

Sistema de Moeda Estudantil Gerência de Configuração

Ana Flávia de Carvalho Santos
Júlia Moreira Nascimento
Lorrayne Marayze Silva de Oliveira
Pedro Henrique Pires Rodrigues
Pedro Henrique de Almeida Costa



Objetivo do Trabalho

- Criar um Plano de Gerenciamento de Configuração (PGC)
- Implementar testes automatizados:
 - Unidade
 - Integração
 - Aceitação
- Configurar pipeline CI/CD com:
 - Build
 - Execução de testes
 - Empacotamento
 - Deploy (Docker)
- Aplicar práticas de DevOps e Integração Contínua

Tecnologias Utilizadas

Backend

- Spring Boot
- Maven
- JDK 17 (Eclipse Temurin)

Frontend

- React
- TypeScript

Infra e DevOps

- Git + GitHub
- CircleCI
- Docker
- MySQL

Funcionamento Geral

- O SyncCoin é um sistema de moeda estudantil usado para registrar e distribuir moedas entre alunos.
- O frontend (React + TypeScript) envia requisições para a API.
- O backend (Spring Boot) trata regras de negócio e acessa o banco.
- O banco de dados MySQL armazena usuários, transações e saldos.
- A aplicação é empacotada em containers Docker para facilitar execução e deploy.

Escopo PGC

- Código-fonte do backend (Spring Boot) e frontend (React).
- Documentação do projeto: requisitos, visão, arquitetura, testes.
- Scripts e estrutura do banco de dados.
- Arquivos de configuração e ambiente Docker.
- Artefatos de build, releases e empacotamento.
- Processos de controle de mudança (Issues + Pull Requests).
- Aplica-se a todos os membros do projeto.

Responsabilidades

Gerente de Projeto

- Aprova baselines.
- Autoriza mudanças de escopo.
- Acompanha cronograma.

Gerente de Configuração

- Administra o repositório GitHub.
- Define branches, releases e backups.
- Garante organização e versionamento.

Desenvolvedores

Realizam commits corretos.

Criam Pull Requests.

Resolvem conflitos e impleme

Responsabilidades

Desenvolvedores

- Realizam commits corretos.
- Criam Pull Requests.
- Resolvem conflitos e implementam mudanças.

Identificação da Configuração

- Itens de configuração divididos por módulos:
 - backend, frontend, banco, documentação, scripts.
- Versionamento semântico: MAJOR.MINOR.PATCH (ex: v1.0.0).
- Artefatos: código, imagens Docker, pacotes Maven.
- Uso de tags para marcar baselines (v0.1.0, v0.5.0, v0.8.0, v1.0.0).
- Estrutura padronizada de diretórios.

