

Projecto Integrador

Enunciado

lapr3-2023-24 - versão 1.3

30 de novembro de 2023

Resumo

Este documento consiste no enunciado do trabalho prático a desenvolver no contexto do Projecto Integrador das Disciplinas ARQCP, BDDAD, ESINF, FSIAP e LAPR3. O trabalho consiste no desenvolvimento de uma solução informática que apoie a gestão de uma empresa responsável pela gestão de uma instalação agrícola em modo biológico. Neste documento são apresentados de forma breve: as especificidades da Agricultura Biológica; uma instalação agrícola e seus componentes; os documentos fundamentais na condução da instalação agrícola, o Caderno de Campo e o Plano de Regas e Fertilizações; por fim são apresentados os requisitos funcionais e não funcionais da solução a desenvolver, bem como modo de funcionamento e detalhes técnicos do Projecto Integrador.

Tabela 1: Histórico de Versões

Versão	Descrição
1.0	Versão inicial
1.1	Adição de USs de ARQCP, BDDAD, FSIAP e LAPR3 do sprint 2; Explicitação de conceitos: sectores de rega, estados fenológicos e operações agrícolas
1.2	Adição de USs de ESINF Adição de USs de LAPR3 e BDDAD relativas a documentos gerais Explicitação de USs de ARQCP
1.3	Adição de USs do sprint 3; Explicitação Sistema de Fertirrega

Conteúdo

1	Projecto Integrador	2
2	Descrição do Problema	3
2.1	Agricultura Biológica	3
2.2	Instalação Agrícola	4
2.3	Sistema de Rega	7
2.3.1	Fertirrega	8
2.4	Estação Meteorológica para Agricultura	9
2.5	Caderno de Campo	10
2.6	Comercialização e Distribuição de Produtos Agrícolas	10
2.7	Utilizadores do Sistema	11
3	Produto Viável Mínimo	11
3.1	Arquitectura de Computadores (ARQCP)	12
3.1.1	Sprint 2	13
3.1.2	Sprint 3	14
3.2	Bases de Dados (BDDAD)	18
3.2.1	Sprint 1	18
3.2.2	Sprint 2	20
3.2.3	Sprint 3	20
3.3	Estruturas de Informação (ESINF)	21
3.3.1	Sprint 2	22
3.3.2	Sprint 3	23
3.4	Física Aplicada (FSIAP)	24
3.4.1	Sprint 2	25
3.4.2	Sprint 3	26
3.5	Laboratório Projecto 3 (LAPR3)	28
3.5.1	Sprint 1	28
3.5.2	Sprint 2	29
3.5.3	Sprint 3	29
3.6	Requisitos não funcionais	30

1 Projecto Integrador

Neste projecto, os alunos deverão ser capazes de fazer a análise, design e implementação de uma solução informática que apoie a gestão de uma empresa responsável pela gestão de uma instalação agrícola em modo de produção biológico (MPB). Este projecto funciona como prova-de-conceito e abarca um conjunto de aspectos críticos na gestão de uma exploração agrícola em MPB, designadamente: a gestão de parcelas (campos), culturas, regas, comercialização de produtos agrícolas em rede, gestão de informação recolhida de sensores de solo e meteorológicos, e dimensionamento e utilização de armazéns agrícolas com necessidade de temperaturas controladas. De acordo com as boas práticas apreendidas no curso e em particular nas disciplinas de Arquitectura de Computadores (ARQCP), Bases de Dados (BDDAD), Estruturas de Informação (ESINF), Física Aplicada (FSIAP) e Laboratório de Projecto III (LAPR3), será utilizado um

processo de desenvolvimento iterativo e incremental. Pelo que deverá ser utilizada uma metodologia ágil baseada em SCRUM para gestão do trabalho de equipa em cada sprint com quatro semanas de duração.

A solução a desenvolver deverá ser composta por conjunto aplicações desenvolvidas em, PL/SQL, Java e C/Assembly, em função dos requisitos. Com o objectivo de aumentar a manutenibilidade da solução e respeitar as boas práticas de desenvolvimento de Software, a implementação deverá seguir uma abordagem TDD (Test-Driven Development).

2 Descrição do Problema

Nesta secção é descrito o problema a resolver, o desenvolvimento de uma solução informática que resolva alguns dos aspectos críticos no funcionamento de uma exploração agrícola em modo biológico. Desta forma é apresentada a agricultura biológica e as suas especificidades, caracterização de uma exploração agrícola e seus componentes incluindo sistemas rega e estações meteorológicas para agricultura. Também são apresentados os modelos de comercialização mais comuns nesta forma de agricultura.

2.1 Agricultura Biológica

A Agricultura Biológica, também designada por Orgânica, é uma forma de produzir alimentos que se baseia na sustentabilidade e responsabilidade agro-ecológica e social. Este modelo é preconizado pelas Nações Unidas como capaz de gerar alimento para a humanidade de forma sustentável e justa.



Figura 1: Agricultura Biológica (<http://www.draplvt.mamaot.pt/DRAPLVT/>)

A Agricultura Biológica permite reverter a massificação agrícola, tendo o solo como suporte de toda a cadeia alimentar, permite aos agricultores promoverem a saúde do ecossistema agrícola, uma vez que este modo de produção fomenta a biodiversidade, os ciclos biológicos e a actividade biológicas dos solos. A prática de rotação das culturas, a utilização de adubos orgânicos, o cultivo de variedades autóctones e a sua consorciação, bem como a utilização da luta biológica contra as pragas, fomenta uma interação entre o solo, as culturas, os animais e as pessoas, que formam assim, um ecossistema onde todos se influenciam, garantindo a sustentabilidade, bem como a preservação do solo

e do meio ambiente. Como resultado das consequências adversas que o modelo agro-industrial produziu no meio ambiente e no tecido social (em particular nos países menos desenvolvidos), nas décadas mais recentes surgiram múltiplos projectos e/ou empresas que têm por objectivo trabalhar em agricultura biológica .

A Agricultura Biológica caracteriza-se por:

- Seguir uma lógica policultural (múltiplas culturas na mesma instalação agrícola) e em menor escala, em múltiplos ciclos ao longo do ano agrícola com o objectivo de manter o sistema em equilíbrio e diminuir o número e dimensão das pragas que podem afectar as culturas;
- Incluir os animais no ciclo de produção pois são elementos que para além de produzirem bens de grande valor económico e alimentar (e.g., ovos, carne e mel) têm um papel fundamental no controlo de infestantes e pragas, polinização e fertilização das culturas;
- Combinar técnicas ancestrais agrícolas (como empalhamento do solo, adubação verde ou rotação cultural) com as mais recentes tecnologias que permitem: monitorizar a saúde das plantas, solos e condições climatéricas; produzir de energia (e.g., eléctrica de fontes solares ou eólicas, térmica de fontes solares ou composteiras); regar de forma eficiente; reduzir a quantidade de insumos a fornecer ao solo e por consequência reduzir o impacto ambiental da actividade agrícola;
- Privilegiar cadeias curtas de consumo (isto significa que existem poucos ou nenhuns intermediários entre produtor e consumidor final);
- Favorecer a cooperação através de redes de comercialização que agreguem múltiplos operadores de pequena dimensão;
- Proteger os recursos comuns, como a água, a atmosfera e os solos.

2.2 Instalação Agrícola

Uma instalação agrícola, ou exploração, é tipicamente uma quinta composta por um conjunto de parcelas e edifícios. Os edifícios que podem ser: estábulos para animais; garagens para máquinas e alfaías; sistemas de rega incluindo tanques agrícola; e armazéns para colheitas, para factores de produção (e.g., fertilizantes) e para rações animais.

- Parcela Agrícola – uma parcela, também designada por campo, é caracterizada por uma designação, área (ha) e culturas (macieira, pereira, trigo, feijão), que podem variar ao longo do tempo. É possível ter simultaneamente mais do que uma cultura numa mesma parcela.
- Cultura – uma cultura é uma espécie vegetal cultivada na agricultura com o objectivo de produzir produtos agrícolas para consumo humano/animal (e.g., flores, frutos, cereais, hortícolas, forragens, fibras) ou para produzir adubação verde. Uma cultura pode ser do tipo permanente (e.g., frutícolas como pereiras ou macieiras) ou temporária (e.g., hortícolas ou forragem).



