UNIVERSIDADE
LUSÓFONA

Configurador Software Aviónico e gerador de código fonte

Trabalho Final de curso

Relatório Intercalar 1º Semestre

Ricardo Miguel Costa Lopes**, a21901328, LEI**

Orientador: Daniel Silveira

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

13/07/2024

Direitos de cópia

Agradecimentos

A realização deste trabalho de curso foi enriquecedora e, muitas pessoas contribuíram de forma significativa para a sua execução.

Em primeiro lugar, agradeço ao professor Daniel Silveira, cuja orientação, paciência e disponibilidade foram fundamentais para o desenvolvimento deste projeto.

Por fim, não posso deixar de agradecer á minha família, pelo suporte emocional, compreensão e incentivo constante.

A todos, muito obrigado!

Resumo

Este projeto tem como referência um projeto do ano anterior, que se pretende dar continuidade.

Assim como no ano passado o projeto mantém o objetivo de aprimorar a configuração de sistemas aviónicos espaciais, especificamente no contexto do Integrated Modular Avionic (IMA) e da arquitetura do software Advanced and Intelligent Robotics (AIR) desenvolvido pela empresa Grupo Mecânica e Voo (GMV).

A solução atualmente proposta pretende dar continuidade, efetuado o desenvolvimento de uma Interface Gráfica Avançada (IGA) que redefine os padrões de eficiência, segurança e conformidade na configuração de sistemas espaciais.

Pretende-se que o IGA seja intuitivo e centrado na usabilidade, proporcionando uma experiência eficaz mesmo para utilizadores com conhecimentos técnicos limitados.

A principal funcionalidade da IGA é a geração automática de código-fonte com base nas configurações modeladas na interface. Isso não apenas otimiza o processo de configuração, mas também garante a precisão e a conformidade com as melhores práticas de programação. Para superar os desafios específicos da configuração errónea do Extensible Markup Language (XML), a IGA incorpora ferramentas de validação automática, reduzindo significativamente o risco de erros e acelerando o desenvolvimento.

Além disso, a IGA ira ser desenvolvida em conformidade com rigorosos padrões de segurança espacial, garantindo que o XML gerado corresponde aos requisitos críticos para a arquitetura. A proposta da IGA não é apenas uma solução técnica, mas uma inovação que responde aos desafios identificados e posicionando-se como uma ferramenta essencial para o desenvolvimento eficiente e confiável de sistemas espaciais.

Este relatório fornece uma visão detalhada do problema, da solução proposta e realizada e dos benefícios do projeto, destacando a importância da IGA.

Palavras chave:

GMV - Grupo Mecânica e Voo (GMV)

AIR - Advanced and Intelligent Robotics

XML - Extensible Markup Language

IMA - Integrated Modular Avionics

IGA - interface gráfica avançada.

Abstract

This project builds upon a reference from a previous year's project, aiming to continue its development.

As in the previous year, the project retains its goal of enhancing the configuration of space avionics systems, specifically within the context of Integrated Modular Avionics (IMA) and the AIR software architecture developed by the company GMV.

The proposed solution seeks to advance this goal by developing an Advanced Graphical Interface (IGA) that redefines standards for efficiency, safety, and compliance in the configuration of space systems.
The IGA is designed to be intuitive and usability-centered, providing an effective experience even for users with limited technical expertise.

The primary functionality of the IGA is the automatic generation of source code based on configurations modeled within the interface. This not only optimizes the configuration process but also ensures accuracy and compliance with programming best practices. To address the specific challenges of erroneous XML configuration, the IGA incorporates automated validation tools, significantly reducing the risk of errors and accelerating development.

Furthermore, the IGA will be developed in compliance with stringent space safety standards, ensuring that the generated XML aligns with critical requirements for the architecture. The IGA is not just a technical solution but an innovation addressing identified challenges and positioning itself as an essential tool for the efficient and reliable development of space systems.