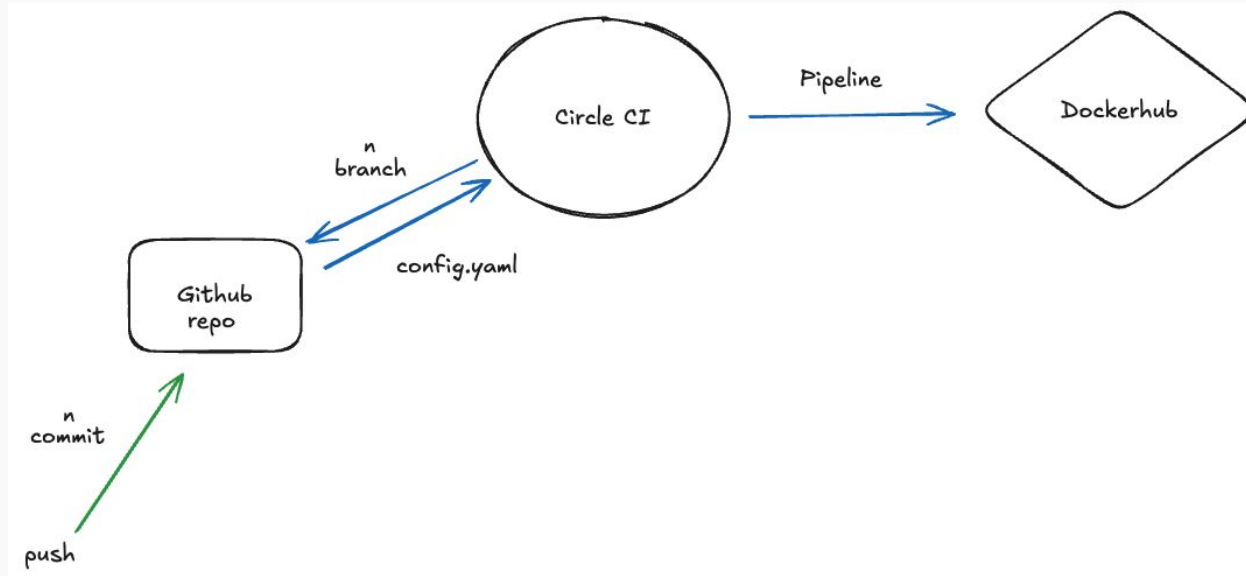
Tarefas do RUP Adaptadas ao Projeto

- Configurar Ambiente CM
 - Definição do repositório, .gitignore e proteção da main
- Estruturar Modelo de Implementação
 - Diretórios organizados: /src, /docs, /db.
- Espaço de Trabalho de Integração
 - Branch main => integração contínua.
- Espaços de Trabalho de Desenvolvimento
 - Branches feature: feature/nova-funcionalidade.

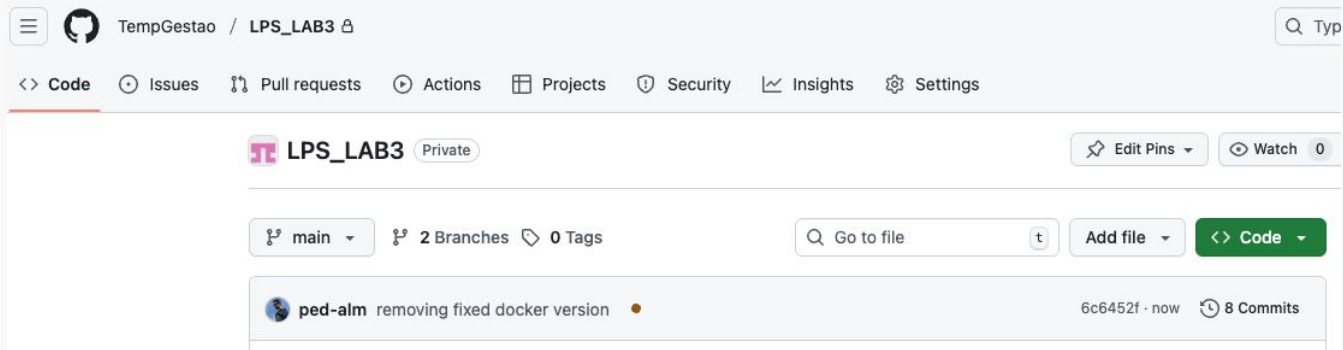
Fluxo de Controle de Mudanças

1. Uma mudança começa como Issue no GitHub.
2. Implementação ocorre em uma branch separada.
3. Pull Request enviado com descrição e evidências.
4. CircleCI executa build + testes automaticamente.
5. Após aprovação, o PR é integrado na main.
6. Baselines são criadas via Tags.

Funcionamento



Github Integration



LPS_LAB3
Private

Edit Pins
Watch 0

main
2 Branches
0 Tags

t
Add file
Code

ped-alm
removing fixed docker version
6c6452f · 1 minute ago
8 Commits

.circleci	removing fixed docker version	1 minute ago
.vscode	adicionando teste de aceitacao e melhorando pipeline	33 minutes ago
Artefatos	copy	1 hour ago
api	Fix acceptance test	27 minutes ago
my-app		
.gitignore		
README.md		
SETUP.md	copy	1 hour ago