Figura 2: Vista aérea de um conjunto de parcelas e edifícios agrícolas

Uma parcela pode ter várias culturas, por exemplo, é possível ter milho, feijão e abóboras porque criam "sinergias", chamadas consorciações (os feijões fixam o azoto no solo, o azoto é fundamental para o milho, as canas do milho protegem as aboboras do sol, e também servem como estacas para o feijão, para além disso as raízes das aboboreiras são muito fortes e profundas, e como tal arejam o solo). Este é um clássico mas existem muitos exemplos destes, seja para repelir pragas, para evitar o aparecimento de infestantes.

Por outro, uma cultura pode estar instalada em múltiplas parcelas, dependendo das necessidades de produção.

- Factor de Produção – os factores de produção são todos os produtos que são aplicados no solo ou nas plantas, por forma a melhorar e nutrir o solo e as plantas, prevenir doenças, corrigir desequilíbrios, e combater pragas e doenças. Podem ser classificados como fertilizantes, adubos, correctivos ou produtos fitofármacos. Um determinado produto é caracterizado pelo nome comercial, formulação (líquido, granulado, pó) e ficha técnica (que inclui entre outros uma lista de elementos e substâncias existentes no produto e respectivas quantidades);
- Operação Agrícola - é toda a acção que é desenvolvida no contexto da actividade agrícola, designadamente: semear, plantar, podar, mondar, desfolhar, tutorar, regar, fertilizar, colher. As operações são usualmente realizadas por operadores frequentemente auxiliados por máquinas e instrumentos agrícolas (como tractores ou tesouras de poda). Podem implicar a utilização de factores de produção. As operações agrícolas são realizadas ao longo do ano, normalmente em função do estado fenológico da cultura e de acordo com uma sequência lógica. De notar que existem operações que são realizadas sobre parcelas nas quais podem não estar instaladas culturas (por exemplo preparação de solo antes de uma sementeira);
- Estado Fenológico - são etapas de desenvolvimento da cultura/planta que podem ser identificáveis por alterações da morfologia externa da mesma, ao nível dos ramos, folhas, flores ou frutos. A evolução e o momento em que acontecem os estados

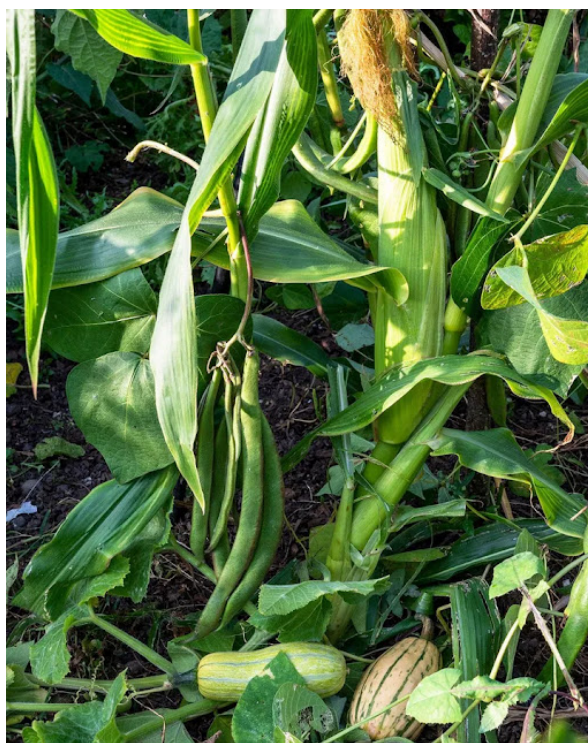


Figura 3: Consorciação de culturas

é determinado pelas condições meteorológicas (luz e temperatura), variedades e manejo cultural. Os estados fenológicos são determinantes para a realização das operações agrícolas.



Figura 4: Estados Fenológicos no mirtilo

Na figura 4 são apresentados a título de exemplo os estados fenológicos da cultura do mirtilo, 1. Gema invernal dormente 2. Gema inchada, 3. Brácteas separadas, 4 pré-floração, 5. Início da floração, 6. Plena floração, 7. Fim de floração, 8. Formação do fruto, 9. Troca de cor do fruto e 10. Frutos maduros para a colheita.

O objetivo de uma cultura é obter uma colheita, nem que seja para adubação verde.

Para facilitar a gestão da instalação agrícola, pretende-se gerir a informação em termos de ciclo de cultura, i.e. uma plantação/preparação e uma ou mais colheitas. Isto é evidente nas plantações temporárias. Nas plantações permanentes considera-se o ciclo vegetativo. Findo este ciclo considera-se que a plantação se pode manter com as mesmas características da plantação anterior ou sofrer alterações (e.g. alterações no número de plantas, etc.), note-se em que qualquer dos casos a idade (calculada das árvores aumenta). A cada plantação está sempre associada uma ou mais colheitas previstas (semana e quantidade). A análise das diferenças entre as previsões e os resultados reais é uma tarefa importante do gestor agrícola.

2.3 Sistema de Rega

Um sistema de rega/fertirrega é um conjunto de equipamentos capaz de fazer chegar às culturas água ou soluções aquosas contendo factores de produção (e.g., fertilizantes). A dimensão e complexidade destes sistemas pode variar muito em função de múltiplos factores, designadamente, da dimensão da exploração, do número de culturas e parcelas, da quantidade e qualidade da água utilizada, variedade de factores de produção utilizados. A rega pode ser realizada por gravidade (quando exista um desnível entre a fonte e as culturas) ou bombeada (através de bombas hidráulicas). A distribuição propriamente dita pode ser feita por aspersão, gotejamento ou pulverização, em funções das necessidades das diferentes culturas. A água é transportada ao longo da exploração através de um sistema de tubagens (primário) e posteriormente distribuída nas parcelas por tubagens de menor dimensão (secundário). O sistema também inclui diferentes filtros necessários para manter as tubagens, aspersores, gotejadores e/ou pulverizadores livres de micro-algas e sedimentos que a água pode transportar.

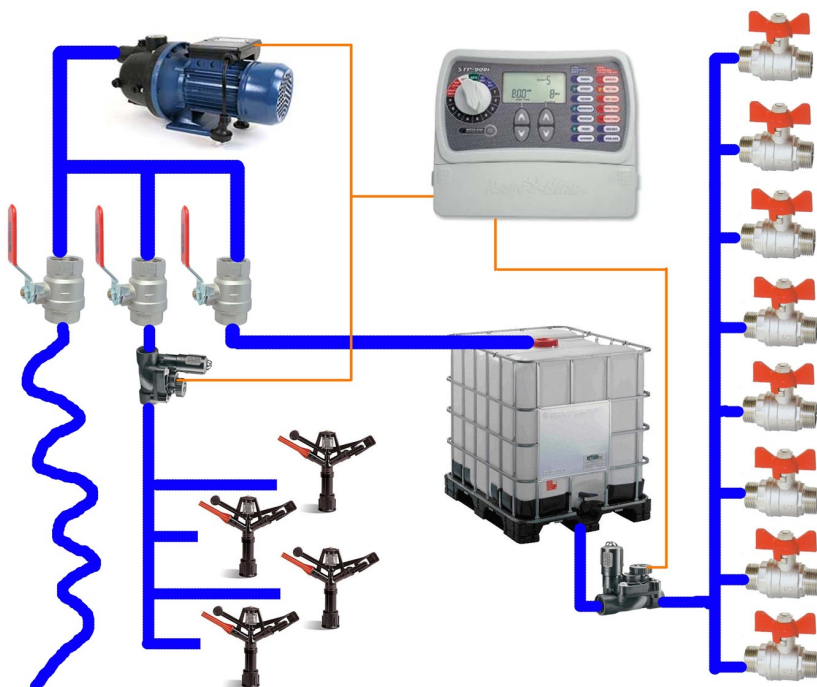


Figura 5: Sistema Rega
fonte <https://forumdacasa.com/>

O controlador é um elemento fundamental do sistema de rega pois permite, através de um conjunto de electro-válvulas, dispensar a água ou as soluções aquosas de acordo com um plano de rega. Neste plano está definida uma ordem com os sectores a regar, os tempos de rega (convertível em quantidade de rega) e a periodicidade com que é realizada a rega em cada sector.

As zonas a irrigar na quinta são agrupadas em sectores de rega. Estes sectores são estruturados de acordo com alguns factores, designadamente:

- Débito máximo permitido pelo sistema (componente de bombagem, tubagens), sendo que um sector de rega não pode exceder a capacidade máxima do sistema;
- Culturas a irrigar, culturas diferentes têm usualmente necessidades de rega e factores de produção (aplicados na fertirrega) diferentes pelo que têm de estar em sectores de rega distintos;
- A forma como é dispensada a água (aspersão, gotejamento, micro-pulverização). Um sector, usualmente só compreende uma das formas.

