**SMART ENERGY**



**Integração de Sistemas de Informação**

**Licenciatura em Engenharia de Sistemas Informáticos**

**Regime Pós-Laboral**

**2022/2023**

**Alunos**

Francisco Pereira – Nº 21156

Tiago Azevedo - Nº 21153

**Orientação**

Profº Óscar Ribeiro

Conteúdo

[Arquitetura Solução 3](#_Toc122243779)

[Serviços Frontend 3](#_Toc122243780)

[Serviço: Criar Utilizador 3](#_Toc122243781)

[Serviço: Ver Utilizadores 4](#_Toc122243782)

[Serviço: Autenticação Normal Frontend 4](#_Toc122243783)

[Serviço: Autenticação Google-Auth Frontend 4](#_Toc122243784)

[Serviço: Ver equipamentos 4](#_Toc122243785)

[Serviço: Ver Eventos equipamento 4](#_Toc122243786)

[Serviços Backend 5](#_Toc122243787)

[Serviço: Autenticação Poste 5](#_Toc122243788)

[Serviço: Adicionar Evento 5](#_Toc122243789)

[Estratégia de Implementação 5](#_Toc122243790)

[Bibliografia 7](#_Toc122243791)

# Arquitetura Solução

Esta secção documenta todas as decisões arquiteturais do projeto SmartEnergy. Este, irá ser composto maioritariamente por três camadas distintas:

Figura : Camadas Lógicas do sistema SmartEnergy

No contexto da disciplina de Integração de Sistemas de Informação, a camada que será descrita neste relatório é a camada de Backend. Para mais informação acerca do projeto aconselha-se a leitura do documento [1].

A camada de Integração tem como objetivo receber e guardar informação enviado pelos postes e providenciar esta informação organizada para a aplicação Mobile (frontend). A aplicação mobile terá a responsabilidade de providenciar uma interface simples e intuitiva aos utilizadores finais.

A camada de Backend tem a responsabilidade de providenciar os seguintes serviços/funcionalidades para o Frontend e Backend, como apresentado na .

Figura : Funcionalidades Camada de Integração

## Serviços Frontend

### Serviço: Criar Utilizador

|  |  |
| --- | --- |
| Nome do Serviço | Criar Utilizador |
| Objetivo | Permitir o Frontend criar um novo utilizador |
| Parâmetros | Username/Email: String  Password: String  Tipo: <Utilizador, Administrador> |
| Retorno | Resultado Operação: Boolean |
| Caminho | /utilizador/criar |
| Pré-condições | Utilizador com login feito e administrador |

### Serviço: Ver Utilizadores

|  |  |
| --- | --- |
| Nome do Serviço | Ver utilizadores |
| Objetivo | Permitir o Frontend Listar Utilizadores |
| Parâmetros |  |
| Retorno | Resultado Operação: Boolean  Lista  Username/Email: String |
| Caminho | /utilizador/listar |
| Pré-condições | Utilizador com login feito e administrador |

### Serviço: Autenticação Normal Frontend

|  |  |
| --- | --- |
| Nome do Serviço | Autenticação |
| Objetivo | Permitir o Frontend autenticar utilizadores |
| Parâmetros | Utilizador: String  Password: String |
| Retorno | Resultado Operação: Boolean |
| Caminho | /autenticacao/normal |
| Pré-condições | Utilizador tem de estar registado no Smart Energy |

### Serviço: Autenticação Google-Auth Frontend

|  |  |
| --- | --- |
| Nome do Serviço | Autenticação |
| Objetivo | Permitir o Frontend autenticar utilizadores utilizando conta google |
| Parâmetros | Email: String  Password: String |
| Retorno | Resultado Operação: Boolean |
| Caminho | /autenticação/2google |
| Pré-condições | Utilizador tem de estar registado no Smart Energy |

### Serviço: Ver equipamentos

|  |  |
| --- | --- |
| Nome do Serviço | Ver Equipamentos |
| Objetivo | Permitir o Frontend receber a lista de equipamentos |
| Parâmetros | ID equipamento: String  Token login: String |
| Retorno | Resultado Operação Boolean  ID equipamento: String  Estado: Online/Offline |
| Caminho | /equipamentos/listar |
| Pré-condições | Utilizador com login feito |