Some checks haven't completed yet

1 pending check

ci/circleci: build Pending
Details

Circle CI

The screenshot displays the Circle CI web interface. At the top, the navigation bar includes the Circle CI logo and a user profile icon. The left sidebar contains a vertical menu with icons for Home, Pipelines, Projects, Deploys, Insights, Runners, Org, and Plan. The main content area shows the breadcrumb path: Dashboard > Project > Branch > Workflow. The current view is for the 'ci-cd-pipeline' workflow, which is in a 'Running' state. A notification banner at the top of the workflow details mentions 'New: Smarter Testing' with a link to 'Join Waitlist'. Below this, a table provides details for the current build: Duration (1m 54s), Branch (main), Commit (6c6452f), and Author & Message (removing fixed docker version). The workflow steps are visualized as a sequence of boxes: 'build 25s' (completed with a green checkmark), 'unit-tests 1m 4s' (completed with a green checkmark), 'integration-tests 19s' (in progress with a blue play button), and 'acceptance-tests' (pending with a grey circle and number 8). A progress bar at the bottom indicates the overall progress of the pipeline.

circleci

Dashboard Project Branch Workflow

All Pipelines > LPS_LAB3 > main > ci-cd-pipeline

ci-cd-pipeline Running Insights Rerun

New: Smarter Testing - Get faster feedback on tests with Test Impact Analysis and Dynamic Test Splitting. [Join Waitlist](#)

Duration	Branch	Commit	Author & Message
1m 54s	main	6c6452f	removing fixed docker version

build 25s — unit-tests 1m 4s — integration-tests 19s — acceptance-tests

Github Integration

The screenshot displays the GitHub interface for a repository named 'LPS_LAB3'. The repository is private and has 2 branches and 0 tags. A commit by user 'ped-alm' titled 'removing fixed docker version' is shown, with a commit hash of '6c6452f' and a timestamp of '18 minutes ago'. The commit message is 'removing fixed docker version'. The file list includes: .circleci, .vscode, Artefatos, api, my-app, .gitignore, README.md, and SETUP.md. A modal window titled 'All checks have passed' is overlaid on the right, indicating that 6 successful checks were completed. The checks listed are: ci/circleci: acceptance-tests, ci/circleci: build, ci/circleci: build-docker-image, ci/circleci: deploy, ci/circleci: integration-tests, and ci/circleci: unit-tests. Each check has a green checkmark and a 'Details' link.

TempGestao / LPS_LAB3

Code Issues Pull requests Actions Projects Security Insights Settings

LPS_LAB3 Private

Edit Pins Watch 0

main 2 Branches 0 Tags

Go to file Add file Code

ped-alm removing fixed docker version ✓ 6c6452f · 18 minutes ago 8 Commits

.circleci removing fixed docker version 18 minutes ago

.vscode

Artefatos

api

my-app

.gitignore

README.md

SETUP.md

README

All checks have passed

6 successful checks

- ✓ ci/circleci: acceptance-tests Details
- ✓ ci/circleci: build Details
- ✓ ci/circleci: build-docker-image Details
- ✓ ci/circleci: deploy Details
- ✓ ci/circleci: integration-tests Details
- ✓ ci/circleci: unit-tests Details

Config:

```
.circleci > ! config.yml
```

```
285
286 workflows:
287   version: 2
288   ci-cd-pipeline:
289     jobs:
290       - build:
291         filters:
292           branches:
293             only:
294               - main
295               - master
296       - unit-tests:
297         requires:
298           - build
299         filters:
300           branches:
301             only:
302               - main
303               - master
304       - integration-tests:
305         requires:
306           - unit-tests
307         filters:
308           branches:
309             only:
310               - main
311               - master
```

```
312 - acceptance-tests:
313   requires:
314     - integration-tests
315   filters:
316     branches:
317       only:
318         - main
319         - master
320 - build-docker-image:
321   requires:
322     - acceptance-tests
323   filters:
324     branches:
325       only:
326         - main
327         - master
328 - deploy:
329   requires:
330     - build-docker-image
331   filters:
332     branches:
333       only:
334         - main
335         - master
336
```

