



Sprint 5 (Final)

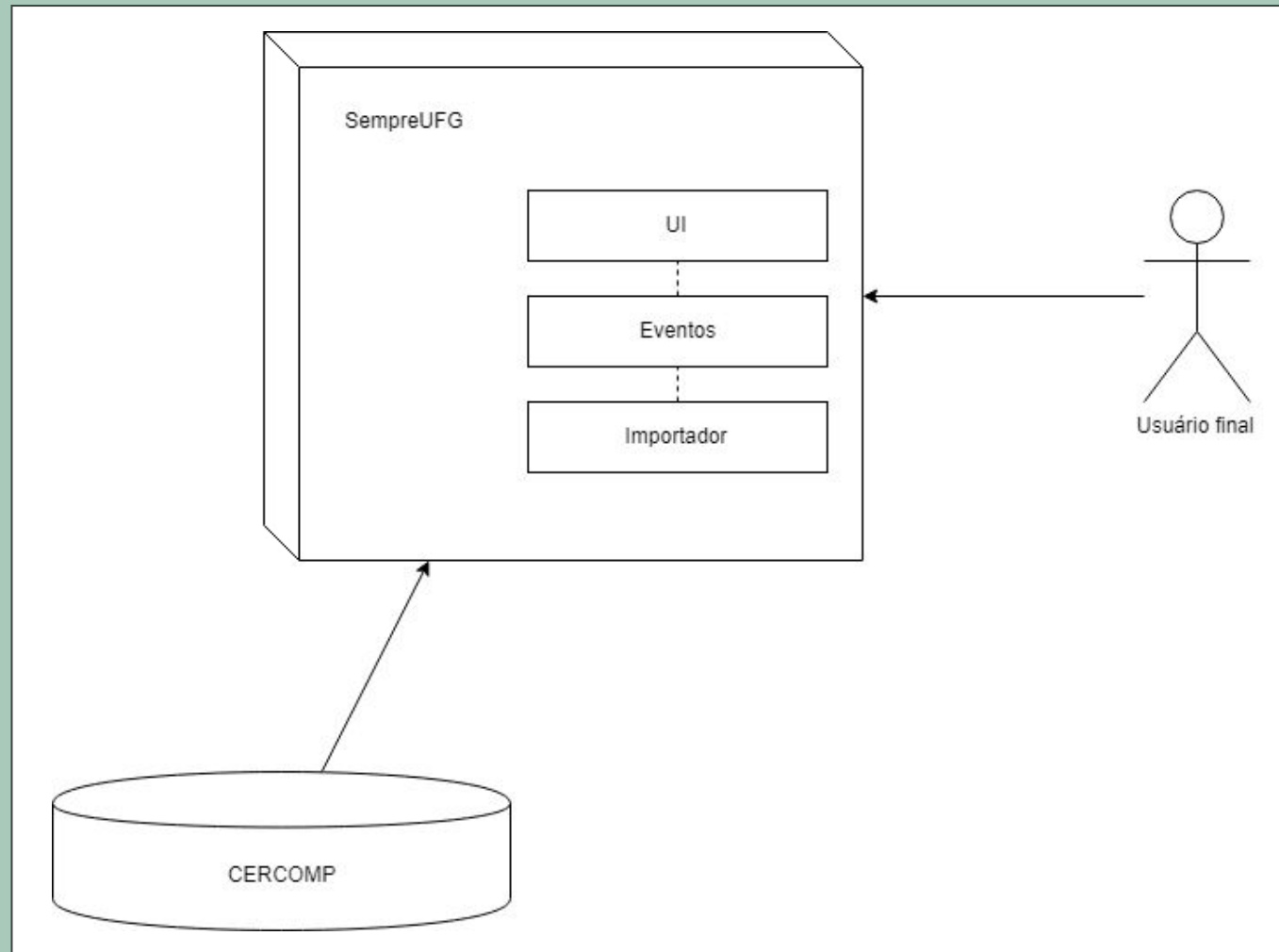
AGENDA

- Visão Geral da Arquitetura
- Linha do tempo
- Viewpoints e Views
- Atributos de Qualidade
- Priorização de atributos na prática
- Checklist de avaliação
- Arquitetura implementada

