

ISPGAYA

instituto superior politécnico

Escola Superior de Ciência e Tecnologia



Licenciatura em Engenharia Informática

2022/2023

SoftSolutions

Alunos:

André Silva (3931)

Filipe Sousa (2020102538)

Inálio Gomes (4238)

João Moita (2020100752)

Joaquim Ferreira (1804)

Pedro Briga (2020102431)

Sérgio Teixeira (2020102403)

Docente: Prof. Dr. Fernando Almeida

18 de dezembro de 2022

ÍNDICE DE CONTEÚDOS

ÍNDICE DE IMAGENS	3
ÍNDICE DE TABELAS.....	3
ABREVIATURAS.....	4
1. INTRODUÇÃO	5
1.1. ENQUADRAMENTO	5
1.2. OBJETIVOS	6
1.3. ANÁLISE DOS RISCOS DO PROJETO.....	6
1.4. PLANO DE CONTINGÊNCIAS	7
1.4.1. Desvio do Escopo	7
1.4.2. Prazos Curtos.....	7
1.4.3. Excesso Carga de Trabalho	7
1.4.4. Falta de Comunicação	7
1.4.5. Falta de Conhecimento	7
1.4.6. Ausência de um membro do grupo.....	7
1.4.7. Desistência de um membro do grupo	7
2. DESCRIÇÃO DO PRODUTO	9
2.1. CONTEXTO DO PRODUTO.....	9
2.2. FUNCIONALIDADES DO PRODUTO	9
2.3. RESTRIÇÕES E DEPENDÊNCIAS.....	10
3. REQUISITOS	11
3.1. ATORES.....	11
3.1.1. Administrador de sistema.....	11
3.1.2. Analista de projetos	11
3.2. CASOS DE UTILIZAÇÃO.....	12
3.3. REQUISITOS FUNCIONAIS	13
3.4. DESENHO E IMPLEMENTAÇÃO.....	14
3.4.1. Diagrama de atividades	14
3.4.2. Diagrama de sequência.....	16
3.5. REQUISITOS NÃO FUNCIONAIS.....	18
4. ARQUITETURA	19
4.1. ARQUITETURA FÍSICA.....	19
4.2. ARQUITETURA LÓGICA	20
4.3. ARQUITETURA DA BASE DE DADOS	21
4.3.1. DIAGRAMA DE CLASSES	21
4.3.2. MODELO RELACIONAL	22
4.3.3. EXPLICAÇÃO DO DIAGRAMA DE CLASSES.....	22
REFERÊNCIAS BIBLIOGRÁFICAS	23

ÍNDICE DE IMAGENS

Figura 1 - Casos de Uso da Aplicação	12
Figura 2 - Diagrama de atividade de inserção de projeto.....	14
Figura 3 - Diagrama de atividade de simulação e exportação da equipa	15
Figura 4 - Diagrama de sequência do login.....	16
Figura 5 - Diagrama de estados do projeto	17
Figura 6 - Arquitetura Física	19
Figura 7 - Arquitetura Lógica.....	20
Figura 8 - Diagrama de Classes.....	21

ÍNDICE DE TABELAS

Tabela 1 - Tabela de Análise de Riscos	6
Tabela 2 - Requisitos Funcionais da Aplicação	13
Tabela 3 - Requisitos Não Funcionais da Aplicação.....	18

ABREVIATURAS

IDE – Integrated Development Environment | Ambiente de Desenvolvimento Integrado;

T-SQL – Transact-SQL;

MSSQL – Microsoft SQL Server;

SGBD – Sistema de Gestão de Base de Dados;

SSMS – SQL Server Management Studio;

CSV – Comma-separated values;

LINQ – Language Integrated Query;

NUTS – Nomenclature of Territorial Units for Statistics.

1. INTRODUÇÃO

1.1. ENQUADRAMENTO

No âmbito da unidade curricular de Sistemas de Apoio à Decisão, foi-nos proposto o desenvolvimento de um projeto para a monitorização das metas municipais de desenvolvimento sustentável.

O cumprimento da Agenda 2030 implica uma responsabilidade compartilhada na implementação dos Objetivos de Desenvolvimento Sustentável (ODS). O governo local, em particular os municípios, está mais próximo dos cidadãos e, portanto, mais bem posicionado para responder às necessidades de cada região.

A Agenda 2030 é um documento que define e atualiza um conjunto de diretrizes a serem seguidas para alcançar o desenvolvimento sustentável. Oferece uma sistematização de medidas concretas e incentivos para reduzir o impacto ambiental das nações e contribui para revitalizar o desenvolvimento em países, para eliminar a pobreza.

