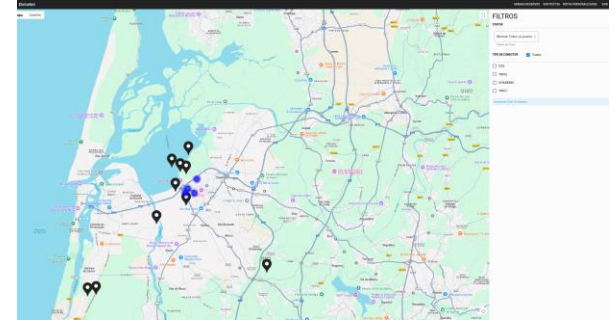
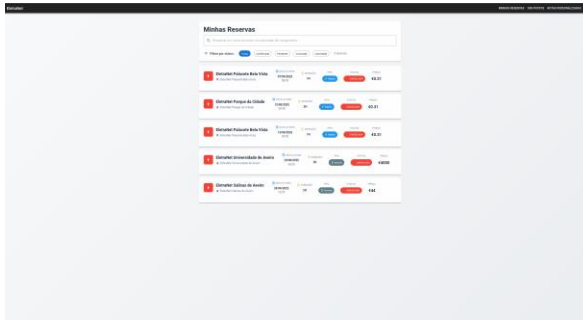


Eletranet

Platafomsrma para carregamentos de veículos elétricos

Plataforma integrada
para gestão, operação e
carregamentos.

