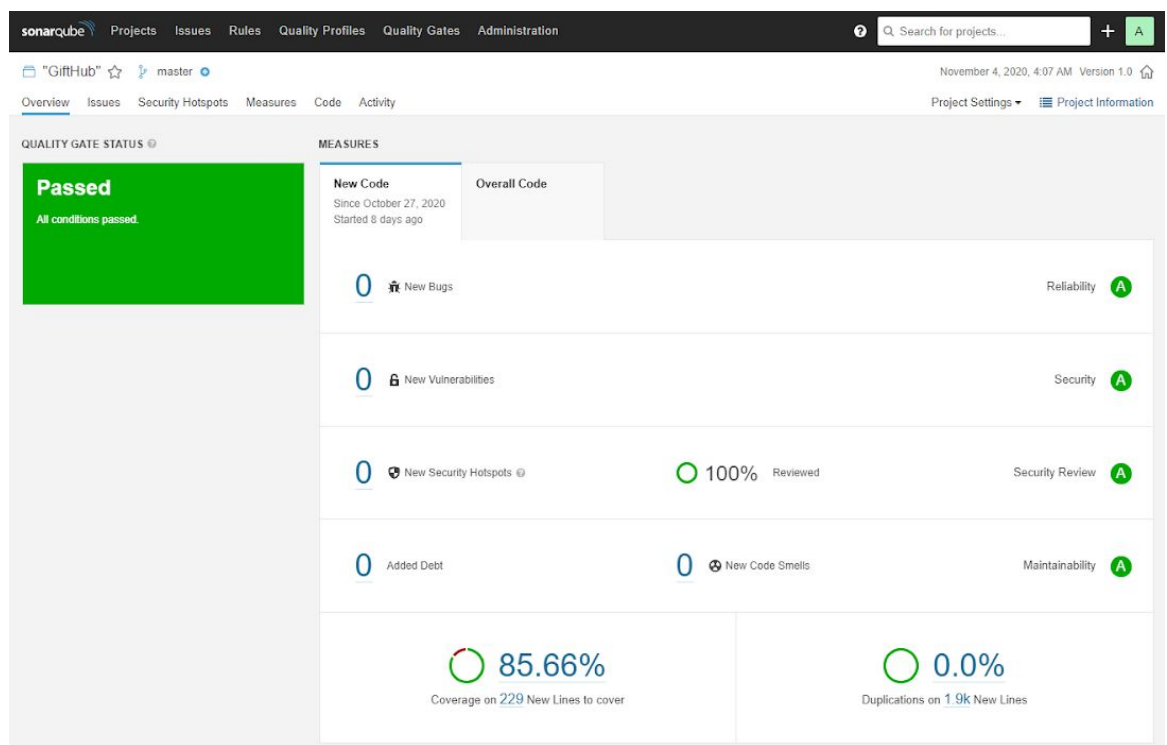
INFO: No report imported, no coverage information will be imported by JaCoCo XML Report Importer
INFO: Sensor JaCoCo XML Report Importer [jacoco] (done) | time=4ms
INFO: Sensor JavaScript analysis [javascript]
INFO: 15 source files to be analyzed
INFO: 15/15 source files have been analyzed
INFO: Java-based frontend sensor [javascript]
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by net.sf.cglib.core.ReflectUtils$1 (file:/C:/Users/leoag/.sonar/cache/2fab92be44e07f1d367f891a552587)
        in method net.sf.cglib.core.ReflectUtils$1.doReflect(java.lang.String,byte[],int,int,java.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of net.sf.cglib.core.ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
INFO: 15 source files to be analyzed
INFO: Java-based frontend sensor [javascript] (done) | time=632ms
INFO: 15/15 source files have been analyzed
INFO: Sensor JavaScript analysis [javascript] (done) | time=6985ms
INFO: Sensor JavaScript/TypeScript Coverage [javascript]
INFO: Analysing [C:\Users\leoag\OneDrive\Documents\GitKraken\github\backend\coverage\lcov.info]
WARN: Found 18 inconsistencies in coverage report. Re-run analyse in debug mode to see details.