A Agenda 2030 também destaca uma proposta de ações a ser realizado por cidadãos individuais, incluindo os cinco P's do desenvolvimento sustentável: pessoas, planeta, prosperidade, parcerias e paz.

1.2. OBJETIVOS

O propósito deste projeto é a criação de um sistema que permita à administração local e central reconhecer sinergias entre as metas de desenvolvimento sustentável de forma a melhorar o cumprimento destas, assim como identificar boas práticas ao nível municipal, regional e nacional que possibilite a adoção de medidas e incentivo a projetos que tenham uma maior probabilidade de sucesso.

1.3. ANÁLISE DOS RISCOS DO PROJETO

Tabela 1 - Tabela de Análise de Riscos

Riscos do Projeto	Descrição do Risco	Gravidade do Risco	Probabilidade de Ocorrência	Impacto
Desvio do Escopo	Os objetivos iniciais não são bem definidos	Alto	Baixo	Alto
Prazos Curtos	Quando as tarefas levam mais tempo do que o esperado	Alto	Baixo	Alto
Excesso Carga de Trabalho	Quando um membro da equipa tem demasiadas tarefas a seu cargo	Moderado	Baixo	Alto
Falta de Comunicação	Falhas na comunicação entre os membros da equipa	Baixo	Baixo	Moderado
Falta de Conhecimento	Um ou mais membros da equipa não estarem à vontade com as tecnologias a adotar	Baixo	Moderado	Moderado
Ausência de um membro da equipa	Ausência de um ou mais membros da equipa por motivos pessoais ou doença	Baixo	Baixo	Moderado
Desistência de um membro da equipa	Um ou mais membros da equipa desistem da Unidade Curricular ou não pretendem continuar na equipa	Alto	Baixo	Alto

1.4. PLANO DE CONTINGÊNCIAS

1.4.1. Desvio do Escopo

Os objetivos do escopo foram bem definidos desde o início do projeto, de forma a evitar a entrega de um projeto que não ia de encontro ao que foi pedido inicialmente.

1.4.2. Prazos Curtos

Sempre que um membro da equipa tem noção que não vai conseguir terminar a tempo as tarefas a que se propôs fazer ou que lhe foram atribuídas, deve comunicar ao Gestor de Projetos para que este possa agilizar com os restantes membros da equipa a distribuição dessas mesmas tarefas.

1.4.3. Excesso Carga de Trabalho

Quando um membro da equipa tem noção de que tem demasiadas tarefas a seu cargo, deve pedir ajuda aos restantes membros da equipa.

1.4.4. Falta de Comunicação

Para que não exista falta de comunicação entre a equipa, foi criada uma equipa dentro do MS Teams e um grupo no WhatsApp.

1.4.5. Falta de Conhecimento

A falta de conhecimento relativamente à linguagem C# e WPF/Windows Forms teve de ser compensada com pesquisas online, vídeos no YouTube e livros.

1.4.6. Ausência de um membro do grupo

Se um membro da equipa não puder estar presente na aula ou mesmo reunido online por motivos pessoais ou de doença, deve comunicar aos membros da equipa através dos canais criados para o efeito (MS Teams ou WhatsApp).

1.4.7. Desistência de um membro do grupo

Caso um membro da equipa desista da Unidade Curricular ou não pretenda continuar na equipa, deve informar com antecedência a todos os membros da equipa ou pelo menos ao Gestor de Projetos da sua intenção. Caso esse membro seja o Gestor de Projetos, o mesmo deve reunir com a equipa e delegar as suas funções a um membro da equipa que tenha capacidades para desempenhar as mesmas. O mesmo acontece com as restantes funções de cada membro da equipa.

1.5. ESTRUTURA DO RELATÓRIO

O relatório encontra-se estruturado em quatro secções distintas, nomeadamente:

1. Introdução - Na introdução são explicados os objetivos e os riscos que analisamos que podem ocorrer e antever medidas de correção de forma a podermos evitar que os mesmos coloquem em risco a viabilidade do projeto.

2. Descrição do produto - Na descrição do produto, descrevemos o contexto do produto, as características que o produto possui, as restrições e dependências do mesmo.

3. Requisitos - Na seção de requisitos apresentámos uma ideia de como o projeto funcionará, temos os atores, casos de uso, requisitos funcionais, design e implementação e os requisitos não funcionais.

4. Arquitetura - Na arquitetura, temos a arquitetura física, lógica e da base de dados.

2. DESCRIÇÃO DO PRODUTO

2.1. CONTEXTO DO PRODUTO

O desenvolvimento deste projeto consiste na criação de um programa de seleção de municípios que apresente os seus possíveis projetos para cada município, através de uma análise de classificações dos projetos, por sua vez podemos identificar os tipos de *Nomenclature of Territorial Units for Statistics* (NUTS). No caso do tipo 2 ou 3, que queremos investigar, usamos o algoritmo APRIORI.