Conceito



 **Painel do Operador**
Eletrinet Services

 DASHBOARD

 ESTAÇÕES

 RESERVAS

Total Estações
20 

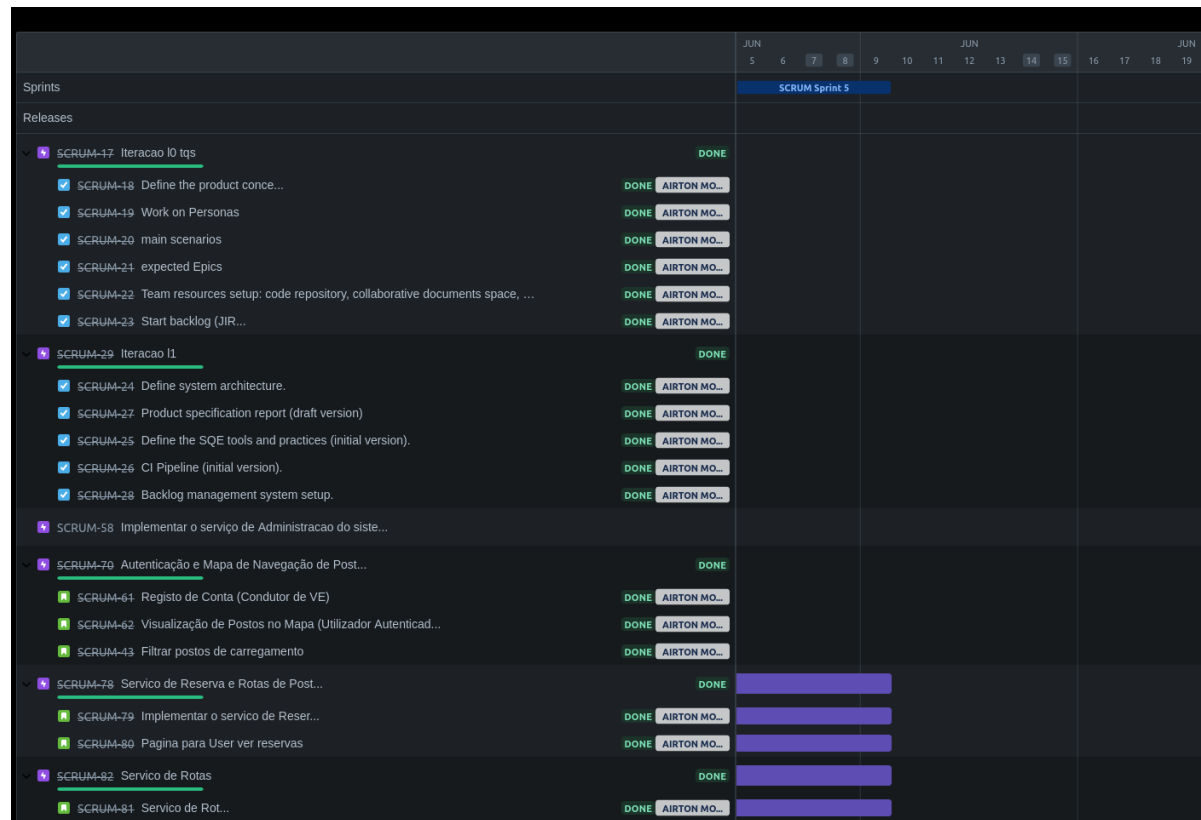
Estações Ativas
17 

Fora de Operação
2 

Total Reservas
8 

Product development roadmap

- Definição dos Epicos e User Stories
- Criação de Branches
- Desenvolvimento e Testes
- Pull Request para a Branch Principal



Arquitetura

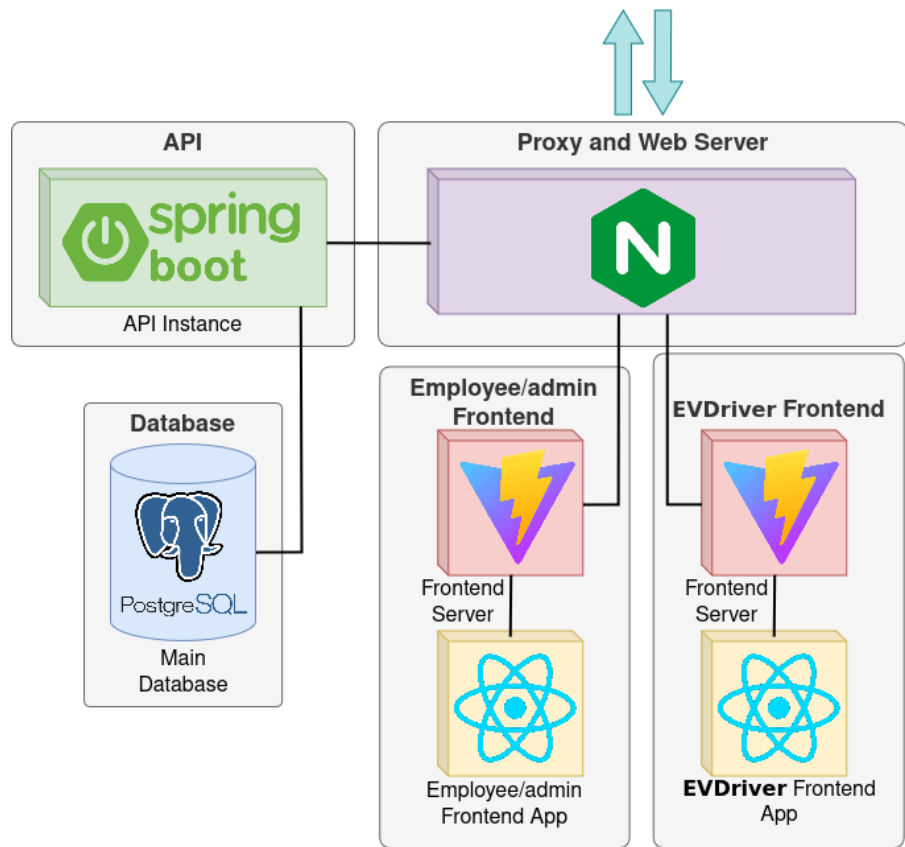
- O sistema é composto por:

- Um proxy e servidor web NGINX;

- Um backend geral;

- Um frontend orientado para o evDriver (clientes);