Legenda: Aa Não modificado
Aa Refatorado
Aa Novo



VISÃO GERAL



LINHA DO TEMPO DO PROJETO

VIEWPOINTS

- Viewpoint Informação
- Viewpoint Desenvolvedor
- Viewpoint Implantador
- Viewpoint Projetista
- Viewpoint Segurança
- Viewpoint UI
- Viewpoint Usuário
- Viewpoint Usuário Final





VISÕES

- Visão de Dados
- Visão da Segurança
- Visão Física
- Visão Lógica
- Visão do Desenvolvimento
- Visão de Requisitos
- Visão Front-end
- Visão do Processo

Atributos de Qualidade

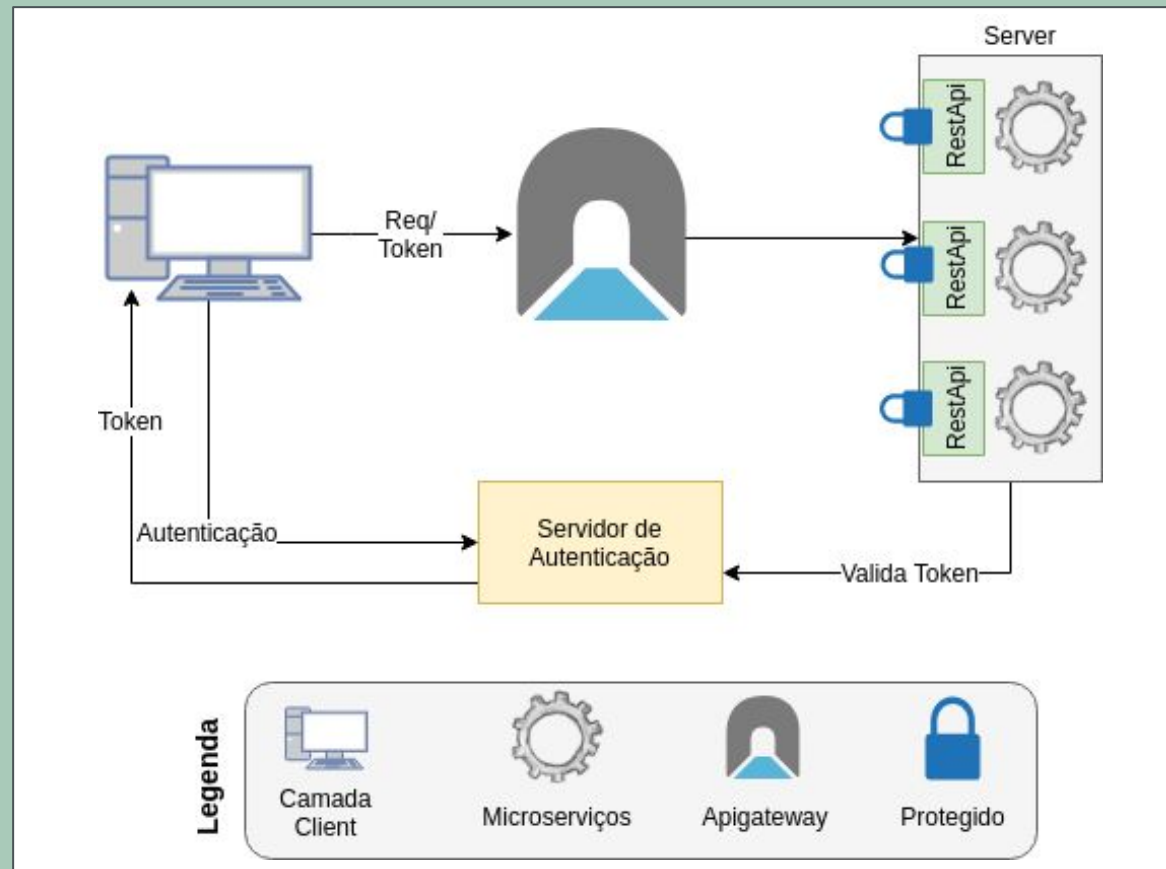
- Segurança
- Portabilidade
- Confiabilidade
- Manutenibilidade



Como os atributos foram
priorizados?