Em termos práticos, um sector de rega pode servir múltiplas parcelas, por exemplo quando estas tenham uma dimensão limitada e a mesma cultura. Por outro, quando uma parcela tenha grandes dimensões, tenha culturas distintas e formas de dispensar a água, pode ser servida por diferentes sectores de rega.

Em função das culturas instaladas é definido um plano (tipicamente anual) das necessidades de rega, correcções minerais e fertilizações a realizar ao longo da época agrícola. Estas aplicações podem ser realizadas por via foliar (pulverização das culturas), através do sistema de rega ou aplicação directa ao solo. As aplicações a realizar discriminam as parcelas, culturas, datas previstas, factores de produção e respectivas quantidades.

2.3.1 Fertirrega

A fertirrigação é uma técnica de fertilização que utiliza a água de irrigação para fazer a aplicação de factores de produção ao solo cultivado (ver 6). Esta aplicação pode ser realizada através do sistema de irrigação mais conveniente à cultura, podendo-se utilizar técnicas como micro-irrigação (por gotejamento ou por micro-aspersão), aspersão (sob pivô central ou convencional).

A solução que é adicionada à água pode ser um factor de produção ou uma mistura (mix). Considere os seguintes exemplos de misturas:

mix1

Epsotop/K+S, 1.5 kg/ha

Solusop/K+S, 2.5 kg/ha

Floracal/Plymag, 1.7 l/ha

mix2

MOL (Matéria Organica Liquida)/Tecniferti, 60 l/ha

AllGrip/AsfertGlobal, 2 l/ha

mix3

Cuperdem/AsfertGlobal, 2.2 l/ha



Figura 6: Sistema de fertirrega
fonte <https://insularregas.com/servicos/>

2.4 Estação Meteorológica para Agricultura

Uma estação meteorológica para agricultura (EMA) é constituída por um conjunto de sensores que permitem medir grandezas atmosféricas como o vento, temperatura, humidade, radiação e pressão. Estas estações são actualmente um elemento muito importante para a condução de uma exploração sustentável. Uma previsão adequada das condições climáticas, designadamente, vento, humidade e temperatura permite adequar as quantidades de água necessária na rega bem como evitar a aplicação de factores de produção que poderiam ser lixiviados em caso de ocorrer chuva após a sua aplicação. Esta adequação pode implicar poupanças muito significativas em termos de água, energia e factores de produção.

Para além dos sensores meteorológicos existentes na EMA, numa instalação agrícola existem sensores que estão instalados no solo. Assim, na estação meteorológica não terá em princípio mais do que um sensor de um determinado tipo mas podem existir múltiplos sensores do mesmo tipo no solo (por exemplo de humidade) na instalação e/ou na parcela. Sendo que Uma estação meteorológica pode ser um equipamento dispendioso, tipicamente servirá toda a instalação, tanto mais que as grandezas meteorológicas não variam significativamente em áreas pequenas. Por outro lado, os sensores de solo podem existir vários por parcela, depende da dimensão da parcela e da precisão desejada dos valores medidos.

Os principais sensores instalados numa estação meteorológica para agricultura medem grandezas como: pluviosidade, velocidade e intensidade do vento, temperatura, humidade do ar e pressão atmosférica. Enquanto que os sensores de solo medem grandezas como: a humidade, temperatura, pH e salinidade.

2.5 Caderno de Campo

O Caderno de Campo é um documento formal (obrigatório) que permite registar todas as operações agrícolas relevantes ocorridas na exploração, em particular:

- Fertilizações (ou aplicação de um qualquer Factor de Produção permitido) realizados por: via foliar, fertirrega ou aplicação directa ao solo com discriminação dos factores e quantidades aplicadas, data da realização, parcela. Este registo as entradas do Plano de Rega e Fertilizações efectivamente realizados bem como acções extra (não planeados);
- Resumo dos dados recolhidos dos sensores meteorológicos e solo;
- Registo colheitas incluído produto, quantidade colhida, data e parcelas;
- Regas executadas, quantidades, data da realização, parcela;
- Operações agrícolas e acções culturais realizadas.

2.6 Comercialização e Distribuição de Produtos Agrícolas

No contexto da Agricultura Biológica são privilegiadas as cadeias de consumo curtas e locais, idealmente sem intermediários entre o agricultor e o consumidor. As razões que suportam esta posição são essencialmente, redução dos custos ambientais associados ao transporte de alimentos ao longo de grandes distâncias (designado de turismo alimentar), preservação da qualidade dos alimentos, criação de vínculo entre consumidores e produtores, e garantia de um retorno económico mais justo para o agricultor.



Figura 7: Mercados Locais

<https://grandeconsumo.com/tag/biologicos/>

São múltiplas as formas de comercializar e distribuir os alimentos produzidos nesta forma de agricultura. Para além da mais simples venda directa ao consumidor, na quinta ou em mercados locais. Existem por exemplo, Grupos de Consumo, Associação

de Manutenção da Agricultura de Proximidade¹ ou Community Supported Agriculture² e também distribuição em rede muitas vezes com base em Portais e soluções informáticas (e.g., Prove³ ou Good Food Hub⁴) que têm a capacidade de agregar múltiplos consumidores e produtores.

2.7 Utilizadores do Sistema

Este sistema pode ser potencialmente utilizado por múltiplos utilizadores, designadamente:

- Gestor Agrícola – pessoa que gere culturas nas parcelas, realiza as diferentes acções culturais e as regista no Caderno de Campo
- Cliente – pessoa que encomenda e consome os produtos agrícolas, distribuídos sob a forma de cabazes
- Conductor – pessoa que recolhe os cabazes na exploração agrícola e os deposita nos Hubs de distribuição
- Gestor de Distribuição – pessoa que gere processo de recolha e transporte dos produtos agrícolas entre explorações, Hubs e posterior recolha pelos clientes

3 Produto Viável Mínimo

O projecto a desenvolver deverá ter em consideração a arquitectura apresentada no diagrama de componentes da 8.

O objectivo deste projecto é desenvolver um Produto Viável Mínimo de forma iterativa e incremental, desta forma, o trabalho será dividido em três Sprints:

- Sprint 1 – semanas 3 a 6 – de 2/Outubro a 29/Outubro
- Sprint 2 – semanas 7 a 10 – de 30/Outubro a 26/Novembro
- Sprint 2 – semanas 11 a 14 – de 27/Novembro a 3 de Janeiro

Uma descrição do PVM é fornecida para cada sprint. As equipas devem seguir as USs fornecidas e ter em conta o seu encadeamento e respectivas dependências e no final de cada sprint cada equipa deve conseguir satisfazer os requisitos especificados. As equipas devem ser capazes de adicionar as USs ao backlog, dimensioná-las adequadamente e distribuí-las pelos membros da equipa. Por simplicidade de leitura as US estão separadas por Unidade Curricular.

