



Praxis Group #1







Hello!


I am Juan Jose Lopera

I am here because I love to learn.

You can find me at

 Juan jose lopera

 jloperad

 @loper4.exe





Homework #1

Unit Testing

1

- Really scared when I stopped understanding the lecture.
- Hard learning week (API, Springboot, Testing, Mockup, Integration, JUnit, J)
- I sat down and tried to do it.
- It was not that difficult :)
- I can do it
- I didn't understand why the test were important.



Homework #2

Refactoring

- > I understood the lecture
- > not too much to say
- > ez
- > tried creational patterns
- > not ez
- > i understood the important of test



Homework #3

Vagrant

- ◇ Dark day came to my life
- ◇ Couple of tries
- ◇ Maven refused to get installed
- ◇ My teammate made the code to work
- ◇ It doesn't work on my computer
- ◇ I understand the real power of linux





Homework #4 UI Testing

BIG PROBLEMS

Import pain from hell

- Loved the lecture, and the concept of automate things
- trouble agreeing and synchronizing with my partner
- Ez to follow the guide
- Not ez to code with only one hand
- Too much code for a half man, need more time pls
- End up with two fully working repos





Homework #5

lov u docker

I didn't understand a single thing

READ DOCUMENTATION

THE HOMEWORK

Have a lot of problems with the cache

I have to delete the images first upss



Thanks Juan Bolaños

Amazed with the concept of container

Started imagining the possibilities

Lot of problems with ENV





Homework #6

API Testing

+ More coding :c , send help pls

+ Github Copilot comes to rescue <3

+ Time to be creative with those test

+ Intrigued by the security risks



Homework #7 CI/CD

- > AWS... this is going to be fun
- > jenkins? No entiendo
- > Documentation, here I come
- > At least the environment variable problems are now worth it
- > REMINDDER: Thank god for the existence of docker-compose

Homework #8

8

- Terraform was fun
- It really hurts doing the destroy
- Ez delivery
- Env variables really useful again



Small break
nice pic



A dramatic night scene of a forest. In the background, a fire is burning, casting a strong orange and yellow glow that illuminates the silhouettes of the trees. The sky is dark, and the overall atmosphere is intense and mysterious.

Final project

One man, one hand, one challenge

Dockerfile

```
FROM maven:3.8.5-openjdk-17-slim
WORKDIR /usr/src/app
COPY . .
```

```
ARG DB_H
ARG DB_U
ARG DB_P
```

```
ENV DATABASE_HOST = $DB_H
ENV DATABASE_USER = $DB_U
ENV DATABASE_PASSWORD = $DB_P
```

```
CMD mvn spring-boot:run
```

Backend

Jenkinsfile

```
stages{
    stage('Test and Report'){
        agent{
            docker "maven"
        }
        environment{
            DB_HOST = "group1-rds.cqmqj66dxtlw.us-east-1.rds.amazonaws.com"
            DB_USER = "postgres"
            DB_PASSWORD = "postgres"
        }
        steps{
            sh 'mvn clean compile -DDATABASE_HOST=${DB_HOST} -DDATABASE_USER=${DB_USER} -DDATABASE_PASSWORD=${DB_PASSWORD}'
            sh 'mvn clean test -DDATABASE_HOST=${DB_HOST} -DDATABASE_USER=${DB_USER} -DDATABASE_PASSWORD=${DB_PASSWORD}'
            sh 'mvn surefire-report:report -DDATABASE_HOST=${DB_HOST} -DDATABASE_USER=${DB_USER} -DDATABASE_PASSWORD=${DB_PASSWORD}'
            sh 'cp -R target/site/surefire-report.html report/'

            publishHTML (
                target : [
                    allowMissing: false,
                    alwaysLinkToLastBuild: true,
                    keepAll: true,
                    reportDir: 'report',
                    reportFiles: 'surefire-report.html',
                    reportName: 'Unit test Report'
                ]
            )
        }
    }

    stage('Build Backend image'){
        agent any
        environment{
            DB_HOST = "group1-rds.cqmqj66dxtlw.us-east-1.rds.amazonaws.com"
            DB_USER = "postgres"
            DB_PASSWORD = "postgres"
        }
        steps{
            sh 'docker build --build-arg DB_H=${DB_HOST} --build-arg DB_U=${DB_USER} --build-arg DB_P=${DB_PASSWORD} -t jloperad/praxis-gildedrose_backend:latest .'
        }
    }

    stage('Docker Push Image') {
        agent any
        steps {
            withCredentials([usernamePassword(credentialsId: 'dockerHub', passwordVariable: 'dockerHubPassword', usernameVariable: 'dockerHubUser')]) {
                sh "docker login -u ${env.dockerHubUser} -p ${env.dockerHubPassword}"
                sh "docker push jloperad/praxis-gildedrose_backend:latest"
                sh "docker logout"
            }
        }
    }
}
```

Dockerfile

```
FROM node:16.10
WORKDIR /app
COPY . .
RUN npm install
CMD npm start
```

FrontEnd

Jenkinsfile

```
pipeline{
  agent any

  stages{
    stage('Build frontend image'){
      agent any
      steps{
        sh 'docker build -t jloperad/praxis-gildedrose_frontend .'
      }
    }

    stage('Docker Push image') {
      agent any
      steps {
        withCredentials([usernamePassword(credentialsId: 'dockerHub', passwordVariable: 'dockerHubPassword', usernameVariable: 'dockerHubUser')]) {
          sh "docker login -u ${env.dockerHubUser} -p ${env.dockerHubPassword}"
          sh 'docker push jloperad/praxis-gildedrose_frontend:latest'
          sh 'docker logout'
        }
      }
    }
  }
}
```