Segurança



- [DSUFG04] Utilização do protocolo OAuth2
- [DSUFG06] Orquestração de microserviços é feita pela api gateway

Segurança



[DSUFG06] Orquestração de microserviços é feita pela api gateway

- Servidores não ficam expostos
- Tratamento de ataques
- Integridade de dados

Portabilidade

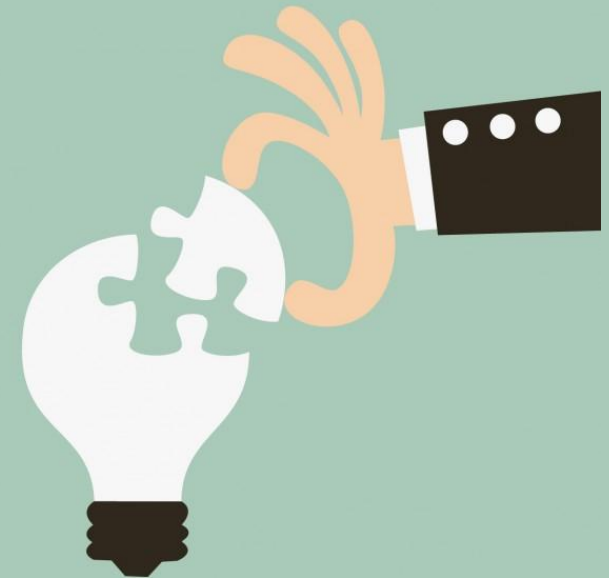
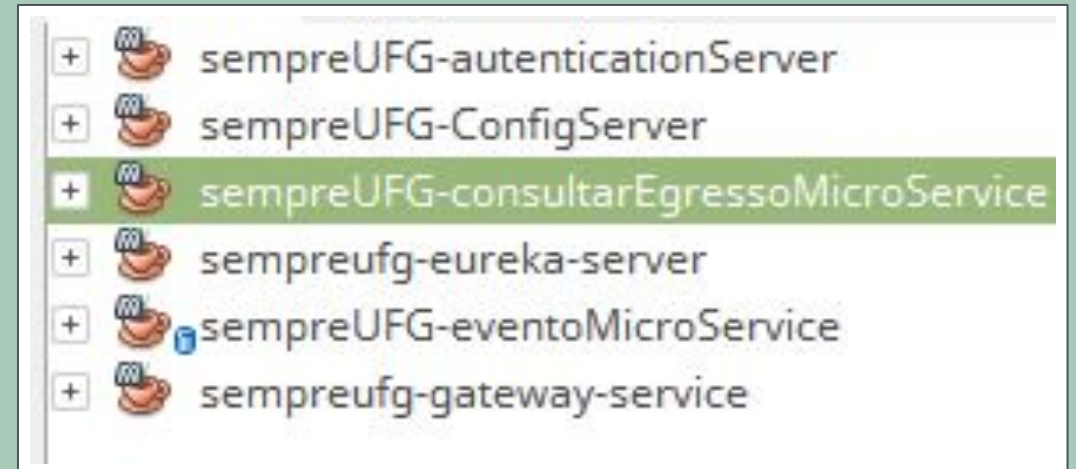


- **[DSUFG02]** Camada “Server” dividida em microserviços
- **[DSUFG05]** Microserviços expõem dados através de apis restfull
- **[DSUFG07]** Microserviços são construídos em Java

Portabilidade

[DSUFG02] Camada “Server” dividida em microsserviços

- Cada microsserviço é um projeto independente
- Eles podem ser portados para outros ambientes