%I to chat, %K to generate

Config: build

```
.circleci > ! config.yml
You, 5 minutes ago | 2 authors (You and one other)
1  version: 2.1
2  jobs:
3    build:
4      docker:
5        - image: maven:3.9-eclipse-temurin-17
6      working_directory: ~/project/api
7      steps:
8        - checkout:
9          path: ~/project
10       - run:
11         name: Verificar versões Java e Maven
12         command: |
13           java -version
14           mvn -version
15       - restore_cache:
16         keys:
17           - maven-v1-{{ checksum "pom.xml" }}
18           - maven-v1-
19       - run:
20         name: Baixar dependências Maven
21         command: mvn dependency:go-offline -B
22       - save_cache:
23         paths:
24           - ~/.m2
25         key: maven-v1-{{ checksum "pom.xml" }}
26       - run:
27         name: Compilar aplicação
28         command: mvn clean compile -B
29         environment:
30           SPRING_PROFILES_ACTIVE: test
```

Config:unit-tests

```
31 unit-tests:
32   docker:
33     - image: maven:3.9-eclipse-temurin-17
34     working_directory: ~/project/api
35   steps:
36     - checkout:
37       path: ~/project
38     - restore_cache:
39       keys:
40         - maven-v1-{{ checksum "pom.xml" }}
41         - maven-v1-
42     - run:
43       name: Executar testes de unidade
44       command: |
45         # Executa testes de unidade (services, models, dtos, exceptions, security)
46         # Exclui testes de integração e aceitação
47         mvn test \
48           -Dtest="*ServiceTest,*Tests,*DTOTest,*ExceptionTest,*FilterTest,*ProviderTest" \
49           -DfailIfNoTests=false \
50           -B
51       environment:
52         SPRING_PROFILES_ACTIVE: test
53     - run:
54       name: Resumo dos testes de unidade
55       command: |
56         if [ -f target/surefire-reports/TEST-*.xml ]; then
57           echo "=== Resumo dos Testes de Unidade ==="
58           find target/surefire-reports -name "TEST-*.xml" -exec grep -h "testsuite" {} \; | head -
59         fi
60       when: always
61     - store_test_results:
62       path: ~/project/api/target/surefire-reports
63     - store_artifacts:
64       path: ~/project/api/target/surefire-reports
65       destination: unit-test-results
```

Config: integration-tests

```
66 integration-tests:
67   docker:
68     - image: maven:3.9-eclipse-temurin-17
69     working_directory: ~/project/api
70   steps:
71     - checkout:
72       path: ~/project
73     - restore_cache:
74       keys:
75         - maven-v1-{{ checksum "pom.xml" }}
76         - maven-v1-
77     - run:
78       name: Executar testes de integração
79       command: |
80         # Executa testes de integração (controllers e testes com Spring context)
81         mvn test \
82           -Dtest="*ControllerTest,ApiApplicationTests" \
83           -DfailIfNoTests=false \
84           -B
85       environment:
86         SPRING_PROFILES_ACTIVE: test
87     - run:
88       name: Resumo dos testes de integração
89       command: |
90         if [ -f target/surefire-reports/TEST-*.xml ]; then
91           echo "=== Resumo dos Testes de Integração ==="
92           find target/surefire-reports -name "TEST-*.xml" -exec grep -h "testsuite" {} \; | head -10
93         fi
94       when: always
95     - store_test_results:
96       path: ~/project/api/target/surefire-reports
97     - store_artifacts:
98       path: ~/project/api/target/surefire-reports
99       destination: integration-test-results
```