### Serviço: Ver Eventos equipamento

|  |  |
| --- | --- |
| Nome do Serviço | Ver Eventos de Equipamento |
| Objetivo | Permitir o Frontend ver detalhes de equipamento |
| Parâmetros | Token login: String  ID equipamento: String |
| Retorno | Resultado Operação: Boolean  ID equipamento: String  Estado: Online/Offline  Mac Address: String  IPV4 Address: String  Localizacao: String  Lista Eventos:  Data: Date  Iluminação: Float  Estado Iluminação: Boolean  Valor Luz: Float  Movimento: Boolean |
| Caminho | /equipamentos/ver?id=<ID> |
| Pré-condições | Utilizador com login feito |

## Serviços Backend

### Serviço: Autenticação Poste

|  |  |
| --- | --- |
| Nome do Serviço | Autenticação |
| Objetivo | Permitir aos postes/equipamentos efetuar autenticação |
| Parâmetros | ID Poste: String |
| Retorno | Resultado Operação: Boolean  Token Autenticação: String |
| Caminho | /autenticação/poste |
| Pré-condições | Nenhuma |

### Serviço: Adicionar Evento

|  |  |
| --- | --- |
| Nome do Serviço | Autenticação |
| Objetivo | Permitir aos postes/equipamentos adicionar um novo evento |
| Parâmetros | ID Poste: String  Token Autenticação: String  Estado: Online/Offline  Mac Address: String  IPV4 Address: String  Localizacao: String  Data: Date  Iluminação: Float  Estado Iluminação: Boolean  Valor Luz: Float  Movimento: Boolean |
| Retorno | Resultado Operação: Boolean |
| Caminho | /poste/evento |
| Pré-condições | Poste tem que estar autenticado |

## Estratégia de Implementação

Para implementar a arquitetura descrita, foi decidido em grupo, efetuar o hospedamento na Google Cloud. A linguagem de programação utilizada será Python com a Framework Django. Será utilizado a biblioteca do django de REST para os webservices onde estarão hospedados os serviços de frontend e backend. A apresenta as ferramentas a ser utilizadas para a implementação da solução Smart Energy.

Figura : Estratégia de Implementação

## Modelo de Dados

Detalhes acerca do modelo de dados estão documentados em detalhe em [1]. A Figura 4, exportada de [1], apresenta o modelo de dados em formato de Entidade-Relação



Figura : Modelo de dados

# Implementação

Este capítulo documenta todas as decisões técnicas consideradas para o desenvolvimento do backend Smart Energy. Como referido anterimente, o projeto foi desenvolvido em Python com a Framework Django [2] e o hosting na Google Cloud [3].

O repositório do código está armazenado em dois sítios:

* Github: <https://github.com/tiagoazevedo22/LESI3_SmartCampus.git>
  + Pasta: ISI
* Google Cloud: <https://franciscorafaeldpereira@gmail.com@source.developers.google.com:2022/p/smartenergy-backend/r/smartenergy-backend>

## Modelo

### Base de dados

A estrutura da base de dados encontra-se apresentada em anexo na Figura 5, onde estão identificadas a vermelho as tabelas apresentadas na página 6, na secção Modelo de Dados. Através da utilização dos mecanismos do Django, foi possível criar várias tabelas de gestão interna do backend. Este facto, justifica o porquê de haver tantas tabelas, no entanto, esta característica permite reutilizar e desenvolver de forma segura e rápida novas aplicações.

Algumas considerações:

1. O Utilizador criado é o disponibilizado por defeito no Django, onde, é possível gerir e associar grupos e assim, permissões. Esta característica será usada para atribuir dispositivos a grupos, e desta forma, cumprir o requisito de gestão de permissões.
2. Um Utilizador pode fazer parte de um ou mais grupos
3. Um dispositivo está associado a um único grupo
4. A base dados foi criada pelo Django através do processamento dos modelos [2], que são apresentados na secção anterior.