Portabilidade



[DSUFG05] Microserviços expõem dados através de apis restfull

- Basta ter acesso ao endpoint;
- Independe de tecnologia;
- Os dados trafegados estão em JSON;

```
{  
  "access_token": "93e60ada-e494-4ac3-a77e-4bb3c9693c95",  
  "token_type": "bearer",  
  "refresh_token": "cc7655f6-d02e-4084-a366-f83d2689ffc7",  
  "expires_in": 1799,  
  "scope": "read write"  
}
```

Confiabilidade

- [DSUFG02] Camada “Server” dividida em microserviços
- [DSUFG09] Cada microserviço terá seu próprio banco de dados



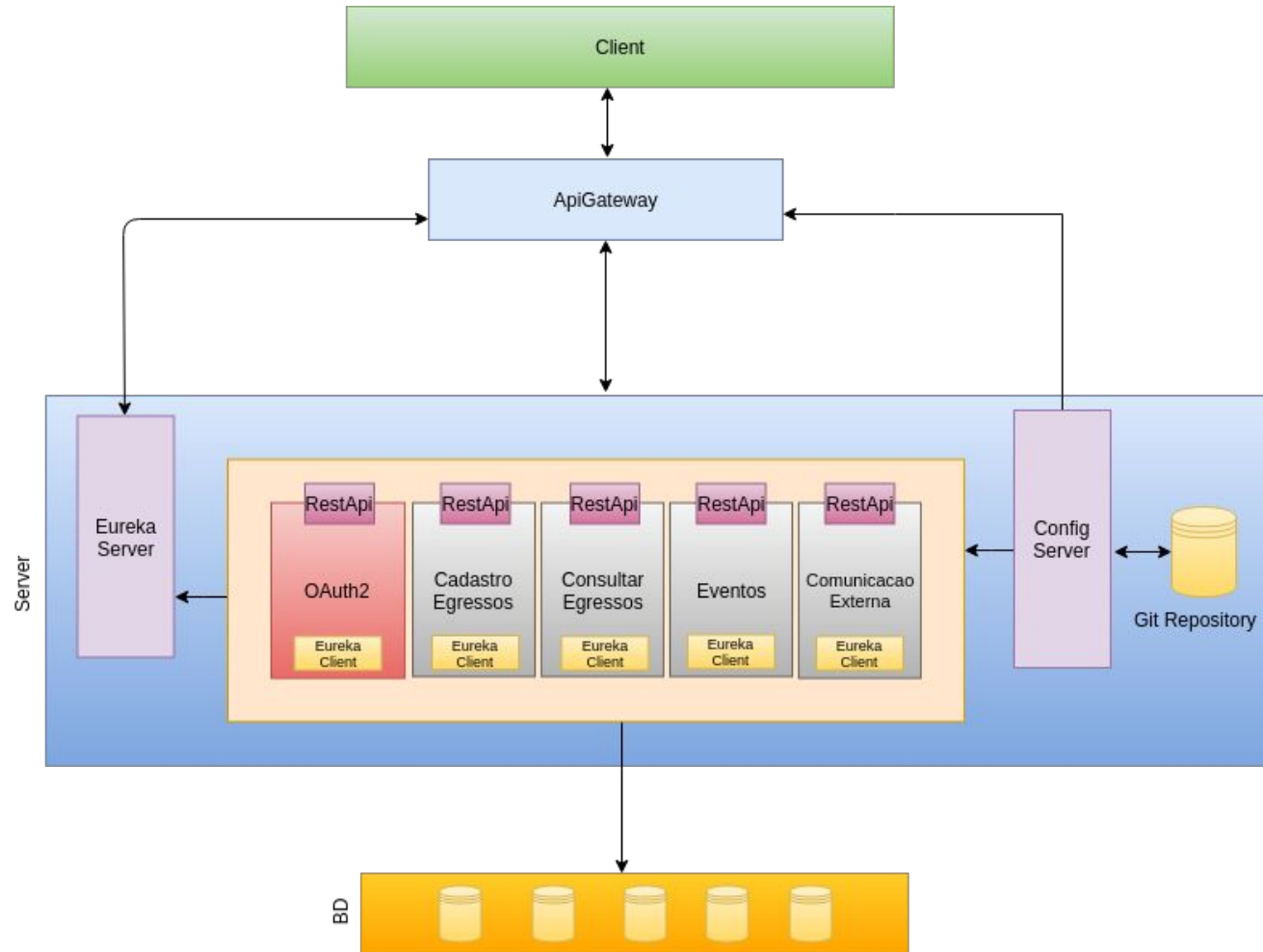
```
List of databases
+-----+-----+-----+-----+-----+-----+
| Name | Owner | Encoding | Collate | Ctype | Access privileges |
+-----+-----+-----+-----+-----+-----+
| postgres | postgres | UTF8 | pt_BR.UTF-8 | pt_BR.UTF-8 | postgres=CTc/postgres |
| sempreufgaluno | postgres | UTF8 | pt_BR.UTF-8 | pt_BR.UTF-8 | postgres=CTc/postgres |
| sempreufevento | postgres | UTF8 | pt_BR.UTF-8 | pt_BR.UTF-8 | postgres=CTc/postgres |
| template0 | postgres | UTF8 | pt_BR.UTF-8 | pt_BR.UTF-8 | postgres=CTc/postgres |
| template1 | postgres | UTF8 | pt_BR.UTF-8 | pt_BR.UTF-8 | postgres=CTc/postgres |
| teste | postgres | UTF8 | pt_BR.UTF-8 | pt_BR.UTF-8 | postgres=CTc/postgres |
+-----+-----+-----+-----+-----+-----+
[saulo-linux@sauro-PC ~]$
```