- Um frontend orientado o operador de sistema



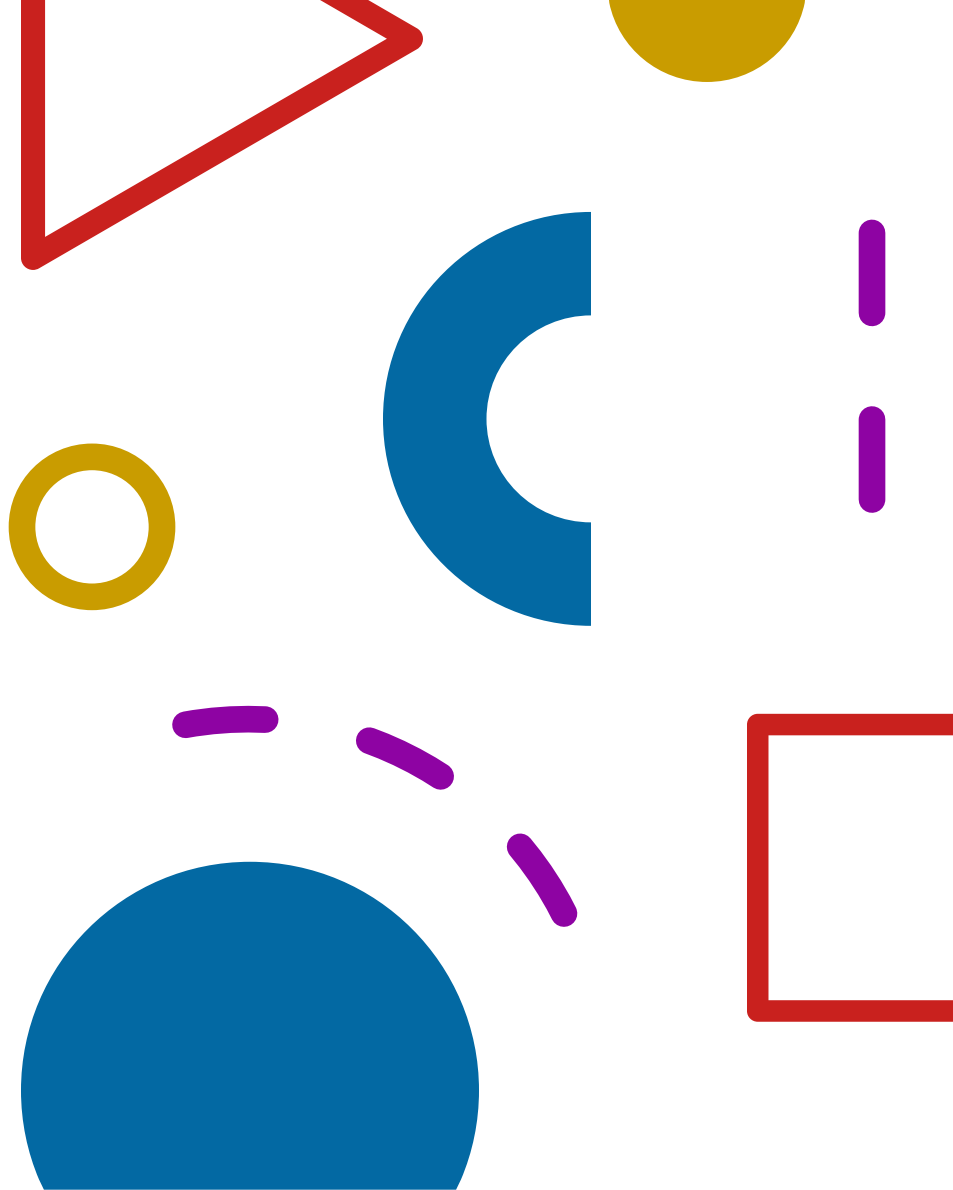
**Development
pipeline: from
stories to
production**

Dev Ops Concept



Definition of Done

- Para definir se uma user story está concluída, concordamos com as seguintes regras:
- A solução desenvolvida cumpre todos os requisitos da user story
- A solução é entregue com testes automáticos unitários, de integração
- Cobertura de testes de pelo menos 80%
- Concluído
- Código revisto por pelo menos 1 outro membro da equipa
- Sem bugs ou problemas de segurança



