



World of Toilets

Sprint Report 2

Outubro/Novembro (27 - 02)

Curso: Engenharia Informática

Semestre: 2025/2026 - Quinto Semestre

Repositório: <https://github.com/nycocado/worldoftoilets>

Nycolas Souza - 20230989

Luan Ribeiro - 20230692

Lohanne Guedes - 20220085

Kira Sousa - 20231205

Data de publicação: 02/11/2025

Data de revisão: 04/11/2025

1. Definição dos Papéis:

Product Owner:

Luan Ribeiro

Scrum Master:

Nycolas Souza

Desenvolvedores:

Nycolas Souza

Luan Ribeiro

Lohanne Guedes

Kira Sousa

2. Objetivos da Sprint:

Completar a infraestrutura de base de dados com todos os scripts necessários, finalizar a arquitetura de sistemas distribuídos com NGINX e replicação de serviços, implementar o sistema de autenticação JWT na API, e finalizar o processamento do grafo de Portugal para a componente de IA.

3. Itens Concluídos:

- **Scripts completos da base de dados**
 - Script de criação (`01-create.sql`) com tabelas, constraints, foreign keys e relacionamentos
 - Índices otimizados em colunas de pesquisa frequente (`coordinates`, `user_id`, `toilet_id`)
 - Script de inserção (`02-insert.sql`) com dados realistas.
 - **Story Points:** 14.
 - **Responsável:** Nycolas, Lohanne.
- **Sistemas Distribuídos - arquitetura completa**
 - NGINX configurado como load balancer com algoritmo round-robin.
 - Health checks a cada 10 segundos nos endpoints `/health` das APIs, servidores WEBs e bases de dados.
 - 2 instâncias da API NestJS.
 - 2 instâncias do servidor web NextJS.
 - 3 instâncias da base de dados MariaDB com Galera Cluster (desistimos do MaxScale).
 - 1 instância de MinIO e Mailhog.
 - Failover automático quando uma instância falha.
 - Testes de tolerância a falhas: derrubamos 2 bases de dados, sistema manteve-se operacional.
 - **Story Points:** 18.
 - **Responsável:** Nycolas, Luan.

- **Sistema de Autenticação JWT**

- Endpoint POST /auth/login retornando access token e refresh token.
- Endpoints POST de logout, para revogação de refresh tokens.
- Middleware de autenticação @UseGuards(JwtAuthGuard) implementado.
- Tokens com expiração: access token 5m, refresh token 30 dias.
- Testes manuais com Postman.
- **Story Points:** 8.
- **Responsável:** Nycolas.

- **Processamento do grafo de Portugal (OSM)**

- Download dos dados OSM de Lisboa (optamos por Lisboa por ser um grafo menor, o de Portugal era insustentável).
- Processamento com biblioteca OSMnx (Python).
- Visualização do grafo em mapa, com um simples algoritmo de rota para teste (ainda precisa ser implementado o algoritmo A*).
- **Story Points:** 8.
- **Responsável:** Luan.

- **Integração NestJS com MariaDB**

- MikroORM configurado e todas as entidades mapeadas.
- Relações entre entidades funcionais.
- **Story Points:** 4.
- **Responsável:** Nycolas.

4. Itens em Progresso:

- **Implementação do algoritmo A***
 - Estrutura base do algoritmo iniciada em Python.
 - Função heurística (Haversine) em desenvolvimento.
 - **Progresso:** 40% - falta integração completa com o grafo e testes.
 - **Responsável:** Luan.
- **Protótipo mid-fidelity da dashboard**
 - Evolução dos wireframes para protótipo mais detalhado.
 - Desenvolvimento no figma para implementação em código.
 - **Progresso:** 50% - houveram problemas pessoais que atrasaram o desenvolvimento.
 - **Responsável:** Kira.

5. Impedimentos / Bloqueios:

- **Atraso no protótipo**
 - Por motivos pessoais, o protótipo mid-fidelity da dashboard foi interrompido.
 - Impacto: pode atrasar o desenvolvimento web.

5. Planeamento para a Próxima Semana:

- **Sprint 3 (03/11 - 09/11) - Finalizar implementação do algoritmo A***
 - Função heurística completa e testada.
 - Integração total com o grafo OSM.
 - Testes com múltiplos cenários: rotas curtas, longas, sem caminho.
 - Métricas preliminares: tempo de execução, nós explorados.
 - **Responsável:** Lohanne.
- **Microserviço Flask para IA**
 - API REST com endpoint POST /route recebendo origem e destino.
 - **Retorno:** lista de coordenadas do caminho, distância total, tempo estimado.
 - Containerização com Docker.
 - Integração com docker-compose do projeto.
 - Responsável: Luan.
- **Implementar autenticação JWT na API**
 - Endpoints de registo e login.
 - Geração e validação de tokens JWT.
 - Middleware de autenticação.
 - Hash de passwords com bcrypt.
 - **Responsável:** Nycolas.

- **Implementação dos endpoints de casas de banho e comentários na API**
 - Endpoints focados na procura de casas de banho, por boxes e proximidade, além de lazy loading.
 - Validação de permissões por cargos.
 - **Responsável:** Nycolas.
- **Protótipo da dashboard em Figma e parte inicial em código**
 - Protótipo desenvolvido em figma com componentes e diferentes cargos.
 - Desenvolvimento do código em paralelo, para ter interação com usuário na apresentação.
 - **Responsável:** Kira.

6. Métricas da Sprint:

- **Velocidade (story points):** 54 pontos concluídos.
- **Story points planejados:** 70 pontos.
- **% de tarefas concluídas:** 90%.
- **Burn-down:** Dentro do esperado, com aceleração no meio da sprint.

7. Observações e Melhorias:

- **O que correu bem:**

- **Excelente produtividade da equipa** - concluímos quase todas as tarefas planejadas
- Base de dados ficou muito bem estruturada, com dados de teste realistas
- Sistema distribuído funcionando perfeitamente, testes de tolerância a falhas bem-sucedidos
- Integração entre Luan e Nycolas na parte Docker foi fluída

- **O que pode melhorar:**

- **Documentação do código ainda insuficiente** - precisamos de mais comentários e documentação de APIs
- Alguns commits muito grandes - devemos fazer commits mais atómicos
- Testes automatizados ainda não iniciados - dependemos de testes manuais

- **Ações para a próxima sprint:**

- Cada desenvolvedor deve documentar suas APIs com comentários (padronização dependendo da linguagem de programação)
- Iniciar testes unitários na Sprint 4 (pelo menos para módulo de autenticação)