INFO: Sensor JavaScript/TypeScript Coverage [javascript] (done) | time=29ms
INFO: Sensor C# Properties [csharp]
INFO: Sensor C# Properties [csharp] (done) | time=1ms
INFO: Sensor JavaXmlSensor [java]
INFO: Sensor JavaXmlSensor [java] (done) | time=1ms
INFO: Sensor HTML [web]
INFO: Sensor HTML [web] (done) | time=2ms
INFO: Sensor VB.NET Properties [vbnet]
INFO: Sensor VB.NET Properties [vbnet] (done) | time=0ms
INFO: Sensor Generic Test Executions Report
INFO: Parsing C:\Users\leoag\OneDrive\Documents\GitKraken\github\backend\coverage\test-sonar.xml
INFO: Imported test execution data for 2 files
INFO: Sensor Generic Test Executions Report (done) | time=56ms
INFO: ----- Run sensors on project
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=12ms
INFO: SCM Publisher SCM provider for this project is: git
INFO: SCM Publisher 3 source files to be analyzed
INFO: SCM Publisher 3/3 source files have been analyzed (done) | time=640ms
INFO: CPD Executor 5 files had no CPD blocks
INFO: CPD Executor Calculating CPD for 10 files
INFO: CPD Executor CPD calculation finished (done) | time=12ms
INFO: Analysis report generated in 68ms, dir size=214 KB
INFO: Analysis report compressed in 413ms, zip size=60 KB
INFO: Analysis report uploaded in 52ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard?id=github
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AXwSBjszLKWakzNDcrF
INFO: Analysis total time: 12.831 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 15.099s
INFO: Final Memory: 14M/54M
INFO: -----
[00:51:24] Analysis finished.

```

Salida esperada de sonarscanner



## Salida de sonarqube



## Vista general del proyecto sonarqube

Size	COMMENTS (%)
New Lines	1,279
Lines of Code	778
Lines	952
Statements	446
Functions	88
Classes	0
Files	15
Comment Lines	26
Comments (%)	3.2%

Comments (%) 3.2%

New Code: Since October 29, 2020

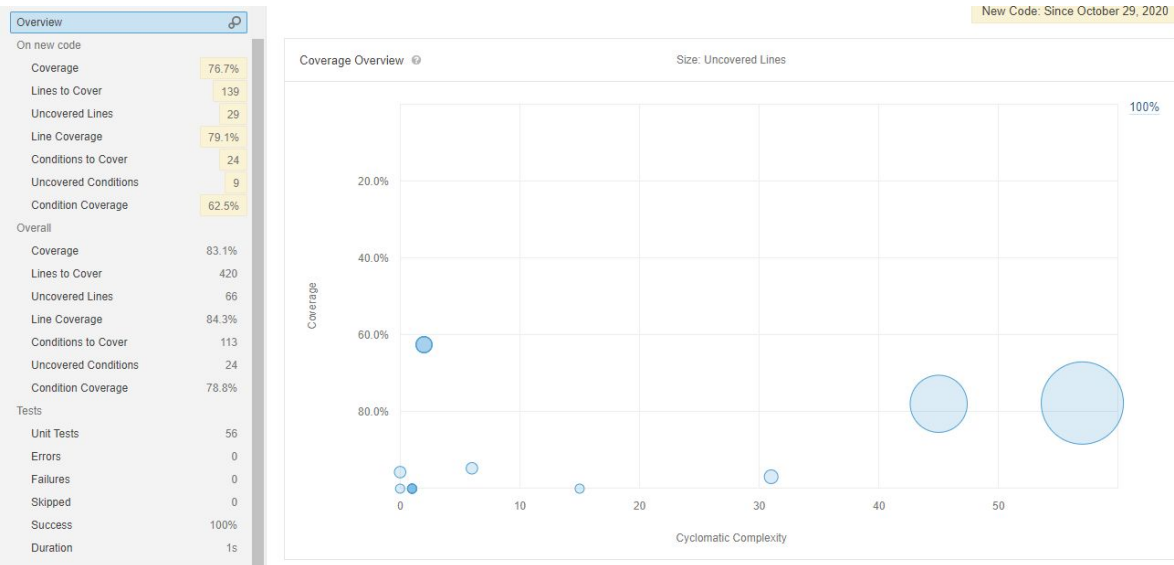
app	3.2%
tests	-

2 of 2 shown

Vista de cantidad de comentarios.

Complexity	CYCLOMATIC COM...
Cyclomatic Complexity	165
Cognitive Complexity	110

Vista de la complejidad ciclomática.



Vista del coverage - "tests"