Relativamente ao programa, podemos analisar os conjuntos de itens frequentes determinados pelo APRIORI, que podem ser usados para determinar regras de associação que por sua vez também destacam tendências gerais num determinado município. À vista desse fator, podemos ver os níveis em que se encontram os SDG (ODS) de cada município, é algo que pode vir a facilitar a escolher os SDG que possam apresentar melhores resultados para cada município.

2.2. FUNCIONALIDADES DO PRODUTO

Dentro das suas funcionalidades, pode ser descrito como sendo um produto com a finalidade de visualização de projetos para cada município e também da exportação de resultados calculados com recurso à simulação usando o algoritmo APRIORI.

Os projetos irão ser inseridos na plataforma, juntamente com os seus dados, municípios e classificações, de seguida estes serão analisados pelo algoritmo e serão organizados em tabelas que estarão disponíveis para o utilizador visualizar consoante o que pretende procurar, podendo fazer uma procura pelo top 10 dos níveis mais alto ou mais baixos.

Um dos objetivos na organização dos SDG para cada município é que estes possuam um vasto número de características específicas que pode trazer grande impacto para cada município. Por sua vez, se a seleção destes mesmo SDG for incorreta pode não ser tão bem aplicado no seu município ou ter o seu devido impacto. Em contrapartida caso o SDG seja bem escolhido podemos ter grandes melhorias ou resultados a nível do seu município, visto que o seu real objetivo é ser útil e ter grande impacto.

2.3. RESTRIÇÕES E DEPENDÊNCIAS

Para que tenhamos ferramentas necessárias no desenvolvimento da aplicação, será preciso instalar um IDE, para isso a nossa equipa instalou o *Visual Studio Code*. Como a aplicação está desenvolvida na linguagem de programação C#, é necessário instalar novos packages, nomeadamente, .NET Desktop Development que disponibilizará as bibliotecas necessárias para o desenvolvimento da aplicação na linguagem pretendida. Como é necessário guardar os diversos dados dos vários municípios é essencial criar uma base de dados. Dessa forma foi criada uma base de dados em linguagem T-SQL e foi escolhido o MSSQL como SGB e o SSMS para configurar, gerir e administrar os componentes do SQL Server.

No funcionamento do programa, é unicamente necessário o programa ter acesso à respetiva base de dados onde foram inseridos os dados.

3. REQUISITOS

3.1. ATORES

Todos os tipos de atores podem ser compostos por um ou vários elementos, consoante a necessidade de utilização da aplicação.

3.1.1. Administrador de sistema

O Administrador é o responsável pela configuração e manutenção do software, assim como pela gestão de utilizadores (analistas de projeto).

3.1.2. Analista de projetos

O Analista de Projetos é o utilizador responsável pela inserção dos projetos municipais e suas respetivas classificações. Este utilizador pode também visualizar todos os projetos existentes, visualizar os resultados dos itens frequentes dos projetos através do algoritmo APRIORI, e exportar esses mesmos resultados.

3.2. CASOS DE UTILIZAÇÃO

O Administrador pode gerir utilizadores (outros administradores ou analistas).

O Analista pode efetuar várias operações com os projetos: importar os dados e classificações dos projetos a partir de um CSV; listar projetos e aplicar filtros de pesquisa nessa mesma listagem. A partir da Listagem, o Analista pode editar as informações de um projeto caso necessário, ou prosseguir para a execução do algoritmo APRIORI com os projetos selecionados. Após a execução do algoritmo, o Analista tem ainda a possibilidade de exportar os dados calculados em CSV.