Manutenibilidade

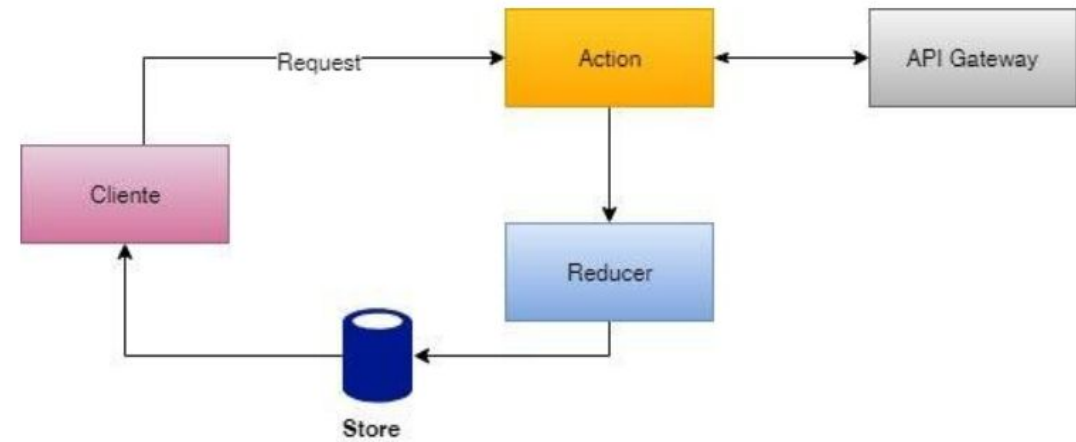
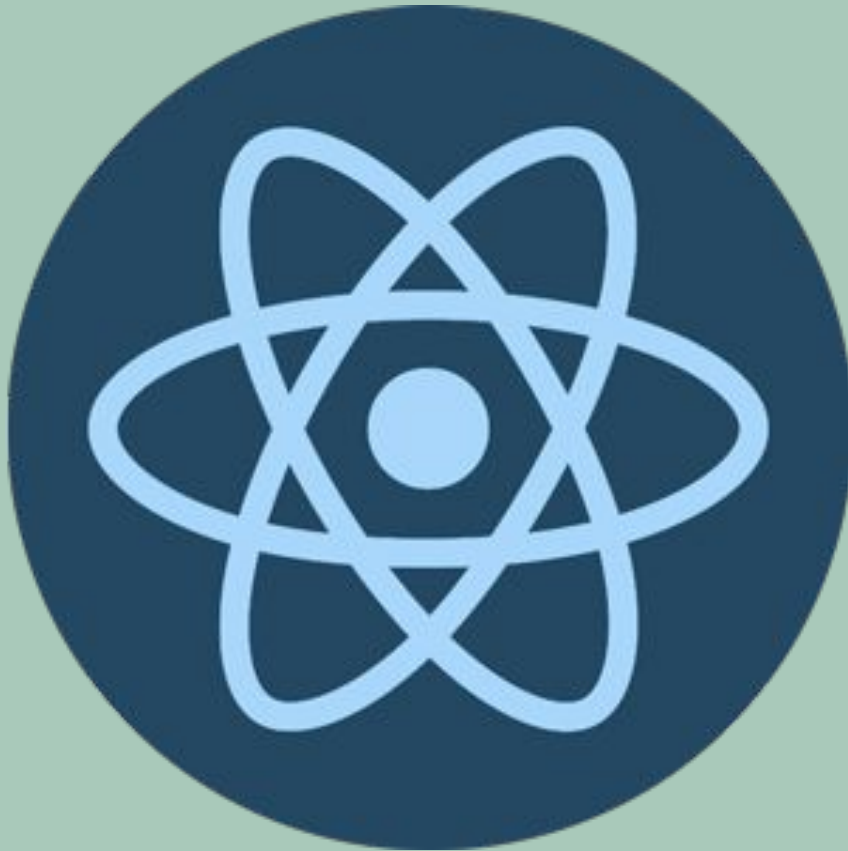