This report provides a detailed overview of the problem, the proposed and implemented solution, and the project's benefits, emphasizing the importance of the IGA.

Key-words:

GMV - Grupo Mecanica e Voo (GMV)

AIR - Advanced and Intelligent Robotics

XML - Extensible Markup Language

IMA - Integrated Modular Avionics

AGI - Artificial general intelligence.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas

API Interface de Programação de Aplicações

GMV Grupo Mecanica e Voo

XML Extensible Markup Language

IMA Integrated Modular

IGA interface gráfica avançada.

1 Introdução

1.1 Enquadramento

O IMA[1] (Integrated Modular Avionics) é um conceito que consiste em centralizar a capacidade de processamento numa única unidade, utilizando a virtualização para separar as funções de software umas das outras, bem como do hardware. A AEEC[2] i (Airlines Electronic Engineering Committee) publicou uma série de normas que definem esta nova arquitetura e as suas interfaces.

O padrão mais relevante para este trabalho é o ARINC[3] 653, que descreve múltiplas propriedades do sistema operativo (OS) executado nas unidades Reparáveis em Linha (LRUs), que agora hospedam diversas funções aviónicas, bem como a aplicação executiva (APEX), uma interface padronizada que permite o desenvolvimento independente do software das aplicações.

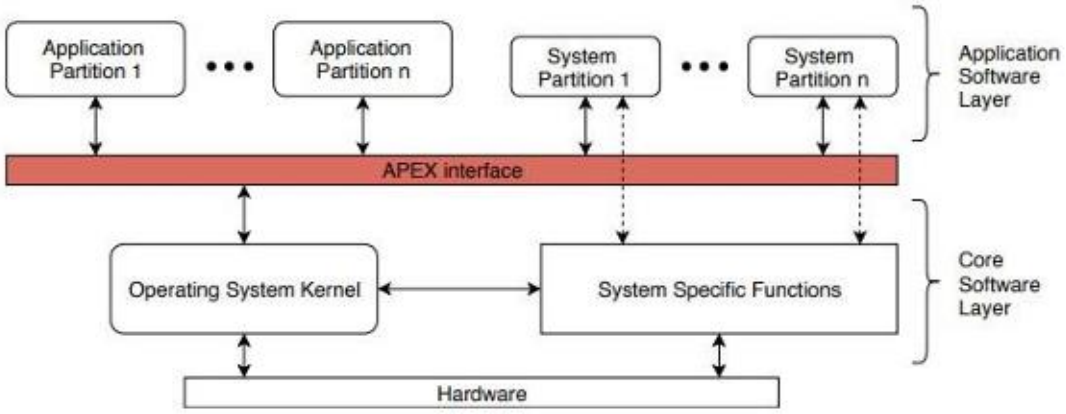


Figura 1 IMA Architecture

1.2 Motivação e Identificação do Problema

Interesse em realizar projetos que envolvam interfaces gráficas, onde a criatividade, a usabilidade, a transversalidade, aliada ao desempenho e arquitetura o tornam um completo para completar a minha aprendizagem.

No decorrer da pesquiza foi tanto verificado com um problema com a leitura XML assim como muitos destes programas não tem:

Compatibilidade de software e hardware: garantir que funcione bem com outras soluções.

[1] IMA - https://en.wikipedia.org/wiki/Integrated_modular_avionics

[2] AEEC - https://aviation-ia.sae-itc.com/activities/aeec

[3] ARINC - https://en.wikipedia.org/wiki/ARINC

Contabilidade de plataformas: maioria das soluções só funciona em Linux limitando assim o número de equipamentos existentes.

1.3 Objetivos

Fundamentalmente pretende-se criar uma interface gráfica que permita efetuar a criação e manutenção evolutiva de uma configuração de um sistema aviónico. Estas configurações são em formato XML, a interface gráfica tem como objetivos:

- Conseguir que este seja flexível.
- Conseguir que este seja configurável.
- Conseguir que este seja universal.
- Deve ser capaz de ser utilizado em dispositivos mobile.
- Deve conseguir der multiplataforma.