Testing Plan

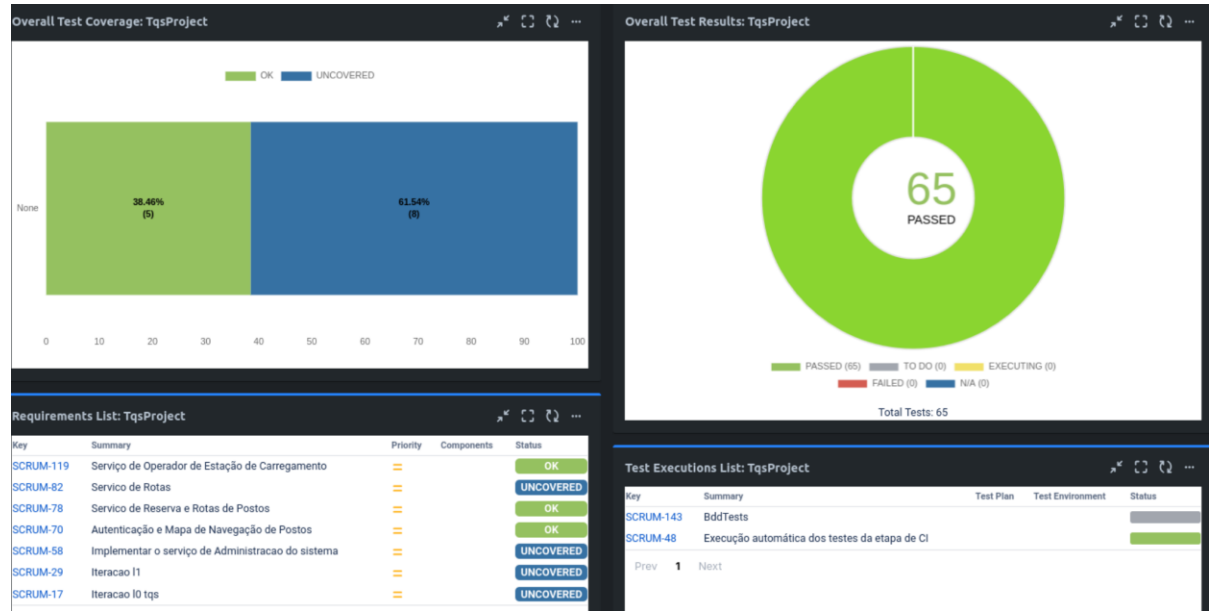
- Para os testes gerais, foram desenvolvidos testes unitários para cobrir:
- Os repositórios;
- Os serviços;
- Os controladores;
- Adicionalmente, foram criados testes para o frontend e para a aplicação completa, de forma a permitir testar os fluxos de utilizador mais comuns, garantindo assim que a integração dos sistemas e o funcionamento da aplicação frontend estão corretos.



Quality Assurance

- X-ray – Gestão Nativa de Testes para o Jira;

. Vários conjuntos de testes ligados a UserStories;