Config: acceptance-tests

```
100 acceptance-tests:
101     docker:
102         - image: maven:3.9-eclipse-temurin-17
103     working_directory: ~/project/api
104     steps:
105         - checkout:
106             path: ~/project
107         - restore_cache:
108             keys:
109                 - maven-v1-{{ checksum "pom.xml" }}
110                 - maven-v1-
111         - run:
112             name: Executar testes de aceitação
113             command: |
114                 # Executa testes de aceitação (end-to-end)
115                 mvn test -Dtest="*AcceptanceTest" -B
116             environment:
117                 SPRING_PROFILES_ACTIVE: test
118         - run:
119             name: Resumo dos testes de aceitação
120             command: |
121                 if [ -f target/surefire-reports/TEST-*.xml ]; then
122                     echo "=== Resumo dos Testes de Aceitação ==="
123                     find target/surefire-reports -name "TEST-*.xml" -exec grep -h "testsuite" {} \;
124                     echo ""
125                     echo "=== Detalhes dos Testes ==="
126                     find target/surefire-reports -name "*.txt" -exec cat {} \;
127                 fi
128             when: always
129         - store_test_results:
130             path: ~/project/api/target/surefire-reports
131         - store_artifacts:
132             path: ~/project/api/target/surefire-reports
133             destination: acceptance-test-results
```

Config: build-docker-image

```
134 build-docker-image:
135   docker:
136     - image: docker:24.0.5
137   working_directory: ~/project/api
138   steps:
139     - checkout:
140       path: ~/project
141     - setup_remote_docker
142     - run:
143       name: Build da imagem Docker
144       command: |
145         docker build -t student-currency-system:latest .
146         docker images student-currency-system:latest
147     - run:
148       name: Testar container Docker
149       command: |
150         # Inicia o container em background
151         docker run -d --name test-container -p 8080:8080 student-currency-system:latest
152         sleep 30
153         # Verifica se a aplicação está respondendo
154         curl -f http://localhost:8080/actuator/health || echo "Health check failed"
155         # Para e remove o container
156         docker stop test-container
157         docker rm test-container
158     - run:
159       name: Salvar imagem Docker
160       command: |
161         docker save student-currency-system:latest | gzip > student-currency-system.tar.gz
162     - store_artifacts:
163       path: ~/project/api/student-currency-system.tar.gz
164       destination: docker-image
```

Config:deploy

```
165 deploy:
166   docker:
167     - image: docker:24.0.5
168   working_directory: ~/project/api
169   steps:
170     - checkout:
171       path: ~/project
172     - setup_remote_docker
173     - run:
174       name: Login no Docker Hub
175       command: |
176         echo "=== Fazendo login no Docker Hub ==="
177         echo "$DOCKERHUB_PASSWORD" | docker login -u "$DOCKERHUB_USERNAME" --password-stdin
178         echo "Login realizado com sucesso!"
179     - run:
180       name: Build da imagem Docker
181       command: |
182         echo "=== Construindo imagem Docker ==="
183         docker build -t ${DOCKERHUB_USERNAME}/student-currency-system:${CIRCLE_SHA1} .
184         docker build -t ${DOCKERHUB_USERNAME}/student-currency-system:latest .
185         echo "Imagens construídas:"
186         docker images | grep student-currency-system
187     - run:
188       name: Push para Docker Hub
189       command: |
190         echo "=== Fazendo push para Docker Hub ==="
191         echo "Push da imagem com tag: ${CIRCLE_SHA1}"
192         docker push ${DOCKERHUB_USERNAME}/student-currency-system:${CIRCLE_SHA1}
193         echo "Push da imagem latest"
194         docker push ${DOCKERHUB_USERNAME}/student-currency-system:latest
195         echo "=====
196         echo "Deploy realizado com sucesso!"
197         echo "Imagem disponível em:"
198         echo "  - ${DOCKERHUB_USERNAME}/student-currency-system:${CIRCLE_SHA1}"
199         echo "  - ${DOCKERHUB_USERNAME}/student-currency-system:latest"
200         echo "=====
201     - run:
202       name: Logout do Docker Hub
203       command: |
204         docker logout
205         echo "Logout realizado"
206   when: always
```

CircleCI: Sucesso

The screenshot displays the CircleCI dashboard for a project named 'LPS_LAB3'. The interface includes a sidebar with navigation links: Home, Pipelines, Projects, Deploys, Insights, Runners, Org, and Plan. The main content area shows the 'All Pipelines' view for 'LPS_LAB3'. A notification banner at the top reads: 'New: Smarter Testing - Get faster feedback on tests with Test Impact Analysis and Dynamic Test Splitting. [Join Waitlist](#)'. Below this, there are filters for 'All pipelines', 'LPS_LAB3', 'All branches', 'Cutoff date', and 'All statuses'. A 'Display options' button is also present. The pipeline list shows a single pipeline named 'build-and-test' with a status of 'Success'. The pipeline details are expanded, showing a list of jobs: 'build', 'unit-tests', 'integration-tests', 'acceptance-tests', 'build-docker-image', and 'deploy'. Each job has a green checkmark indicating success. The duration for each job is listed on the right.

