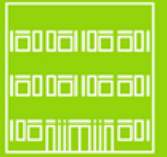




AAUAV



universidade
de aveiro



deti

Taça UA

Apresentação Final

Bernardo Borges 103592
Alexandre Regalado 124574
Mateus Rocha 122949
Diego Aguilar 117122
Gonçalo Simões 119412



Índice

- Contexto
- Arquitetura
- Metodologia
- Qualidade & Testes
- Live Demo
- Obstáculos Encontrados
- Revisão de Metas



Contexto

**A Taça UA é uma das maiores
competições desportiva informal
intra-universitária em Portugal.**

Contexto - Problema Atual

Uso de Excel

- Uso de Excel para guardar e manipular informação - principalmente resultados

Informação Descentralizada

- Informação separada em varios ficheiros, com propagação dependente de redes sociais

Utilização Complexa e Informal

- O trabalho administradores gerais e de núcleo fica fragmentado e feito a partir de comunicação informal

Contexto - Estado de Arte

Critério	FADU	Taça UA	Taça UA (Projeto)
Gestão de torneios / modalidades	SIM	PARCIAL	SIM
Gestão por Núcleos	NÃO	GESTÃO INFORMAL	SIM
Hierarquia Administrativa	NÃO	GESTÃO INFORMAL	SIM
Agendamento de Jogos	SIM	MANUAL	SIM
Registo de resultados	SIM	REDES SOCIAIS	SIM
Pontuação automática por modalidade	NÃO	EXCEL	SIM