- [DSUFG01] Divisão da arquitetura em camadas
- [DSUFG02] Camada “Server” dividida em microserviços
- [DSUFG03] Utilização de *React* e *Redux* na camada “Client”
- [DSUFG06] Orquestração de microserviços é feita pela api gateway
- [DSUFG08] A gerência dos bancos de dados é feita pelo PostgreSQL
- [DSUFG09] Cada microserviço terá seu próprio banco de dados
- [DSUFG10] Os arquivos de configuração ficarão centralizados

[DSUFG01] Divisão da arquitetura em camadas



[DSUFG03] Utilização de *React* e *Redux* na camada “Client”










- Facilita criação de UI
- Favorece o reuso;
- Favorece a modularidade;
- Fluxo de dados previsível;

[DSUFG10] Os arquivos de configuração ficarão centralizados

14 lines (12 sloc) | 275 Bytes

Raw Blame History

```
1 server:
2   port: 9091
3
4 eureka:
5   instance:
6     hostname: localhost
7   client:
8     registerWithEureka: true
9     fetchRegistry: false
10    serviceUrl:
11      defaultZone: http://${eureka.instance.hostname}:${server.port}/eureka/
12  server:
13    wait-time-in-ms-when-sync-empty: 3000
```

 saullocalixto	Delete sample-config-app.yml	Latest commit 100b2d0 just now
 README.md	Initial commit	a month ago
 sempreufg-apigateway.yml	Update sempreufg-apigateway.yml	23 hours ago
 sempreufg-auth-server.yml	Create sempreufg-auth-server.yml	a month ago
 sempreufg-consultar-egresso-servi...	Rename sempreufg-aluno-service.yml to sempreufg-consultar-egresso-ser...	10 hours ago
 sempreufg-eureka-server.yml	Rename sempreufg-eurekaserver.yml to sempreufg-eureka-server.yml	a month ago
 sempreufg-evento-service.yml	Update sempreufg-evento-service.yml	10 hours ago

DÚVIDAS?



REFERÊNCIAS

- ISO/IEC/IEEE 42010, 2011, Systems and software engineering —Architecture description
- Documento de requisitos do software SempreUFG, 2016
- Guilherme Germoglio, 2010, Arquitetura de Software

Autores da apresentação:
Gustavo Batista, Murillo Nunes, Saulo Calixto