Pipeline	Status	Workflow	Checkout source	Trigger event	Start	Duration	Actions
build-and-test 7	Success	ci-cd-pipeline	main 6c6452f removing fixed docker version	Push commit pushed	20m ago	6m 53s	🔄 🔄 ✕ ...
Jobs							
✓ build	22					25s	
✓ unit-tests	23					1m 4s	
✓ integration-tests	24					58s	
✓ acceptance-tests	25					54s	
✓ build-docker-image	26					1m 55s	
✓ deploy	27					1m 13s	

CircleCI: Falha

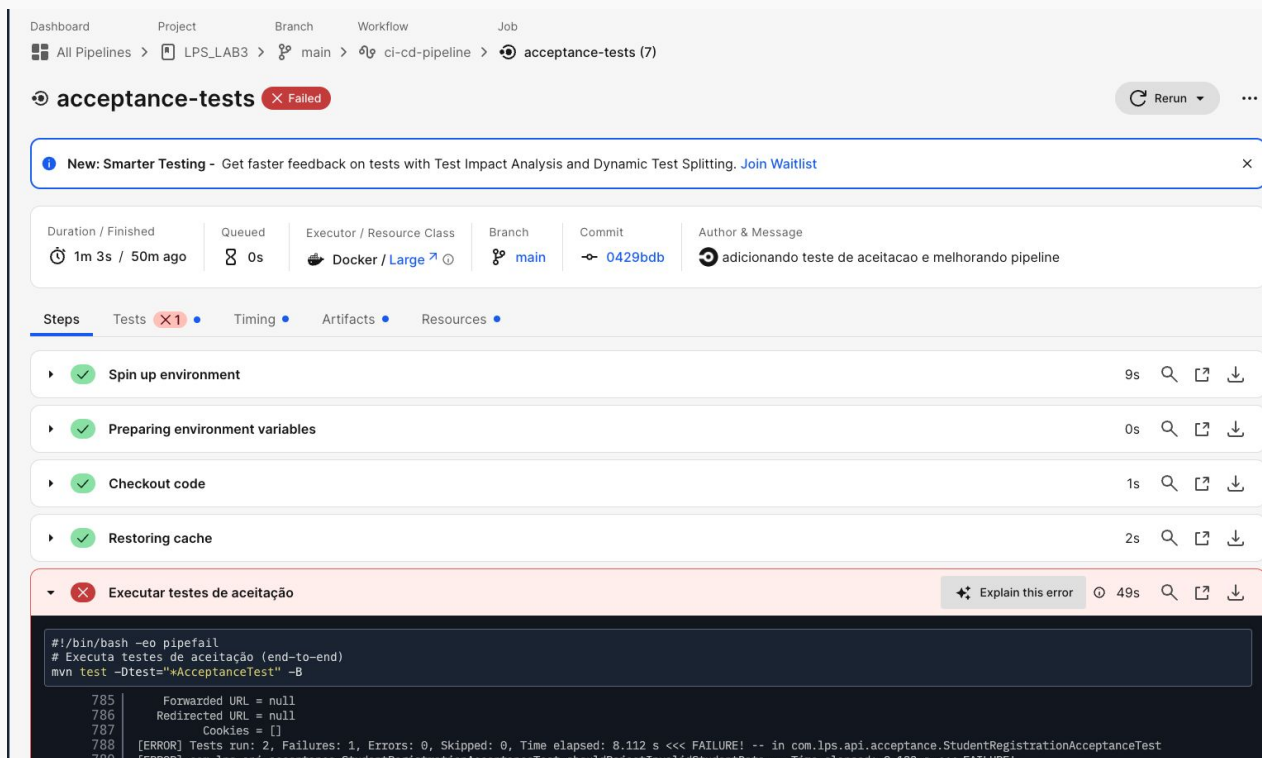
build-and-test 6 ❌ Failed ci-cd-pipeline main Push commit pushed 30m ago 3m 18s 🔄 🔍 🗑️ ⋮