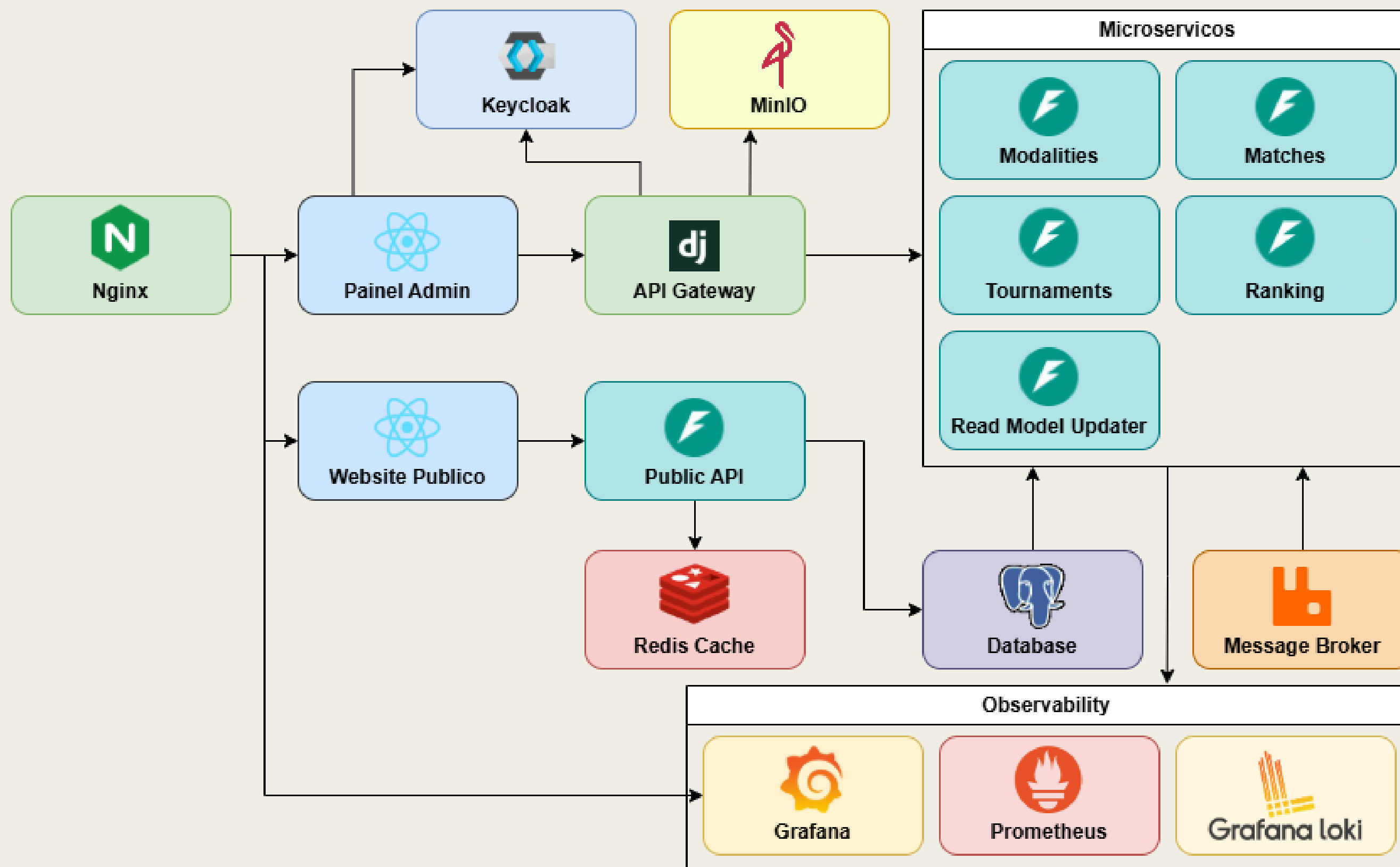
Contexto - Estado de Arte

Critério	FADU	Taça UA	Taça UA (Projeto)
Classificação geral intermodalidades	NÃO	EXCEL	SIM
Inscrição de atletas	SIM	FORMS	SIM
Validação de sócios	NÃO	MANUAL	SIM
Fichas de Jogo (PDF)	NÃO	MANUAL	SIM
Histórico multi-época	SIM	EXCEL	SIM
Calendário público com filtros	SIM	NÃO	SIM



Arquitetura

Arquitetura





Metodologia

Metodologia - Organização

Organização do Projeto

- Separação do projeto em 6 sprints. Foco em metodologia ágil para desenvolvimento

Revisão de código

- Revisões de código via pull requests antes de integrar código - obrigatório em pulls para dev e main

Controlo de commits

- Uso de pre-commit hooks para garantir a qualidade automática do código

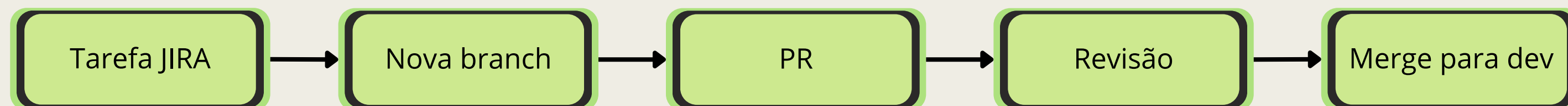
Metodologia - Gestão de Tarefas

Feramentas

- Jira para organização de tarefas. Branches são criadas no GitHub pelas tasks, com nomes gerados a partir do nome da task

Branches

- Branches geradas a partir da branch 'dev', onde eventualmente converge as mudanças





Qualidade & Testes



Qualidade & Testes - Usabilidade

Tipo de Teste

- Questionário SUS (System Usability Scale) aplicado a utilizadores reais focado na parte administrativa

Resultados

- Pontuação média de 80.7/100 - acima do limiar estabelecido de 68

Melhorias Feitas

- Uso de 'Toast Notifications' para mostrar erros, uso de botões de ajuda e aria labels para ajudar o utilizador

Qualidade & Testes - Performance

● Tipo de Teste

Uso de K6 + monitorização com Grafana/Loki

Resultados Foco no Público

● Testes focados no público devido à carga elevada esperada de pedidos

Melhorias Feitas

● Introdução de cache (Redis) e de uma pool de pedidos maior na base de dados

Qualidade & Testes - Integração

Camada	Framework	Foco do Teste
Microserviços (backend)	pytest	Rotas CRUD, comportamento de endpoints
Painel Admin (frontend)	Vitest	Fluxos de UI com mock de API
Website Público (frontend)	Vitest	Paginas de consulta pública

Tipo de Teste

- Teste entre componentes, não apenas unidade isoladas. Backend usa base de dados em memória nos testes.