¹AMAP - <https://amap.movingcause.org/>

²CSA - <https://communitysupportedagriculture.org.uk/>

³<http://www.prove.com.pt/www/>

⁴<https://goodfoodhubs.pt/>

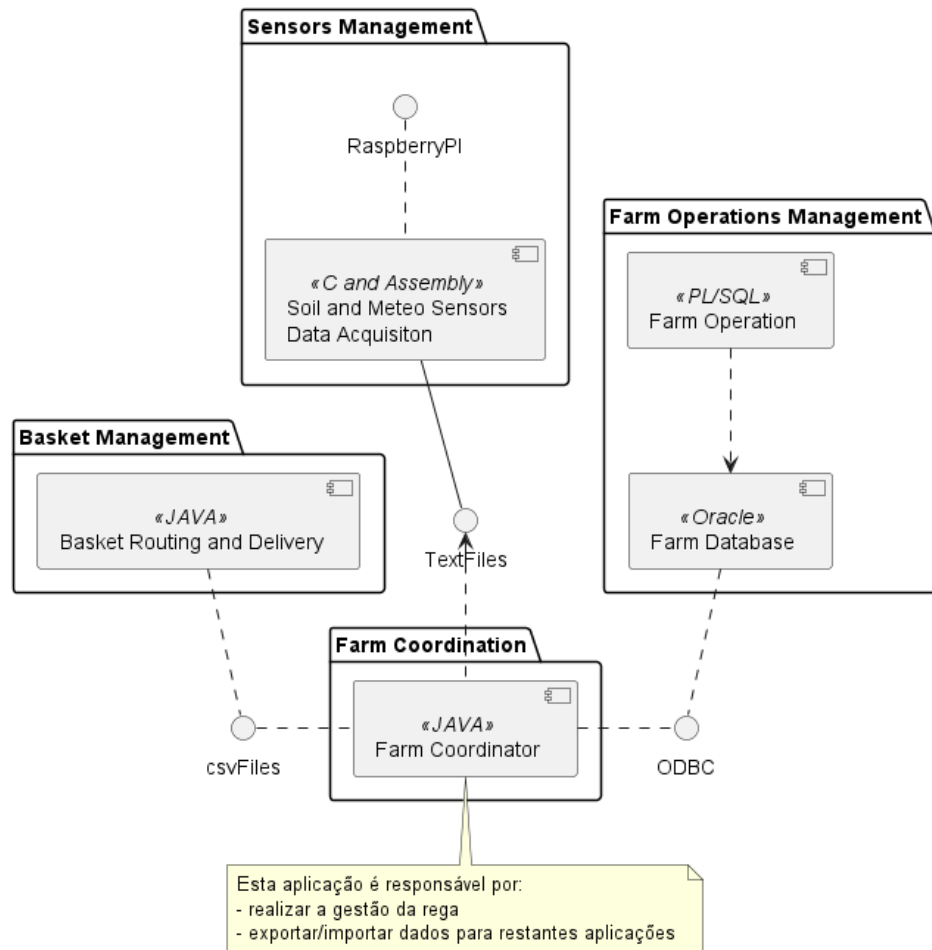


Figura 8: Arquitectura sugerida para solução a desenvolver

3.1 Arquitectura de Computadores (ARQCP)

ARQCP é responsável por implementar o componente Sensors Management (ver figura 8). Este componente deve seguir a arquitectura lógica apresentada na Figura 9.

O componente ColetorDeDados é responsável por periodicamente obter dados dos sensores.



Figura 9: Arquitectura lógica do componente Sensors Management

O componente ProcessadorDeDados recebe os dados do componente ColetorDeDados, processa-os e disponibiliza-os ao componente de SaidaDeDados.

O componente SaidaDeDados periodicamente gera um ficheiro com a informação recebida do componente ProcessadorDeDados.

Em termos físicos o componente ColetorDeDados é composto por um Microcontrolador Raspberry Pi Pico W que tem um conjunto de sensores acoplados e periodicamente obtém os dados dos sensores.

Cada sensor é caracterizado por um identificador e pelo tipo (ao qual está associado a unidade de medida). Existem sensores de vários tipos: temperatura, velocidade do vento, direção do vento, humidade atmosférica, humidade do solo e pluviosidade.

Para cada sensor o componente ColetorDeDados envia a seguinte informação e no seguinte formato:

```
\sensor_id:" + id + \#type:" + type + "#value:" + value + "#unit:" + unit + "#time:" + time  
sensor_id:8#type:atmospheric_temperature#value:21.60#unit:celsius#time:2470030
```

onde id é um valor numérico que identifica o sensor, type é uma string que identifica o tipo de sensor, value é um valor número com a medida lida do sensor (com duas casas decimais), unit é uma string com a indicação da unidade de medida e time é o timestamp da leitura (valor em milissegundos).

sensor_id, type, value, unit e time serão designados como tokens.

O componente ProcessadorDeDados é responsável por efetuar o processamento da informação dos sensores, como por exemplo calcular dados estatísticos de cada sensor, aplicar a técnica de filtragem MovingMedian⁵.

Determinar o que é um outlier e etc. Alertar o sistema quando está um determinado período sem receber dados de um determinado sensor. A componente tem uma restrição que é: não suporta números reais, só inteiros. Contudo, os valores das medições são números reais com duas casas decimais.

O componente SaidaDeDados sempre que recebe dados do componente ProcessadorDeDados cria um ficheiro de texto com esses dados. O nome para os ficheiros segue um padrão (a definir) assim como o formato de representação da informação. Este componente ao contrário do ProcessadorDeDados suporta números reais e terá de transformar os valores das medições (value) em números reais com duas casas decimais.

A implementação da componente de ARQCP terá de seguir uma estrutura de acordo com a arquitetura lógica apresentada. Portanto, terá de existir uma pasta por cada componente cujo nome deve ser igual: ColetorDeDados; ProcessadorDeDados e SaidaDeDados.

3.1.1 Sprint 2

Neste sprint pretende-se implementar as seguintes US relacionadas com a componente ProcessadorDeDados. As componentes de ProcessadorDeDados e SaidaDeDados devem ser implementadas na máquina virtual usada em ARQCP.

- USAC01 Desenvolva uma função em assembly que dada uma string (input) com a informação gerada pelo ColetorDeDados e um token extraia a informação (campo value da string) referente a esse token (output).

```
void extract_token(char* input, char* token, int* output);
```

É preciso ter em atenção que o campo value da string é um número real. Este componente tem uma limitação que é só suporta números inteiros.

⁵<https://reference.wolfram.com/language/ref/MovingMedian.html>

- USAC02 Desenvolva uma função em assembly que insira um valor (int value) num array (int * array). Este array deve ser implementado de acordo com as regras de um buffer circular.

```
void enqueue_value(int* array, int length,
    int* read, int* write, int value);
```

Um buffer circular é um array de tamanho fixo (definido por length) cuja política de inserções e remoções dos elementos do array permite a criar a circularidade, isto é, faz com que o fim do array pareça estar ligado ao início (int * read, int * write). Se o array estiver cheio o elemento mais antigo no array deve ser descartado.

- USAC03 Desenvolva uma função em assembly que remova n valores (int num) de um array (buffer circular) e os copie para outro array (int * vec)

```
int move_num_vec(int* array, int length, int *read,
    int *write, int num, int* vec);
```

No caso de não existirem os n elementos no array, a função não deve copiar os elementos e retornar 0, caso contrário deve retornar 1.

- USAC04 Desenvolva uma função em assembly que ordene um dado array.

```
void sort_array(int* vec, int num);
```

- USAC05 Desenvolva uma função em assembly que dado um array retorne a sua mediana, no caso de num ser um número par, deve retorna o elemento que está na posição num /2.

```
int mediana(int* vec, int num);
```

3.1.2 Sprint 3

De acordo com arquitectura proposta na figura 9 é necessária a implementação de três componentes: ColetorDeDados, ProcessadorDeDados e SaidaDeDados. A implementação base do componente ColetorDeDados foi disponibilizada através de um tutorial disponível na página moodle de ARQCP.

Relativamente ao componente ProcessadorDeDados importa referir alguns conceitos:

- **serializar** - consiste em transformar os dados que estão armazenados numa estrutura de dados em texto (numa string) de acordo com um determinado formato.
- **sensor em situação de erro** - um sensor está numa situação de erro quando a diferença entre o instante do último registo e o instante actual é superior a um dado valor designado de timeout.

O componente `ProcessadorDeDados` tem como entrada de dados (através da função `main`) quatro argumentos: `a`, `b`, `c` e `d`

- `a` - o caminho do ficheiro que permite receber dados do componente `ColetorDeDados`
- `b` - o ficheiro de configuração do componente
- `c` - o diretório para colocar os ficheiros os dados para o componente `SaidaDeDados`.
- `d` - o número de leituras que faz antes de enviar dados para o componente `SaidaDeDados`.

exemplo:

```
./processadordedados /etc/ttyS0 config intermedio 50
```

O ficheiro `config`, de texto, com a configuração e em que cada linha tem a seguinte estrutura:

```
sensor_id#type#unit#buffer_len#window_len#timeout
```

onde:

- `sensor_id` - o identificador do sensor
- `type` - a designação do tipo de sensor
- `unit` - unidade de medida do sensor
- `buffer_len` - o tamanho do buffer circular para guardar os valores do sensor
- `window_len` - refere-se ao número de elementos a considerar para obter a mediana⁶ (define o valor do parâmetro `r` que aparece na expressão `MovingMedian[list,r]`).
- `timeout` - o valor em milissegundos a partir do qual se considera que o sensor não está a efetuar leituras, isto é, está em situação de erro.

Exemplo:

```
1#soil_humidity#percentage#50#10#40000
2#soil_humidity#percentage#60#15#50000
3#atmospheric_humidity#percentage#70#10#20000
4#atmospheric_humidity#percentage#80#20#20000
5#atmospheric_humidity#percentage#50#15#20000
6#atmospheric_humidity#percentage#50#10#50000
7#atmospheric_temperature#celsius#40#10#80000
8#atmospheric_temperature#celsius#50#14#400000
9#atmospheric_temperature#celsius#50#13#20000
10#atmospheric_temperature#celsius#50#12#20000
```

⁶<https://reference.wolfram.com/language/ref/MovingMedian.html>

O componente deve criar as estruturas de dados necessárias para a gestão da informação recebida por cada sensor do componente ColletorDeDados. Portanto, deve criar um tipo de dados específico (uma struct) para gerir cada sensor:

- sensor_id,
- type,
- unit,
- Buffer circular (alocado dinamicamente) (recomenda-se a criação de um tipo de dados específico para modelar o buffer circular)
- array para o cálculo da mediana (alocado dinamicamente)
- instante temporal da última leitura recebida.
- timeout
- write_counter, contador de escritas.