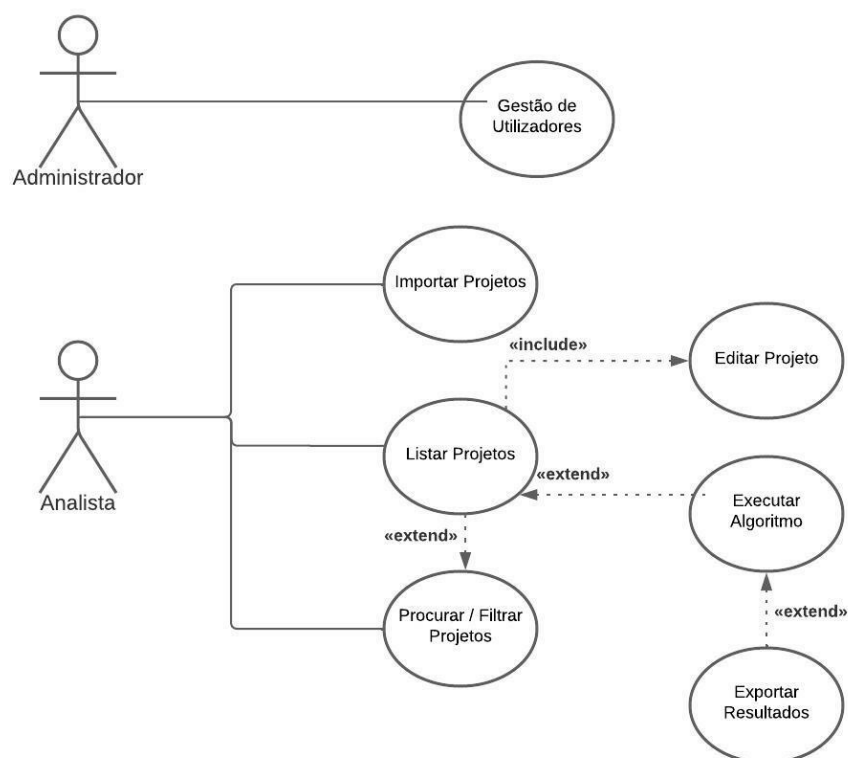


Figura 1 - Casos de Uso da Aplicação

3.3. REQUISITOS FUNCIONAIS

Tabela 2 - Requisitos Funcionais da Aplicação

Identificador	Nome	Atores	Descrição
RF01	Autenticação	Administrador Analista	O utilizador pode autenticar-se na aplicação através das credenciais fornecidas por um administrador. Caso contrário o acesso será negado.
RF02	Gestão de Utilizadores	Administrador	São mostrados todos os utilizadores existentes e permite criar novos e alterar ou apagar os existentes.
RF03	Listagem de Projetos	Analista	Tabela com a listagem de todos os projetos já inseridos na aplicação
RF04	Importar Projetos	Analista	O analista pode importar projetos e respetivas classificações através de um ficheiro CSV
RF05	Pesquisa de Projetos	Analista	Ferramenta de pesquisa à disposição do analista para que este possa pesquisar ou filtrar projetos por determinadas características.
RF06	Detalhes do Projeto	Analista	Ao abrir um projeto a partir da listagem, o analista pode visualizar toda a informação detalhada desse mesmo projeto.
RF07	Editar Projeto	Analista	O analista pode alterar dados de um projeto caso este tenha sido mal importado ou revisto posteriormente à importação.
RF08	Execução do Algoritmo	Analista	O analista seleciona o(s) Município(s), Nuts2 ou Nuts3, especifica o nível de suporte e inicia a execução do algoritmo APRIORI para cálculo das frequências dos seus itens.
RF09	Exportação dos Resultados do Algoritmo	Analista	Após a execução do algoritmo, o analista tem a possibilidade de exportar os resultados para um ficheiro CSV,

3.4. DESENHO E IMPLEMENTAÇÃO

3.4.1. Diagrama de atividades

3.4.1.1. Inserção do Projeto

O programador terá primeiro que adicionar os projetos no programa para aceder a tela principal do dashboard.

Na página principal encontrará um botão para inserir dados necessário para o projeto. Ao clicar no botão de inserção poderá subir os dados necessário para base dados, A inserção dos dados será validada pelas tabelas existente, caso os dados não estejam correto, será mostrado uma mensagem de alerta para que os dados seja verificados, caso os dados estejam corretos serão inseridos na base de dados e podemos estar preparado para o calculo do Dashboard.