## Vistas

# Google Cloud

# Bibliografia

|  |  |
| --- | --- |
| [1] | G. 6, “Relatório de Projeto Aplicado: Smart Energy,” IPCA, Barcelos, 2022. |
| [2] | Django Rest Framework, “Django Rest Framework,” [Online]. Available: https://www.django-rest-framework.org/#quickstart. [Acedido em 3 1 2023]. |
| [3] | google, “Django on Cloud Run,” [Online]. Available: https://codelabs.developers.google.com/codelabs/cloud-run-django. [Acedido em 3 1 2023]. |
| [4] | Django, “Django Project,” [Online]. Available: https://www.djangoproject.com/. |

# Anexos

## Diagrama Entidade Relação – Django

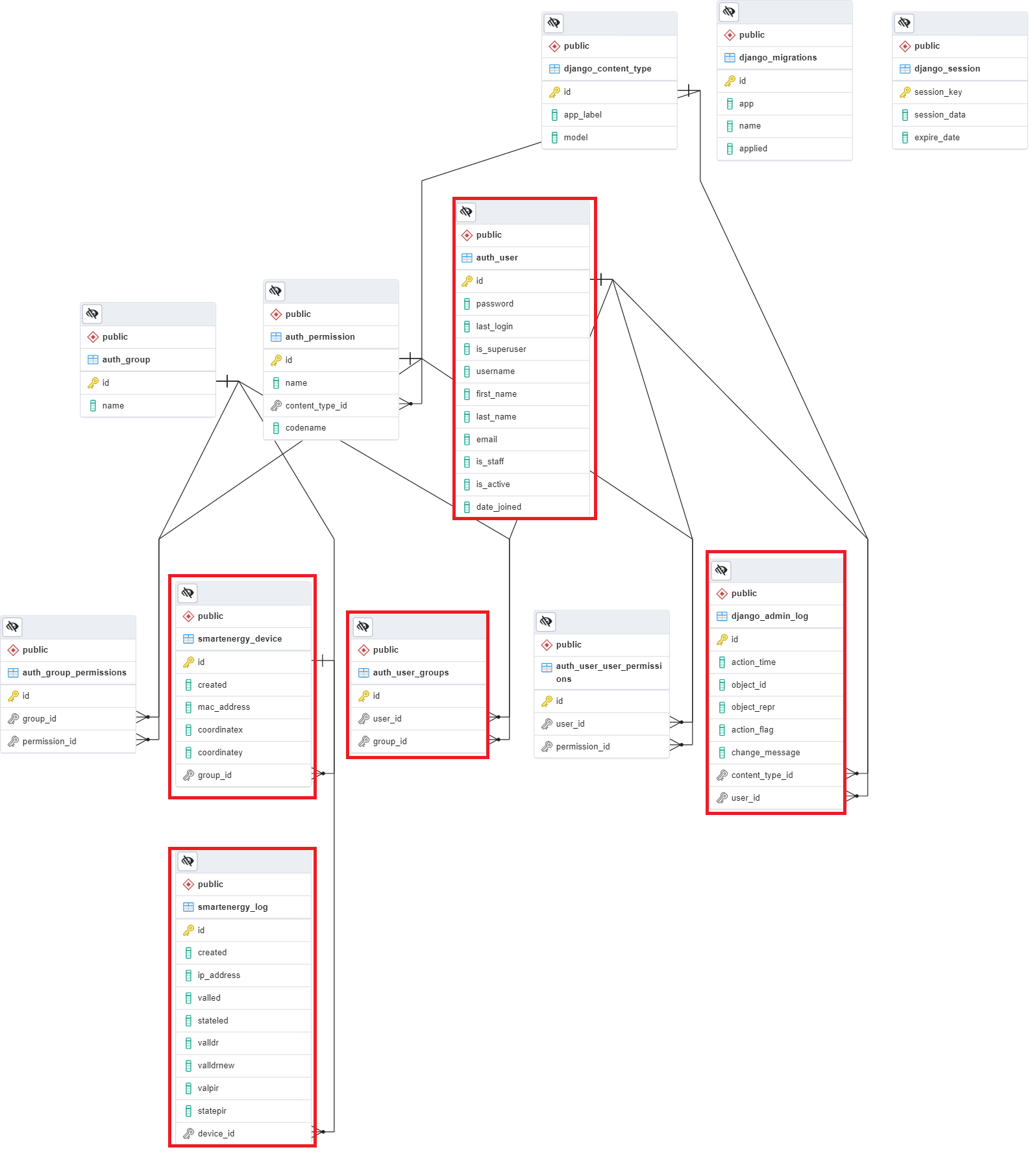


Figura : ERD Django