Dado que o número de sensores não está definido, depende do número de linhas do ficheiro de configuração, o elemento que vai acolher os dados de todos os sensores deve, também, ser criado dinamicamente.

O funcionamento do componente ProcessadorDeDados é cíclico:

Algorithm 1: An algorithm with caption

```

while 1 do
    ContadorLeitura  $\leftarrow$  0;
    repeat
        Read data from sensor ;      /* received from ColectorDedados */
        Extract info;
        Insert info into sensor Data Struct;
    until ContadorLeitura=d;
    foreach sensor do
        Compute median; write_counter  $\leftarrow$  write_counter + 1;
        serialize info;
    end
    create the 'AAAAMMDDHHMMSS_sensors.txt' file and write all sensors
    info ;                          /* sent to SaidaDeDados */
end

```

O nome do ficheiro deve ter a seguinte designação: AAAAMMDDHHMMSS_sensors.txt. Onde AAAA é o ano, MM o número do mês, DD, o dia, HH hora, MM, minutos, SS segundos. Portanto, com exceção do ano todos os elementos tem dois dígitos. O ano tem quatro.

Estrutura das entradas no ficheiro AAAAMMDDHHMMSS_sensors.txt:

```

sensor_id,write_counter,type,unit,mediana#
sensor_id,write_counter,type,unit,mediana#
...

```

Exemplo de uma entrada no ficheiro com um sensor em erro (neste caso o sensor com o sensor_id igual 3:

```
1,1,soil_humidity,percentage,2160#
3,1,atmospheric_humidity,percentage,error#
4,1,atmospheric_humidity,percentage,2000#
....
```

O componente SaidaDeDados tem como entrada de dados (através da função main) três argumentos:

- a - o diretório que o componente ProcessadorDeDados colocou os ficheiros para o componente SaidaDeDados.
- b - o diretório para colocar os ficheiros os dados para o componente Farm Coordinator.
- c - a periodicidade das execuções do componente SaidaDeDados em milisegundos

exemplo:

```
./SaidaDeDados intermedio output 60000
```

O componente SaidaDeDados periodicamente (de acordo com o valor dado, no exemplo 60000 milisegundos) lê os últimos ficheiros, transforma os valores dos sensores em números reais com duas casas decimais e cria um ficheiro de texto com os últimos dados de cada sensor, que coloca na pasta adequada.

Portanto, o ficheiro contém os dados de todos os sensores, contudo, só um registo por cada sensor. O formato do ficheiro será definido pelo cliente.

Todas as as funcionalidades devem ser implementadas com a linguagem de programação C. As funcionalidades do Sprint 2 (desenvolvidas em Assembly terão de ser integradas no desenvolvimento deste sprint).

Componente **ProcessadorDeDados**:

- USAC06 - Altere a função:

```
void extract_token(char* input, char* token,int* output)
```

desenvolvida na USAC01 para:

```
int extract_token(char* input, char* token, int* output)
```

que retorna 1 em caso de conseguir extrair a informação do token e 0 em caso contrário.

Critério de aceitação: Esta função deve ser implementada em Assembly.

- USAC07 - Desenvolva uma função que aloque dinamicamente as estruturas de dados necessários para o componente e configure o componente (verifique se os diretórios necessários existem, caso não existam deve criar esses diretórios).

- USAC08 - Desenvolva uma função que receba os dados enviados pelo componente ColetorDeDados.
- USAC09 - Desenvolva uma função que insira os dados recebidos do componente ColetorDeDados nas estruturas de dados.
- USAC10 - Desenvolva uma função que serializa a informação armazenada nas estruturas de dados e a escreva num ficheiro de texto
- USAC11 - Desenvolva uma função que implemente o Algoritmo do componente.

Componente **SaidaDeDados**:

- USAC12 -Desenvolva uma função que aloque dinamicamente as estruturas de dados necessários para o componente e configure o componente (verifique se os diretórios necessários existem, caso não existam deve criar esses diretórios).
- USAC13 - Desenvolva uma função que periodicamente lê os últimos ficheiros, transforma os valores dos sensores em números reais com duas casas decimais e cria um ficheiro de texto com os últimos dados de cada sensor, que coloca no directório adequado. Portanto, o ficheiro contém os dados de todos os sensores, contudo, só um registo por cada.

3.2 Bases de Dados (BDDAD)

Neste componente deve ser concebida uma Base de Dados que modele uma exploração agrícola que permita suportar as seguintes USs:

3.2.1 Sprint 1

- USB01 - Como Product Owner, pretendo que seja elaborado dicionário de dados/glossário.
- USB02 - Como Product Owner, pretendo que seja elaborado o modelo relacional (nível lógico).

Critérios de aceitação:

- O modelo de dados deve cobrir a atividade da instalação agrícola, ficando excluída a comercialização e distribuição de produtos e a gestão da rega. Apesar do sistema de rega fazer parte da instalação agrícola, não é necessária manter na base de dados a topologia detalhada. Pode-se considerar cada sistema de rega como uma "caixa preta", associando cada sistema às parcelas que rega.
- Esperado: modelo "básico" partilhado no Visual Paradigm ou docs no git (e.g., PlantUML).
- Acima do esperado: modelo completo no Visual Paradigm, que inclua todas as restrições e que permita a geração automática de script de modelo físico.

- USBD03 - Como Product Owner pretendo que seja instanciado o modelo relacional (nível físico).
 - Será demonstrado no Oracle LiveSQL.
 - Esperado: criação manual do script
 - Acima do esperado: geração automática a partir do Visual Paradigm (gestão centralizada de alterações)
- USBD04 - Como Product Owner pretendo que sejam importados os dados de um sistema legacy, fornecidos numa folha de cálculo.
 - Esperado: construção manual dos scripts de inserção dos dados
 - Acima do esperado: geração automática do código SQL de inserção de dados a partir da folha de cálculo (e.g. fórmulas Excel, scripts noutra linguagem qualquer, etc.)
- USBD05 - Como Gestor Agrícola, pretendo saber a quantidade de produtos colhidos numa dada parcela, para cada produto, num dado intervalo de tempo.
- USBD06 - Como Gestor Agrícola, pretendo saber o número de fatores de produção aplicados numa dada parcela, para cada tipo de fator, num dado intervalo de tempo.
- USBD07 - Como Gestor Agrícola, pretendo saber o número de operações realizadas numa dada parcela, para cada tipo de operação, num dado intervalo de tempo.
- USBD08 - Como Gestor Agrícola, pretendo saber o fator de produção com mais aplicações na instalação agrícola num dado intervalo de tempo.
- USBD09 - Como Gestor Agrícola, pretendo saber o número de aplicações de cada tipo de fator de produção aplicados na instalação agrícola num dado intervalo de tempo.
- USBD10 - Como Gestor Agrícola, pretendo saber qual a parcela com mais operações de rega num dado intervalo de tempo.

Critério de aceitação das USBD05 a BD10:

- Critério mínimo de aceitação: só serão avaliadas as US em que haja dados que permitam avaliar o seu funcionamento.
- Esperado: demonstrado com dados criados pelo grupo
- Acima do esperado: demonstrado com dados importados do sistema legacy

3.2.2 Sprint 2

- USBD11 - Como Gestor Agrícola, quero registar uma operação de sementeira
- USBD12 - Como Gestor Agrícola, quero registar uma operação de monda
- USBD13 - Como Gestor Agrícola, quero registar uma operação de colheita
- USBD14 - Como Gestor Agrícola, quero registar uma operação de aplicação de fator de produção
- USBD15 - Como Gestor Agrícola, quero registar uma operação de poda
- USBD16 - Como Gestor Agrícola, pretendo obter a lista dos produtos colhidos numa dada parcela, para cada espécie, num dado intervalo de tempo
- USBD17 - Como Gestor Agrícola, pretendo obter a lista dos fatores de produção aplicados numa dada parcela, e respetivas quantidades, para cada tipo de substância componente, num dado intervalo de tempo
- USBD18 - Como Gestor Agrícola, pretendo obter a lista de operações realizadas numa dada parcela, para cada tipo de operação, num dado intervalo de tempo.
- USBD19 - Como Gestor Agrícola, pretendo obter a lista de aplicações de fator de produção aplicados na instalação agrícola, incluindo a parcela e cultura (se aplicável), por tipo de fator de produção, num dado intervalo de tempo.
- USBD20 - Como Gestor Agrícola, pretendo obter os totais de rega mensal de cada parcela, num dado intervalo de tempo.
- USBD21 - Como Product Owner, pretendo que seja atualizado o modelo relacional (nível lógico e físico) à luz dos novos requisitos e user stories.
- USBD22 - Como Product Owner, pretendo que sejam importados os dados atualizados, fornecidos numa folha de cálculo.
- USBD23 - Como Product Owner, pretendo que seja actualizado o dicionário de dados/glossário.