e75738f fixing docker hub

Jobs

✓ build	16	25s
✓ unit-tests	17	57s
✓ integration-tests	18	52s
✓ acceptance-tests	19	55s
❌ build-docker-image	20	0s
🔧 deploy	21	

CircleCI: Teste quebrado



The screenshot displays the CircleCI interface for a job named 'acceptance-tests'. The job status is 'Failed', indicated by a red 'X' icon. The breadcrumb navigation shows the path: Dashboard > Project > LPS_LAB3 > main > ci-cd-pipeline > acceptance-tests (7). A notification banner at the top promotes 'New: Smarter Testing' with Test Impact Analysis and Dynamic Test Splitting. Below this, a table provides job details: Duration (1m 3s / 50m ago), Queued (0s), Executor (Docker / Large), Branch (main), Commit (0429bdb), and Author & Message (adicionando teste de aceitacao e melhorando pipeline). The 'Steps' section shows four successful steps: 'Spin up environment' (9s), 'Preparing environment variables' (0s), 'Checkout code' (1s), and 'Restoring cache' (2s). The final step, 'Executar testes de aceitação', is marked as failed with a red 'X' and a duration of 49s. An 'Explain this error' button is available for this step. The terminal output for the failed step shows the command 'mvn test -Dtest="*AcceptanceTest" -B' and a detailed error message: '[ERROR] Tests run: 2, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 8.112 s <<< FAILURE! -- in com.lps.api.acceptance.StudentRegistrationAcceptanceTest [ERROR] com.lps.api.acceptance.StudentRegistrationAcceptanceTest.shouldRedirectInvalidStudentData -- Time elapsed: 0.112 s <<< FAILURE!'. The interface also includes tabs for 'Steps', 'Tests' (showing 1 failed), 'Timing', 'Artifacts', and 'Resources'.

Dashboard > Project > LPS_LAB3 > main > ci-cd-pipeline > acceptance-tests (7)

acceptance-tests Failed Rerun ...

New: Smarter Testing - Get faster feedback on tests with Test Impact Analysis and Dynamic Test Splitting. [Join Waitlist](#)

Duration / Finished	Queued	Executor / Resource Class	Branch	Commit	Author & Message
1m 3s / 50m ago	0s	Docker / Large	main	0429bdb	adicionando teste de aceitacao e melhorando pipeline

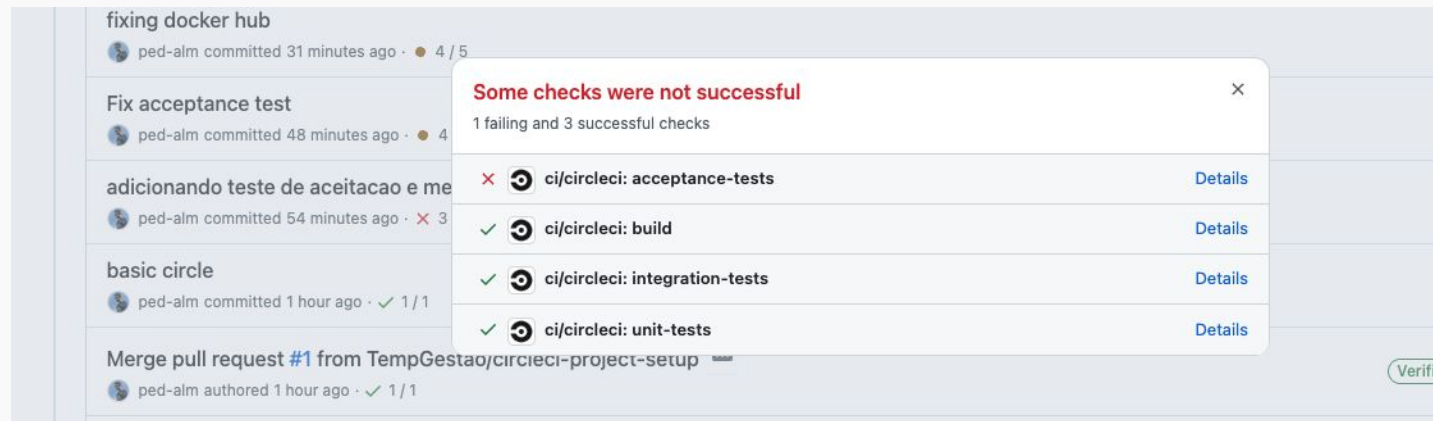
Steps Tests 1 Timing Artifacts Resources