Docker-compose

```
version: "3.3"

services:
  backend:
    container_name: "backend"
    image: jloperad/praxis-gildedrose_backend:latest
    ports:
      - 8081:8080
    environment:
      - DATABASE_HOST=group1-rds.cqmqj6dxtlw.us-east-1.rds.amazonaws.com
      - DATABASE_USER=postgres
      - DATABASE_PASSWORD=postgres

  frontend:
    container_name: "frontend"
    image: jloperad/praxis-gildedrose_frontend:latest
    ports:
      - "4200:4200"
    environment:
      - BACKEND_HOST=backend
    depends_on:
      - backend
```

CD Pipeline

Jenkinsfile

```
pipeline {
  agent any

  stages {
    stage('Delete Running Containers and images') {
      steps {
        sh 'docker rm -f $(docker ps -aq) || true'
        sh 'docker rmi -f $(docker images -a -q) || true'
      }
    }

    stage('Pull latest images') {
      steps {
        sh 'docker pull jloperad/praxis-gildedrose_backend'
        sh 'docker pull jloperad/praxis-gildedrose_frontend'
      }
    }

    stage('Run images') {
      steps {
        sh 'docker-compose up -d'
      }
    }
  }
}
```

Jenkinsfile

```
pipeline {
  agent any

  options {
    ansiColor('xterm')
  }

  stages{
    stage("Install dependencies") {
      steps {
        dir("UI-Tests"){
          sh '''npm ci
          npm install mocha mochawesome cypress-mochawesome-reporter --save-dev
          npm install cypress-multi-reporters --save-dev
          npm install mochawesome-merge --save-dev
          npm install mochawesome-report-generator --save-dev
          ...
        }
      }
    }
  }
}
```

```
    stage("Test the UI") {
      steps {
        dir("UI-Tests"){
          sh 'TERM=dumb'
          sh 'npm test'
        }
      }
    }
  }

  post{
    always{
      publishHTML (
        target : [
          allowMissing: false,
          alwaysLinkToLastBuild: true,
          keepAll: true,
          reportDir: 'UI-Tests/cypress/reports',
          reportFiles: 'report.html',
          reportName: 'UI-Tests Report'
        ]
      )
    }
  }
}
```

UI Testing



Jenkinsfile

```
pipeline {
  agent any

  options {
    ansiColor('xterm')
  }

  stages{
    stage("Install dependencies") {
      steps {
        dir("API-tests"){
          sh 'npm ci'
          sh 'npm install mocha mochawesome cypress-mochawesome-reporter --save-dev'
        }
      }
    }
    stage("Test the UI") {
      steps {
        dir("API-tests"){
          sh 'npm test'
        }
      }
    }
  }
}
```

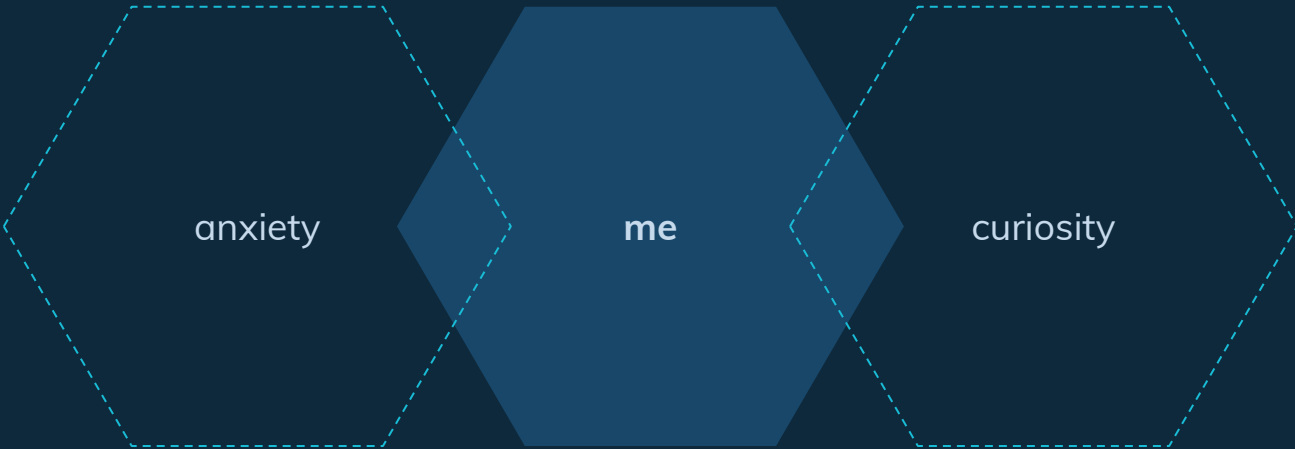
```
post{
  always{
    publishHTML (
      target : [
        allowMissing: false,
        alwaysLinkToLastBuild: true,
        keepAll: true,
        reportDir: 'API-tests/report/',
        reportFiles: 'ApiTesting.html',
        reportName: 'API-Tests Report',
        reportTitles: 'API-Tests Report',
      ]
    )
  }
}
```

API Testing






Resume



anxiety

me

curiosity





Conclusions

- ◇ I learned a lot
- ◇ Understand a little more of the big picture
- ◇ From big mistakes , comes great knowledge
- ◇ It's important to understand the why its failing, not only the solution