Live Demo



Obstáculos Encontrados

Obstáculos Encontrados

Integração com Microserviços

- Dificuldades com comunicação assíncrona e propagação de erros. *Criação de logs centralizados com o Loki*

Falta de Dados Reais

- Falta de uma fonte boa de dados reais. *Criação de dados a partir de especulação para testes*

Problemas de Performance

- A partir dos testes de performance foi possível identificar um bottleneck. *Criação de cache no lado público*

Obstáculos Encontrados

Routing do NGINX

- Problemas com o routing do Nginx no ambiente de staging na VM.

VM de Staging Atrasada

- Entrega da Virtual Machine de staging adiada devido a mudanças internas.

Problemas de Orientação no Código

- Problemas com o código antigo, causando atrasos e problemas desnecessários. *Sessão de refactor do código*



Revisão de Metas



Revisão de Metas - O que Foi Feito

Requisitos

- Foram levantados o total de 31 requisitos funcionais no início do ano letivo. Todos foram confirmados como feitos

Testes

- Cobertura ampla de testes de integração e de performance

Manutenção

- O projeto está pronto para manutenção e eventuais mudanças devido à documentação dos métodos do código

Revisão de Metas - O que Falta

Deploy em Produção

- Só temos ambiente de staging.
Precisamos de confirmar o deploy para produção com a AAUAv

Legalidades

- Questões de tomada de posse do projeto dos estudantes para a AAUAv não foram formalizadas

Features Extra

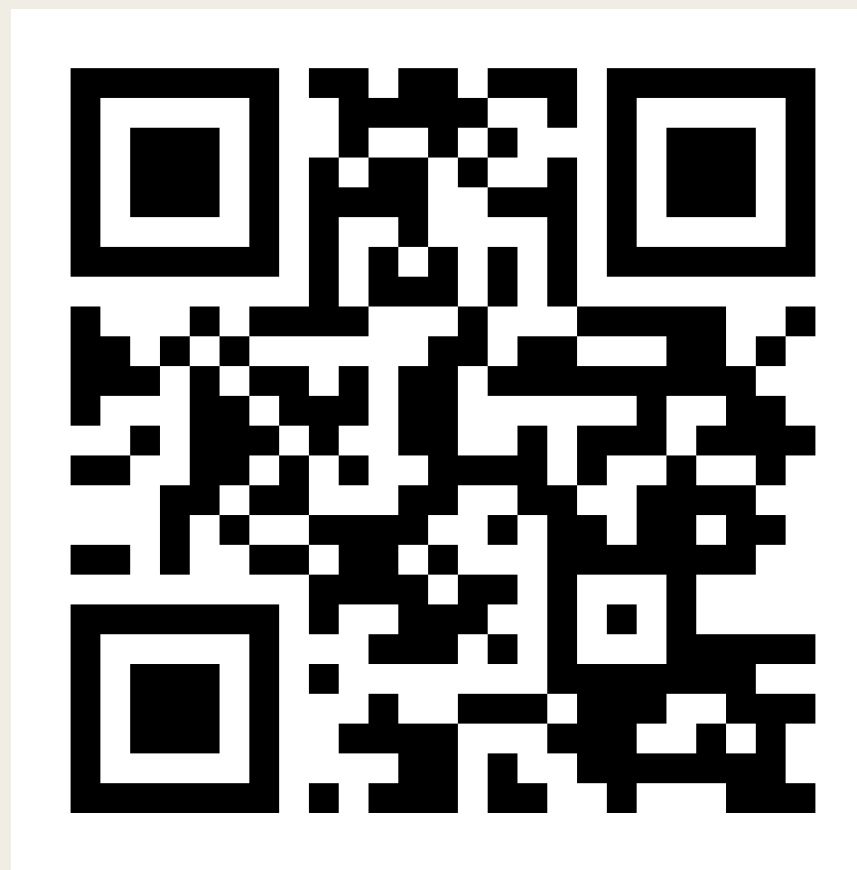
- Certas features como uma dashboard “to-do” para administradores de núcleo com tarefas pendentes

Links importantes

Microsite



Organização



Jira



**Obrigado pela
atenção**