- ✓ Spin up environment 9s
- ✓ Preparing environment variables 0s
- ✓ Checkout code 1s
- ✓ Restoring cache 2s
- ✗ Executar testes de aceitação 49s Explain this error

```
#!/bin/bash -eo pipefail
# Executa testes de aceitação (end-to-end)
mvn test -Dtest="*AcceptanceTest" -B

785 Forwarded URL = null
786 Redirected URL = null
787 Cookies = []
788 [ERROR] Tests run: 2, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 8.112 s <<< FAILURE! -- in com.lps.api.acceptance.StudentRegistrationAcceptanceTest
789 [ERROR] com.lps.api.acceptance.StudentRegistrationAcceptanceTest.shouldRedirectInvalidStudentData -- Time elapsed: 0.112 s <<< FAILURE!
```

Github: Teste quebrado



The screenshot displays a GitHub pull request interface. On the left, a list of commits is visible: 'fixing docker hub', 'Fix acceptance test', 'adicionando teste de aceitacao e me', 'basic circle', and 'Merge pull request #1 from TempGestao/circleci-project-setup'. The 'Fix acceptance test' commit is selected. On the right, a modal window titled 'Some checks were not successful' is open, showing a summary of 1 failing and 3 successful checks. The failing check is 'ci/circleci: acceptance-tests', while the others ('ci/circleci: build', 'ci/circleci: integration-tests', and 'ci/circleci: unit-tests') are successful. Each check has a 'Details' link. A 'Verify' button is visible at the bottom right of the modal.

fixing docker hub
ped-alm committed 31 minutes ago · 4 / 5

Fix acceptance test
ped-alm committed 48 minutes ago · 4

adicionando teste de aceitacao e me
ped-alm committed 54 minutes ago · 3

basic circle
ped-alm committed 1 hour ago · 1 / 1

Merge pull request #1 from TempGestao/circleci-project-setup
ped-alm authored 1 hour ago · 1 / 1

Some checks were not successful
1 failing and 3 successful checks

- ❌ ci/circleci: acceptance-tests [Details](#)
- ✅ ci/circleci: build [Details](#)
- ✅ ci/circleci: integration-tests [Details](#)
- ✅ ci/circleci: unit-tests [Details](#)

Verifi

DockerHub

The screenshot displays the Docker Hub 'My Hub' page for a user named 'pedr0almeida'. The interface includes a top navigation bar with 'Explore' and 'My Hub' tabs, a search bar, and user profile icons. A left sidebar lists navigation options: Repositories, Hardened Images, Collaborations, Settings, Default privacy, Notifications, Billing, Usage, Pulls, and Storage. The main content area is titled 'Repositories' and shows a list of repositories within the 'pedr0almeida' namespace. A search bar and a dropdown for 'All content' are present. A table lists the repository 'pedr0almeida/student-currency-system' with details on last pushed time, content type, visibility, and scout status. A 'Create a repository' button is located in the top right of the repository list.

Repositories

All repositories within the **pedr0almeida** namespace.

Search by repository name All content Create a repository

Name	Last Pushed ↑	Contains	Visibility	Scout
pedr0almeida/student-currency-system	17 minutes ago	IMAGE	Public	Inactive

1~1 of 1



Obrigado !