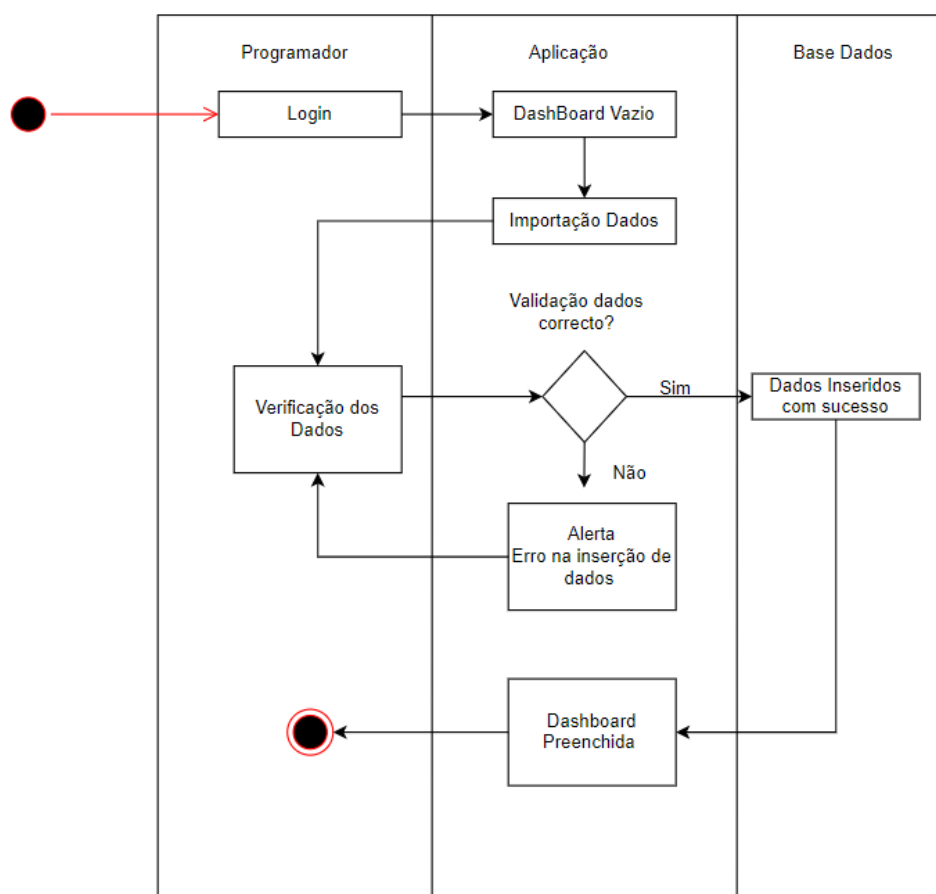


Figura 2 - Diagrama de atividade de inserção de projeto

3.4.1.2. Simulação e Exportação dos resultados

O programador terá primeiro que adicionar o projeto no programa para aceder a tela principal do dashboard.

Na página principal encontrará um botão de Apriori buscar dos dados a base de dados e realizar o cálculo do Apriori para gerar os resultados, depois os resultado serão mostrados no Dashboard com opção de exportação no final.

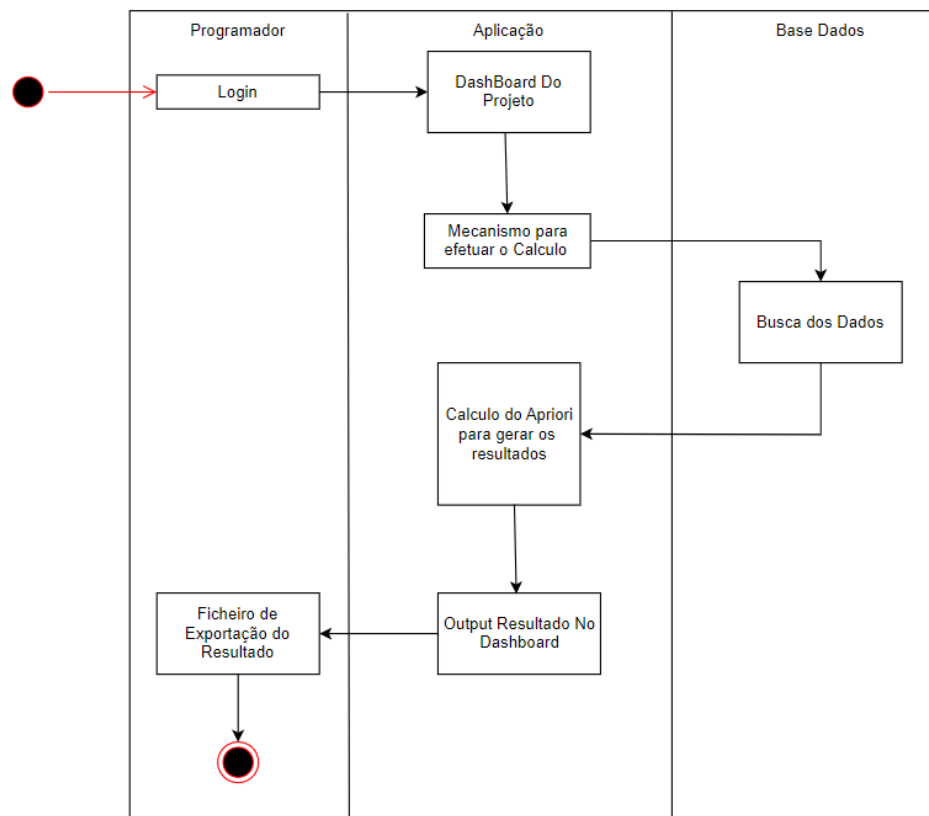


Figura 3 - Diagrama de atividade de simulação e exportação da equipa

3.4.2. Diagrama de sequência

O programador irá enviar os dados inseridos e verificar se há dados existentes para o login. O sistema retornará os dados, sendo verdadeiro e falso, e consoante a sua resposta será demonstrado na interface o resultado da verificação. Caso seja verdadeiro, o utilizador será redirecionado para a tela principal da aplicação. No caso de a resposta ser falsa, vai ser demonstrado um erro na interface ao coordenador.