1.4 Estrutura do Documento

Pertinência e Viabilidade

1. **Pertinência –** Demostração que o trabalho tem um impacto positivo e resolve os problemas indicados.
2. **Viabilidade –** medida em que a solução é aplicada
3. **Análise Comparativa com Soluções Existentes –** analise comparativa de soluções que já existem.
4. **Proposta de inovação e mais-valias –** informações que mostram que esta proposta de trabalho é inovadora
5. **Identificação de oportunidade de negócio –** propostas de negócio para a exploração comercial.

Especificação e Modelação

1. **Análise de Requisitos** – apenas contem o ponto 3.1.1 pois este contem uma tabela com a enumeração dos requisitos assim como sua descrição 3.1.3 fica como 3.1.2.
2. **Modelação –** imagem do diagrama de entidade-relação.
3. **Protótipos de Interface** – imagem de um desenho de um protótipo.

Solução Desenvolvida

1. **Introdução –** uma introdução breve sobre a solução apresentada.
2. **Arquitetura –** descrição funcional da proposta de trabalho.
3. **Tecnologias e Ferramentas Utilizadas** – Ferramentas que são utilizadas para a elaboração do projeto.
4. **Ambientes de Teste e de Produção** – ainda não esta desenvolvido
5. **Abrangência** – unidades curriculares que são importantes para a realização deste projeto.
6. **Componentes** – Componentes principais do trabalho
7. **Interfaces** – ainda não desenvolvido.

Testes e Validação – ainda não foram efetuados testes

Método e Planeamento

1. **Planeamento inicial** – planeamento do projeto com as datas e previsões de cada etapa de elaboração
2. **Análise Crítica ao Planeamento**

Resultados – ainda não desenvolvido

Conclusão – ainda não desenvolvido

2 Pertinência e Viabilidade

2.1 Pertinência

Fundamentalmente pretende-se criar uma interface que permita efetuar a criação e manutenção evolutiva de uma configuração em formato XML, que permite validar os dados inseridos e gerar código para ser executado por um sistema computacional. Isto significa que a plataforma a construir, tem de ser flexível, configurável e multiplataforma, para poder ser utilizado sem depender do sistema do sistema operativo onde vai ser instalado e sem depender de técnicos especializados para a introdução de novos parâmetros, para que atualmente é pedido. Tem de ser acessível em qualquer dispositivo eletrónico e em qualquer local desde que existam meios de comunicação. A capacidade de configuração permitirá substituir campos em texto por parametrizações que vão diminuir consideravelmente a geração de erros nas configurações finais.

Perante o explicitado, considera-se existir pertinência na sua execução.

2.2 Viabilidade

O movimento NewSpace[4] representa uma nova era na indústria espacial, marcada por empresas privadas, algumas reconhecidas mediaticamente (Blue Origin[5] e SapceX[6]), empenhadas em reduzir os custos de acesso ao espaço e tornar a exploração espacial mais acessível e comercialmente viável. Entre os seus principais princípios e características destacam-se:

- Empreendedorismo: Empresas privadas e Start-ups estão a desempenhar um papel fundamental na condução da inovação e na criação de soluções espaciais.
- Redução de custos: uma ênfase na redução de custos e eficiência, torna a exploração espacial mais acessível.
- Inovação tecnológica: A adoção de tecnologias inovadoras, como foguetes reutilizáveis, impressão 3D de componentes espaciais e novos sistemas de propulsão.
- Acessibilidade espacial: tornar o espaço acessível a uma variedade de empresas e até mesmo a indivíduos para fins comerciais, de pesquisa e exploração.