3.2.3 Sprint 3

Pretende-se que o software em desenvolvimento suporte a certificação de uma unidade em modo de produção biológico. Nesse sentido, devem ser incluídos mecanismos de logging ao nível das operações, bem como alguma proteção contra adulteração de alguns registos chave.

- USBD24 Como Gestor Agrícola, pretendo que todos os registos relacionados com operações tenham registado o instante em que foram criados, gerado pelo SGBD.
- USBD25 Como Gestor Agrícola, pretendo que a identificação da operação seja um número sequencial, não gerado automaticamente pelo SGBD, que deve ser gerado no contexto da transação de registo da operação. Se este registo falhar, não deve haver consequências, nomeadamente a existência de "buracos" na numeração.

- USBD26 Como Gestor Agrícola, pretendo que a criação ou alteração de um registo de uma operação deva ser registado num log que indique o instante, tipo de operação e todos os dados relacionados com a operação (e.g. data da operação, parcela, etc.).
- USBD27 Como Gestor Agrícola, pretendo que não seja possível alterar ou apagar os logs.
- USBD28 Como Gestor Agrícola, pretendo que não seja possível apagar operações, mas deve ser possível anular uma operação, ficando isso registado na BD.
- USBD29 Como Product Owner, pretendo que seja atualizado o modelo relacional (nível lógico e físico) à luz dos novos requisitos e user stories.
- USBD30 Como Gestor Agrícola, pretendo anular uma operação que estava prevista e não se realizou ou que foi criada por engano, sabendo que isso só é possível até aos 3 dias seguintes à sua data prevista de execução, se não houver operações posteriores dependentes desta.
- USBD31 Como Gestor Agrícola, pretendo registar uma receita de fertirrega para usar em operações de rega.
- USBD32 Como Gestor Agrícola, pretendo registar uma operação de rega, incluindo a componente de fertirrega (se aplicável).
- USBD33 Como Gestor Agrícola, pretendo obter a lista das culturas com maior consumo de água (rega) para um dado ano civil. O consumo é em minutos e, em caso de empate, devem ser dadas todas essas culturas.
- USBD34 Como Gestor Agrícola, pretendo obter a lista das substâncias de fatores de produção usadas noutros anos civis, mas não usadas no ano civil indicado.

3.3 Estruturas de Informação (ESINF)

A empresa GFH é um operador logístico que faz a distribuição de cabazes com produtos agrícolas em rede. Neste sistema, os produtores, são as entidades que produzem produtos agrícolas e os vendem em cabazes, isto significa que podem existir produtores mais focados em frutas ou hortícolas, os derivados da produção animal (ovos, carne, leite). Um cabaz é conjunto de produtos de um produtor. Os clientes, são as entidades que encomendam cabazes de diferentes produtores. Na prática, um cabaz é corresponde a um encomenda (que pode ser regular) que consiste numa lista de produtos (de um produtor específico) e respectivas quantidades. Os hubs são as instituições/empresas nas quais são entregues cabazes, posteriormente levantados pelos clientes. Os hubs podem ser universidades, hospitais, ginásios e empresas, e têm um período de funcionamento para entrega e levantamento de cabazes (em principio não coincide com o horário de funcionamento da instituição).

A procura agregada é resultado de todas as encomendas colocadas e de forma similar a oferta agregada é o conjunto de cabazes disponibilizados pelos produtores. A GFH é empresa responsável pela distribuição, não é um comercializador de produtos agrícolas, é um distribuidor de cabazes, não fazendo a composição de cabazes.

Localidade	Horário
CT1 ... CT105	9h:00 – 14h:00
CT106 ... CT215	11h:00 – 16h:00
CT216 ... CT323	12h:00 – 17h:00

3.3.1 Sprint 2

Com recurso às classes que implementam a interface Graph no presente projeto pretende-se a criação de um conjunto de classes e testes que permitam gerir uma rede de distribuição de cabazes de produtos agrícolas.

Esta rede é constituída por vários vértices representativos de localidades onde poderão existir hubs de distribuição.

Os cabazes de produtos são transportados em veículos elétricos, explorados pela GFH, com autonomia limitada, ficando disponíveis nos hubs para posterior recolha pelos interessados. A distribuição dos produtos é condicionada ao horário de funcionamento do hub.

Para maior simplicidade assuma que as viaturas têm uma autonomia máxima de A km, deslocando-se a uma velocidade média de V km/h, os carregamentos dos veículos só podem ser feitos nas localidades, nunca no percurso entre localidades, e cada carregamento de bateria dos veículos demora T_c min. Considere ainda que o tempo médio de descarga dos cabazes de produtos nos hubs é de cerca de T_d min.

- USEI01 – Construir a rede de distribuição de cabazes a partir dos ficheiros (distancias_XXX.csv e locais_XXX.csv) com o formato disponibilizado. O grafo deve ser implementado usando a representação mais adequada para realizar de forma eficiente as funcionalidades pretendidas.
- USEI02 – Determinar os vértices ideais para a localização de N hubs de modo a otimizar a rede de distribuição segundo diferentes critérios:
 - influência: vértices com maior grau
 - proximidade: vértices mais próximos dos restantes vértices
 - centralidade: vértices com maior número de caminhos mínimos que passam por eles

Para cada hub o número de colaboradores do hub é igual ao número do identificador da localidade em que o hub reside. Por exemplo um hub na localidade CT149 terá 149 colaboradores. O horário de funcionamento dos hubs deve ser definido de acordo com a tabela:

Critério de Aceitação: Devolver todas as localidades e respetivos critérios ordenado por ordem decrescente de centralidade e influência.

- USEI03 – Dado um veículo, a sua autonomia e atendendo a que os carregamentos só podem ser feitos nas localidades, determinar o percurso mínimo possível entre os dois locais mais afastados da rede de distribuição, indicando o número de paragens necessárias para carregamentos do veículo.

Critério de Aceitação: Devolver o percurso entre os dois locais mais afastados da rede de distribuição: local de origem, os locais de passagem (indicando os locais onde foi feito carregamento da viatura), distância entre os locais do percurso, local destino, distância total do percurso e o número total de carregamentos.

- USEI04 – Determinar a rede que liga todas as localidades com uma distância total mínima.

Critério de Aceitação: Devolver a rede de ligação mínima: locais, distância entre os locais e distância total da rede.

- USEI05 – Considerando a rede de ligação mínima de distribuição de cabazes e dada uma lista de N hubs, dividir a rede em N clusters conexos e o mais coesos/separados possível.

Critério de Aceitação: Devolver a rede de cada cluster e o respetivo coeficiente de coesão/separação (coeficiente de silhouette⁷).

3.3.2 Sprint 3

- USEI06 – Encontrar para um produtor os diferentes percursos que consegue fazer entre um local de origem e um hub limitados pelos Kms de autonomia do seu veículo elétrico, ou seja, não considerando carregamentos no percurso.

Critério de Aceitação: Devolver para cada percurso o local de origem, os locais de passagem (sendo um hub, identificá-lo), distância entre todos os locais do percurso, distância total e tempo total de percurso.

- USEI07 – Encontrar para um produtor que parte de um local origem o percurso de entrega que maximiza o número de hubs pelo qual passa, tendo em consideração o horário de funcionamento de cada hub, o tempo de descarga dos cestos em cada hub, as distâncias a percorrer, a velocidade média do veículo e os tempos de carregamento do veículo.

Critério de Aceitação: Devolver o local de origem do percurso, os locais de passagem (sendo um hub, identificá-lo), hora de chegada no formato hh:mm em todos os locais do percurso (tratando-se de um hub indicar também a hora de partida, no formato hh:mm, após a descarga dos cestos), a distância total do percurso, o número de carregamentos do veículo e o tempo total do circuito (discriminando o tempo afeto aos carregamentos do veículo, ao percurso e ao tempo de descarga dos cestos em cada hub).