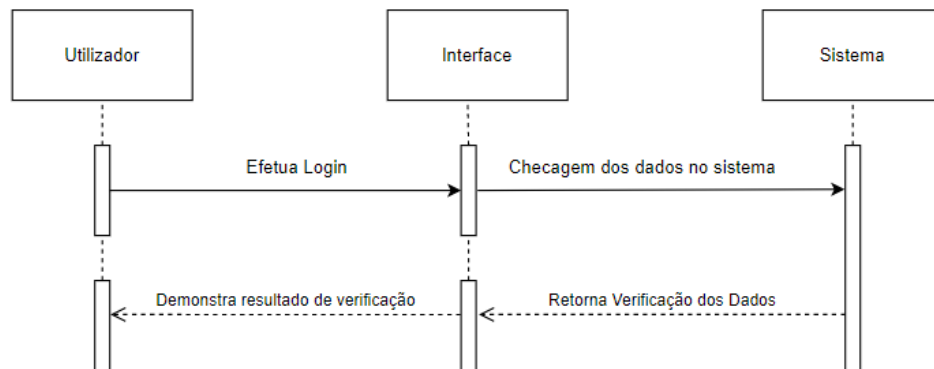


Figura 4 - Diagrama de sequência do login

3.4.3. Diagrama de Estados

O projeto pode assumir 3 estados:

- Projeto importar: Fase de validação para adicionar os dados dentro do programa
- Projeto Calcular: Fase de calculo do algoritmo para originar uma solução
- Projeto Exportar: Fase com possibilidade de exportação do resultado gerado

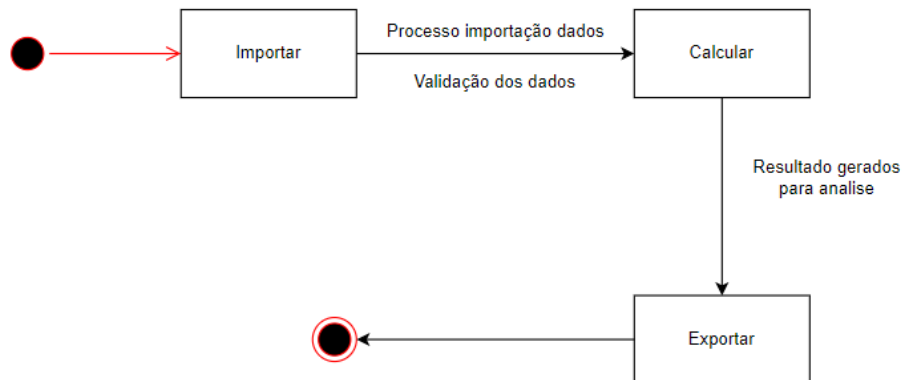


Figura 5 - Diagrama de estados do projeto

3.5. REQUISITOS NÃO FUNCIONAIS

Tabela 3 - Requisitos Não Funcionais da Aplicação

Identificador	Requisito	Descrição	Prioridade
RNF01	Usabilidade	O sistema deve ser intuitivo e de fácil utilização para os utilizadores, de forma a tornar fácil a procura e a edição de itens da aplicação.	Essencial
RNF02	Escalabilidade	A aplicação deve ser concebida de forma a permitir a adição e alteração de módulos ou funcionalidades, para que no futuro, se a informação obtida para cada projeto for diferente, seja possível a adaptação desta aplicação aos novos dados recolhidos e também abrir portas à adaptação da aplicação para outras áreas.	Desejável
RNF03	Fiabilidade	A aplicação deve conter dados atualizados e garantir a sua integridade, através de uma boa estrutura de base de dados, evitando perdas de informação.	Essencial
RNF04	Segurança	Garantir a proteção dos dados transmitidos e armazenados	Opcional
RNF05	Desempenho	Realização de pesquisas e manipulação de dados o mais rápido e eficientemente possível.	Desejável