CI / CD

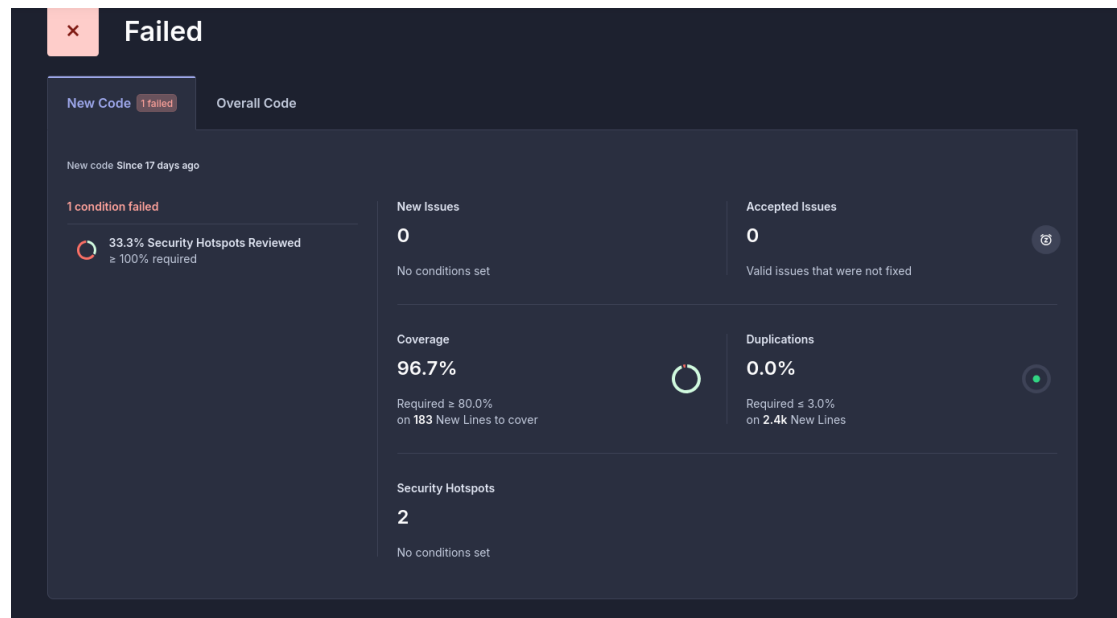
AUTOMATIC TESTING

SONARCLOUD ANALYSIS

- A cada *push* ou *merge*, os testes implementados são executados automaticamente para garantir que nenhuma alteração com erros ou problemas chegue à produção.

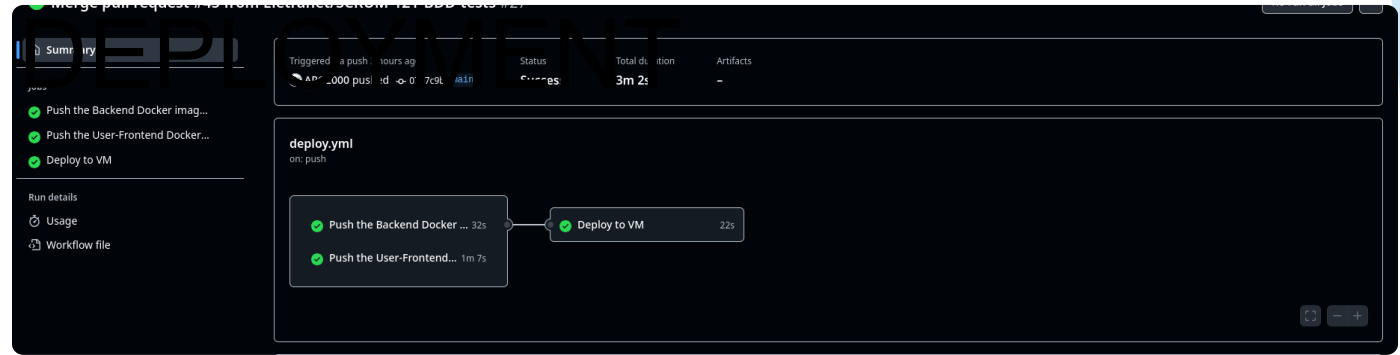
Os resultados são depois publicados no SonarCloud, um analisador estático de código e resultados de testes que facilita a detecção de alterações problemáticas no código.

A análise com o SonarCloud é realizada tanto nos componentes do frontend como do backend.



CI / CD AUTOMATIC

- Construção e envio das imagens Docker para o Docker Hub
- Runner GitHub **self-hosted** a correr numa máquina virtual
- Pull das imagens mais recentes do Docker Hub e construção das mesmas no ambiente de produção



```
deploy:
  runs-on: self-hosted
  name: "Deploy to VM"
  needs: [build-backend-image, build-user-frontend-image]

  steps:
    - name: "Checkout"
      uses: actions/checkout@v4
    - name: Create deployment (Jira)
      id: create_deployment
      uses: chrnorm/deployment-action@releases/v1
      with:
        token: ${ secrets.GITHUB_TOKEN }
        environment: production
        ref: ${ github.ref }
        description: "Deploy da versão para produção"

    - name: "Deploy"
      run: |
        echo "CREATE DATABASE tqg_backend;" > init.sql
        sudo docker compose -f docker-compose.prod.yml pull
        sudo docker compose -f docker-compose.prod.yml up -d --remove-orphans
        sudo docker container prune -f
        sudo docker image prune -af
        sudo docker builder prune -af

    - name: Update deployment status (Jira)
      uses: chrnorm/deployment-status@releases/v1
      with:
        token: ${ secrets.GITHUB_TOKEN }
        deployment_id: ${ steps.create_deployment.outputs.deployment_id }
        state: success
```



DEMO