- USEI08 – Encontrar para um produtor o circuito de entrega que parte de um local origem, passa por N hubs com maior número de colaboradores uma só vez e volta ao local origem minimizando a distância total percorrida. Considere como número de hubs: 5, 6 e 7.

Critério de Aceitação: Devolver o local de origem do circuito, os locais de passagem (sendo um hub, indicar o respetivo número de colaboradores), a distância entre todos os locais do percurso, a distância total, o número de carregamentos

⁷[https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering))

e o tempo total do circuito (discriminando o tempo afeto aos carregamentos do veículo, ao percurso e ao tempo de descarga dos cestos em cada hub).

- USEI09 – Organizar as localidades do grafo em N clusters que garantam apenas 1 hub por cluster de localidades. Os clusters devem ser obtidos iterativamente através da remoção das ligações com o maior número de caminhos mais curtos entre localidades até ficarem clusters isolados. Não deverá fornecer soluções de clusters de localidades sem o respetivo hub.

Critério de Aceitação: Devolver lista de Hubs e o respetivo conjunto de localidades que pertencem ao cluster desse Hub.

- USEI10 – Considere uma rede constituída por N hubs definidos segundo o critério de centralidade e admita que nesta rede entre dois hubs circula sempre o mesmo veículo com capacidade limitada a X cabazes de produtos. Pretende-se para um hub origem e um hub destino determinar a rede que permita transportar o número máximo de cabazes.

Critério de Aceitação: Devolver a rede máxima de transporte de cabazes entre dois hubs e número máximo de cabazes transportados.

- USEI11 – Como Product Owner pretendo carregar um ficheiro com os horários de funcionamento de uma lista de hubs. Caso os hubs já existam, os horários devem ser redefinidos, caso contrário deve ser emitida uma mensagem de erro. Considere o seguinte exemplo:

CT1, 14:00, 17:00

CT214, 11:00, 15:30

3.4 Física Aplicada (FSIAP)

Para esta tarefa, temos como ponto de partida, uma estrutura já existente, com todas as paredes feitas em granito, e um telhado de duas águas (dupla inclinação) em telha de barro vermelho, com as seguintes dimensões: 6 metros de largura, 8 metros de comprimento e 2.5 metros de altura, na zona mais baixa da parede. Tem uma porta de acesso pequena e uma janela, mas que devem ser alteradas para estarem de acordo com as atuais necessidades. Assim, pretende-se ampliar este espaço, ou estrutura, recorrendo a outros materiais. A estrutura final deve ter as seguintes dimensões, 150m² e telhado com inclinação dupla. A zona nova deverá ter 4 metros de altura, na zona mais baixa da parede. O espaço interior, considerando as duas estruturas, deve ser dividido em 5 zonas (no total das duas estruturas consideradas), e podem/devem suportar diferentes temperaturas. As zonas têm áreas a definir (assim, cada grupo define os espaços e áreas que vai considerar). Considerem-se as seguintes zonas:

- Zona A - zona que contém a porta de acesso, a receção terá um maior contacto direto com o exterior, dado que é a zona preferencial de receção e distribuição para os restantes espaços. E deve permitir ter uma temperatura de 5 °C abaixo da temperatura ambiente considerada (como ponto de partida pode ser de 20 °C, mas no final do projeto devemos ser capazes de considerar este valor dinâmico), com um mínimo 20m².

- Zona B - deve poder manter o seu interior a -5°C , com um mínimo 20m^2 e máximo 35m^2 .
- Zona C - a temperatura interior será de 0°C , com um mínimo 15m^2 .
- Zona D - a temperatura interior deve ser mantida a 7°C , com um mínimo 15m^2 e máximo 35m^2 .
- Zona E - sem ligação interior às restantes, só com ligação direta ao exterior. Deve permitir manter-se a uma temperatura de 5°C abaixo da temperatura exterior que for considerada (esta zona poderá estar associada ao armazenamento de produtos e/ou de excedentes de produção) com um mínimo 40m^2 e máximo 65m^2 .

3.4.1 Sprint 2

Pretende-se elaborar uma estrutura composta por diferentes materiais, com as características que permitam manter as temperaturas indicadas anteriormente. Cada parede, das novas a construir, deve conter três materiais a separar as temperaturas interiores das exteriores. As paredes de granito e o telhado existentes poderão ser complementadas, ou não, com os novos materiais a usar no edificado restante. Situação que vai depender da localização dos espaços, que cada grupo considerará. Este telhado deve ter dupla inclinação mínima, mas que permita o bom escoamento das águas, com o cume ao longo de todo o comprimento.

- USFA01 - Apresente um croqui de uma estrutura, que considere a estrutura inicial e a sua expansão, e as divisões internas:
 - Esta estrutura terá uma porta grande, que possa subir, de dimensões a definir, mas que permita o acesso a um veículo de transporte de mercadorias, tipo furgão de grandes dimensões, e que dará acesso à zona de receção, zona A.
 - A estrutura deve ter ainda uma outra porta de duas folhas, com dimensões a definir, que servirá exclusivamente para acesso à zona de armazenamento de produtos e/ou excedentes, zona E.
 - A estrutura deve ter um mínimo de duas janelas, ambas com dimensões a definir. Uma posicionada na zona de receção, zona A, e a outra na zona de armazenamento, zona E.
 - O interior será dividido em cinco espaços ou zonas, separados fisicamente por paredes e uma porta de acesso ao seu interior. Com exceção da zona de armazenamento, que só terá acesso pelo exterior.
 - A sua disposição, dimensões individuais e portas de acesso são definidas pelos grupos.
- USFA02 - Pretende-se saber qual o conjunto de potenciais materiais a usar nas paredes da estrutura de expansão e paredes interiores, assim como do respetivo telhado.

- Quais as características térmicas e materiais, a usar na constituição das paredes exteriores.
- Quais as características térmicas e materiais, a usar no telhado.
- Quais as características térmicas e materiais, a usar na constituição das portas e janelas.
- USFA03 - Pretende-se saber quais os materiais a usar nas paredes divisórias (interiores) por forma a definir os espaços indicados e para funcionarem às temperaturas indicadas. A disposição dos espaços fica ao critério dos grupos.
 - Quais as características térmicas e materiais, a usar na constituição das paredes interiores.
 - Quais as características térmicas e materiais, a usar na constituição das portas de acesso.
- USFA04 - Com base no kit de hardware fornecido, de LAPR3, realizar leituras nos diferentes meios, com um termopar fornecido.

Como resultado das USs (FSIAP01 a 03) apresentar num documento resumo, a escolha de materiais considerados para as diferentes paredes consideradas, portas, janelas e telhado, e as respetivas características térmicas. Um documento, em formato pdf, com as escolhas consideradas para cada tipo de parede e telhado, assim como para as janelas e portas consideradas.

3.4.2 Sprint 3

Assumindo a estrutura criada anteriormente, com os espaços interiores às temperaturas indicadas, e numa fase inicial em que o ambiente exterior à estrutura tem uma temperatura inicial de 20 °C.

Deve recorrer a uma folha de calculo, para facilitar os cálculos recursivos e equivalentes necessários ao desenvolvimento do projeto. No final do sprint deve ser possível alterar a temperatura exterior para um qualquer valor admissível, e obter os valores solicitados de resistências térmicas e energia de forma imediata.

- USFA05 - Pretende-se saber qual a resistência térmica das paredes, para cada zona ou espaço definido e para cada temperatura de funcionamento. Sendo que cada parede deve conter três materiais nas suas paredes. Um para o material exterior, outro para o material intermédio e outro para o material interior.
 - Para a divisão ou zona B, determinar a resistência térmica, de cada parede e total, com a inclusão da porta de acesso à divisão.
 - Para a divisão ou zona C, determinar a resistência térmica, de cada parede e total, com a inclusão da porta de acesso à divisão.
 - Para a divisão ou zona D, determinar a resistência térmica, de cada parede e total, com a inclusão da porta de acesso à divisão.