4. ARQUITETURA

4.1. ARQUITETURA FÍSICA

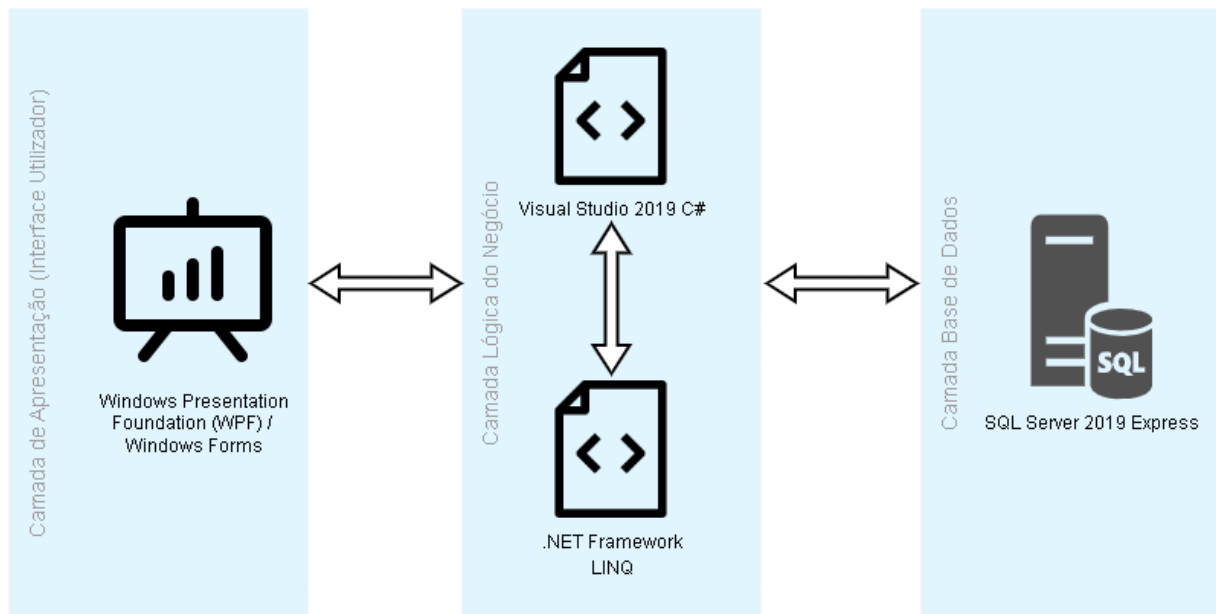


Figura 6 - Arquitetura Física

A camada da base de dados foi desenvolvida usando o SQL Server 2019 Express, que por sua vez permite fazer a gestão da base de dados relacional através do SQL Server Management Studio.

A camada lógica do negócio foi programada em C#, usando para esse efeito o Visual Studio 2019, que permitiu implementar todas as funcionalidades inerentes ao projeto, neste caso fazendo uso do algoritmo APRIORI. A comunicação entre a camada lógica e a camada de base de dados é estabelecida usando o LINQ, um componente .NET, que permite a consulta dos dados.

A camada de apresentação, ou interface de utilizador, foi desenvolvida em Windows Forms, uma solução da Microsoft que permite desenvolver aplicações para Windows.