- Para a restante estrutura, e que possa envolver as restantes divisões, determinar a resistência térmica, de cada parede e telhado, com a inclusão das portas de acesso à receção e de armazenamento e janelas consideradas, de acordo com a escolha dos materiais realizada.
- USFA06 - Pretende-se saber qual a energia necessária para manter os espaços ou zonas com as temperaturas interiores solicitadas, por cada hora de funcionamento, com uma temperatura exterior, na ordem dos 20 °C.
 - Determinar a energia total a fornecer, à zona B com temperatura interior de funcionamento de -5 °C
 - Determinar a energia total a fornecer, à zona C com temperatura interior de funcionamento de 0 °C
 - Determinar a energia total a fornecer, à zona D com temperatura interior de funcionamento de 7°C
- USFA07 - Pretende-se saber qual a energia total a fornecer, a toda a estrutura, com todas as suas zonas, A, B C, D e E, mantidas às temperaturas indicadas anteriormente.
 - Determinar a energia total a fornecer, a toda a estrutura, com as divisões internas às suas temperaturas de trabalho, admitindo uma temperatura exterior de 20 °C e por hora de funcionamento.
 - Determinar a potência (energética) necessária para manter cada uma das zonas, ou divisões, e toda a estrutura, às temperaturas indicadas e por hora de funcionamento.
- USFA08 - Fazendo uso do dispositivo de hardware disponibilizado, pretende-se determinar quais as temperaturas e humidade registadas nos diferentes meios ambientes criados.
 - Depois de ter o dispositivo completamente funcional, esperar alguns minutos, sensivelmente 5 minutos, com o dispositivo em cima da bancada e sem fontes térmicas perto, registar os valores de temperatura e humidade, no ambiente da sala de aula. Registar pelo menos 15 pares de valores, ao longo de 15 minutos. No fim, fazer a média dos valores obtidos.
 - Repetir o procedimento anterior, para mais duas temperaturas diferentes. Falar com o professor sobre as fontes de temperatura diferentes.

Como resultado das USs (USFA05 a USFA07) apresentar num documento resumo, as energias e potências encontradas nas zonas solicitadas anteriormente.

Apresentar num documento, em formato .pdf, as considerações e um exemplo de cálculo da resistência térmica, da energia, e da potencia, que serviram de suporte às USs (US FA05 a 07).

Apresentar num documento, em formato .xls, ou folha de calculo, os cálculos realizados para todas as zonas solicitadas, realçando as células com os resultados finais solicitados em cada US.

3.5 Laboratório Projecto 3 (LAPR3)

3.5.1 Sprint 1

- USLP01 Como Product Owner, pretendo que seja elaborado o modelo de domínio (nível conceptual). Este modelo será um elemento fundamental de comunicação entre todos os interessados na solução a desenvolver. O diagrama do modelo de domínio é um documento “vivo” e deverá reflectir em cada momento o entendimento partilhado sobre o domínio pelos interessados.

Critérios de aceitação:

- O modelo de dados deve cobrir a atividade da instalação agrícola, ficando excluída a comercialização e distribuição de produtos e a gestão da rega. Apesar do sistema de rega fazer parte da instalação agrícola, não é necessária manter na base de dados a topologia detalhada.
- Esperado: modelo “básico” partilhado no Visual Paradigm ou docs no git (e.g., PlantUML).
- USLP02 – Como Product Owner, pretendo que seja definida uma funcionalidade que consiste em simular um controlador do sistema de rega.

Para tal deve ser consumido um ficheiro de texto com um conjunto de instruções e gerado um plano de rega para 30 dias. O ficheiro de texto deverá ter a seguinte informação e formato:

<Horas de rega>

Nesta linha são definidos as horas que se inicia um ciclo de rega, por exemplo: 8:30 e 17:00, significa que existem dois ciclos de rega diários que se iniciam respectivamente às 8:30 e 17:00.

<Sector, Duração, Regularidade>

Existe uma linha para cada sector a regar. Sector (uma ou mais parcelas) identifica a zona a ser regada (controlado por uma electroválvula); Duração, o tempo em minutos que o sector deve receber rega; e a Regularidade, a fórmula de recorrência que define os dias que o sector deve ser regado T, todos; I, ímpares, P, pares, 3, a cada 3 dias

Exemplo:

8:30, 17:00

A,14,T

B,8,T

C,25,P

D,25,I

E,7,T

F,10,3

Note-se que por questões capacidade, os sectores são regados de forma sequencial. A necessidade de regar em dias pares ou ímpares resulta da necessidade de balancear as necessidades de rega de todos sectores em função da capacidade máxima do sistema.

O controlador de rega, contém um plano de rega para 30 dias a partir da data de criação e em qualquer momento (data/hora) sabe responder se está a regar ou não, e em caso afirmativo qual o sector que está a regar e quantos minutos faltam para terminar.

3.5.2 Sprint 2

- USLP03 - Como Product Owner, pretendo que seja desenvolvida uma funcionalidade que, de forma automática, consuma o plano de rega gerado pelo simulador do controlador desenvolvido na USLP02 e que de forma escalonada após a conclusão da rega em cada sector registre essa operação no caderno de campo. Nota: No desenvolvimento desta US serão intencionalmente ignorados aspectos estudados em Sistemas Distribuídos (como por exemplo tolerância a falhas).
- USLP04 a USLP08 - Como Product Owner, pretendo que seja desenvolvidas funcionalidades que permitam a evocação das USBD11 a USBD15. Não é pretendido o desenvolvimento de uma interface gráfico para a aplicação "Farm Coordinator", esta interface pode ser em modo texto.
- USLP09 Como Product Owner, pretendo que seja actualizado o modelo de domínio (nível conceptual).

3.5.3 Sprint 3

- USLP10 – Como Product Owner, pretendo que seja melhorada a funcionalidade que consiste em simular um controlador do sistema de rega previamente desenvolvida na USLP02. A melhoria consiste modelar aspectos relacionados com a fertirrega.

Para tal deve ser consumido um ficheiro de texto com um conjunto de instruções e gerado um plano de rega para 30 dias. O ficheiro de texto deverá ter a seguinte informação e formato:

<Horas de rega>

Nesta linha são definidos as horas que se inicia um ciclo de rega, por exemplo: 8:30 e 17:00, significa que existem dois ciclos de rega diários que se iniciam respectivamente às 8:30 e 17:00.

<Sector, Duração, Regularidade, [Mix, Recorrencia]>

Existe uma linha para cada sector a regar. Sector (uma ou mais parcelas) identifica a zona a ser regada (controlado por uma electroválvula); Duração, o tempo

em minutos que o sector deve receber rega; e a Regularidade, a fórmula de recorrência que define os dias que o sector deve ser regado T, todos; I, ímpares, P, pares, 3, a cada 3 dias. Opcionalmente a linha pode contar informação sobre a fertirrega a realizar, definida por um mix (composição de factores de produção, e a fórmula de recorrência da sua aplicação, por exemplo 7, significa que será aplicada nos ordinais 1, 8, 15, ... (primeiro, oitavo e décimo quinto) do plano de fertirrega.

Exemplo:

```
8:30, 17:00
A,14,T,mix1,5
B,8,T,
C,25,P,mix2,7
D,25,I
E,7,T,mix1,3
F,10,3
```

Note-se que por questões capacidade, os sectores são regados de forma sequencial. A necessidade de regar em dias pares ou ímpares resulta da necessidade de balancear as necessidades de rega de todos sectores em função da capacidade máxima do sistema.

O controlador de rega, contem um plano de rega para 30 dias a partir da data de criação e em qualquer momento (data/hora) sabe responder se está a regar ou não, e em caso afirmativo qual o sector que está a regar e quantos minutos faltam para terminar.

- USLP11 - Como Product Owner, pretendo que seja melhorada a funcionalidade desenvolvida na USLP03 que, de forma automática, consume o plano de fertirrega gerado pelo simulador do controlador desenvolvido na USLP10 e que de forma escalonada após a conclusão da rega e/ou fertirrega em cada sector registre essa operação no caderno de campo. Nota: No desenvolvimento desta US serão intencionalmente ignorados aspectos estudados em Sistemas Distribuídos (como por exemplo tolerância a falhas).

3.6 Requisitos não funcionais

Esta secção descreve alguns dos requisitos não funcionais que devem ser considerados na implementação do projecto.

- A validação das regras de negócio que devem ser respeitadas aquando do registo e actualização de dados.
- A Base de Dados será o repositório principal de informação do sistema e deverá reflectir a necessária integridade de dados. A informação deverá ser persistida num SGBD remoto.
- Por forma a potenciar a interoperabilidade entre sistemas existentes ou a desenvolver, a aplicação principal será desenvolvida em Java. No entanto alguns

componentes deverão ser desenvolvidos noutras linguagens, a criação e gestão da Base de dados utilizará PL/SQL, e a interacção com estações meteorológicas será desenvolvida em C/Assembly. Uma parte significativa da integração será realizada através de ficheiros.

- A estrutura de classes deve ser concebida por forma a permitir a sua fácil manutenção e adição de novas funcionalidades, de acordo com as melhores práticas de OO.