4.2. ARQUITETURA LÓGICA

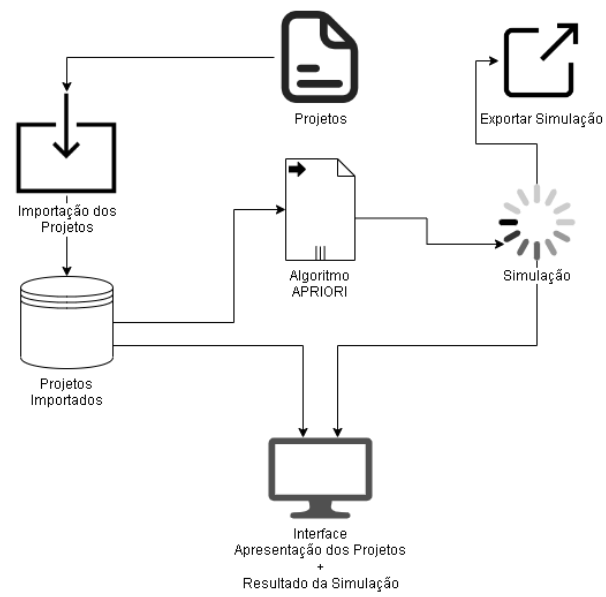


Figura 7 - Arquitetura Lógica

4.3. ARQUITETURA DA BASE DE DADOS

4.3.1. DIAGRAMA DE CLASSES

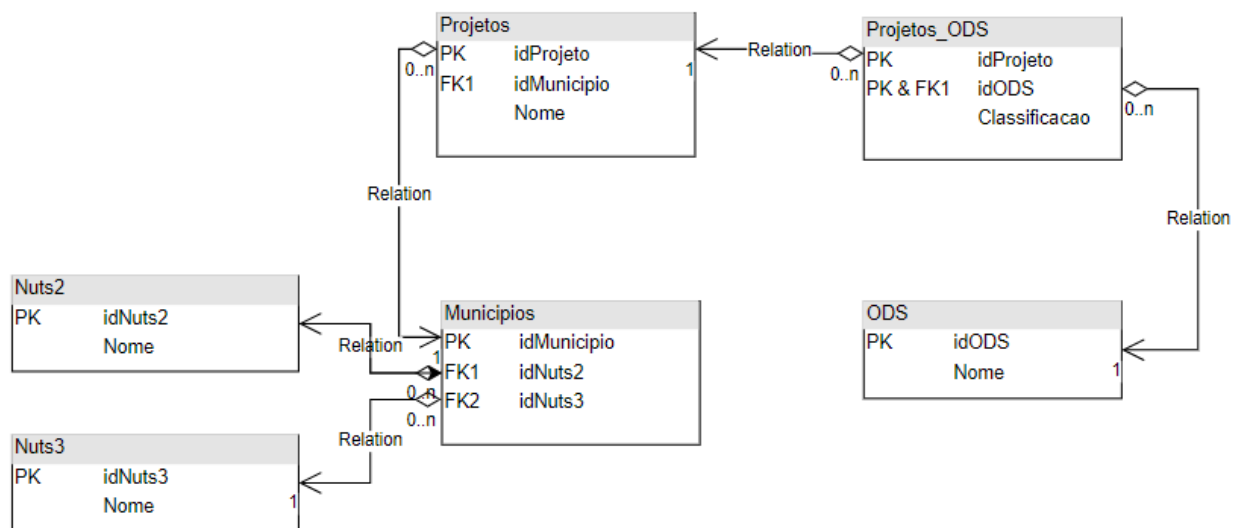


Figura 8 - Diagrama de Classes

4.3.2. MODELO RELACIONAL

Projetos(idProjeto, nome, idMunicipio -> Municipios)
Projetos_ODS(idProjeto,, classificacao, idODS -> ODS)
ODS(idODS, nome)
Municipios(idMunicipio, idNuts2 -> Nuts2, idNuts3 -> Nuts3)
Nuts2(idNuts2, nome)
Nuts3(idNuts3, nome)

4.3.3. EXPLICAÇÃO DO DIAGRAMA DE CLASSES

A tabela “Projetos” tem três atributos, sendo o “idProjeto” a chave primária da mesma e é do tipo INTEGER e auto-incremental. Possui uma chave externa para a tabela “Municipios”, com o atributo “idMunicipio”, que é do tipo INTEGER. O atributo “nome” é do tipo STRING.

Esta tabela tem uma relação de “um para muitos” com as tabelas “Projetos_ODS”. E também uma relação “muitos para um” com a tabela “Municipios”.

A tabela “Municipios” tem três atributos, sendo o “idMunicipio” a chave primária da mesma e é do tipo INTEGER e auto-incremental. Possui uma chave externa para a tabela “Nuts2”, com o atributo “idNuts2”, que é do tipo INTEGER, numa relação de “muitos para um”. Por fim, o atributo “idNuts3” assume função semelhante ao anterior, em relação à tabela “Nuts3”.

A tabela “Projetos_ODS” tem 3 atributos, sendo a chave primária composta pelos atributos “idProjeto” e “idODS”, ambos os atributos são do tipo INTEGER e auto-incremental. Possui uma chave externa para a tabela “ODS”, com o atributo “idODS”, numa relação de “muitos para um”. O atributo “classificação” é do tipo DOUBLE.

A tabela “ODS” tem 2 atributos, sendo o “idODS” a chave primária da mesma e é do tipo INTEGER e auto-incremental. Por fim, o atributo “nome” é do tipo STRING.

REFERÊNCIAS BIBLIOGRÁFICAS

- Almeida, F. (2022). The Contribution of Local Agents and Citizens to Sustainable Development: The Portuguese Experience. Sustainability.
DOI:10.3390/su141912696
- Leão, A., & Cunha, P. (2013). *Relatório de Especificação de Requisitos*.
Faculdade de Engenharia da Universidade do Porto.
- Asana. (n.d.). Sete riscos de projeto comuns e as maneiras de preveni-los • Asana. Asana.
Retrieved December 18, 2022, from <https://asana.com/pt/resources/project-risks>
- Augmented Startups (Director). (2017, October 31). Apriori Algorithm (Associated Learning) - Fun and Easy Machine Learning.
https://www.youtube.com/watch?v=WGIMIS_Yydk
- Apriori Algorithm. (2018, September 4). GeeksforGeeks.
<https://www.geeksforgeeks.org/apriori-algorithm/>