A capacidade de criar uma configuração num determinado formato universal, sem ambiguidades, com capacidade de introduzir novos parâmetros sem necessidade de reprogramar, que tenham uma representação numérica inequívoca, multiplataforma e multicanal, torna a mesma viável.

[4] NewSpace - https://www.newspaceportugal.org/

[5] Blue Origin – https://www.blueorigin.com/

[6] SapceX – http://spacex.com/

2.3 Análise Comparativa com Soluções Existentes

No contexto da arquitetura Integrated Modular Avionic (IMA) e sua expansão para a indústria aviónica espacial, é imperativo realizar uma análise\comparação abrangente dos produtos (Benchmarking), nomeadamente o PikeOS, XtratuM, VxWorks, RT-Xen e especificamente o AIR, destacado pela sua natureza Open Source.

2.3.1 Algumas das Soluções existentes

- PikeOS[7] - O sistema operacional PikeOS é um sistema de virtualização leve que é capaz de hospedar quase todos os sistemas operativos padrão da indústria e ambientes de execução. Suporta os mais rigorosos testes em tempo real, multiprocessador, estando associado às indústrias onde a segurança dos sistemas é crítica.

- XtratuM [8]- Tem a capacidade de isolar aplicativos de criticidade mista, executados na mesma plataforma de hardware. Em simultâneo, aplicações críticas e não críticas são executadas no mesmo computador em contextos separados. Oferece suporte a diversos sistemas operacionais, como RTEMS, LithOS/ARINC-653 e Linux. Definição estática do sistema via arquivo de configuração (XML).

- VxWorks [9]– VxWorks é um sistema operativo de tempo real (ou RTOS). Foi projetado para uso em sistemas embebidos, que exigem desempenho determinístico em tempo real, sendo que, em muitos casos, para desempenhar tarefas de alta segurança e proteção para indústrias como aeroespacial, defesa, dispositivos médicos, equipamentos industriais, robótica, energia, transporte, redes, automotiva e eletrónica de consumo.

- RT-Xen baseado no Xen[10] - é um software livre de para sistemas complexos em tempo real, em ambientes virtualizados. O RT-Xen, a primeira estrutura de agendamento virtual em tempo real para o Xen, o monitor de máquina virtual (VMM) de código aberto mais utilizado. O RT-Xen é desenvolvido numa plataforma de código aberto para investigadores e integradores desenvolverem

[7] PikeOS - https://en.wikipedia.org/wiki/PikeOS

[8] XtratuM - https://en.wikipedia.org/wiki/XtratuM

[9] VxWorks - https://pt.wikipedia.org/wiki/VxWorks

[10] RT-Xen baseado no Xen - https://sites.google.com/site/realtimexen/

e avaliarem técnicas de escalonamento em tempo real, que até o momento foram estudadas predominantemente por meio de análises e simulações. Diversos resultados experimentais demonstram a viabilidade, eficiência e eficácia do agendamento hierárquico em tempo real de prioridade fixa. O RT-Xen pode fornecer agendamento eficaz em tempo real para sistemas operativos Linux convidados, ao mesmo tempo em que incorre em sobrecarga apenas moderada para todos os algoritmos de servidor de prioridade fixa.

- RTEMS (AIR) [11]- Arquitetura de virtualização que suporta a execução de tarefas críticas em tempo real. Tal como o XtratuM, foi um projeto iniciado pela ESA (European Space Agency) com o intuito de alocar partições de espaço e tempo definidos para pacotes de software em sistemas espaciais. O AIR é uma prova de conceito de arquitetura para se conceber o sistema ARINC-653 especificamente desenhado para o domínio espacial (Rufino, Craveiro, Schoofs, Tatibana, & Windsor, 2009).

  A partir de 2014, o AIR é referenciado como um produto de código aberto oferecido por empresa internacional de aeronáutica, GMV. Segundo a GMV, a AIR está atualmente no TRL nível 5, o que significa que o sistema foi testado e protótipado em um nível elevado.

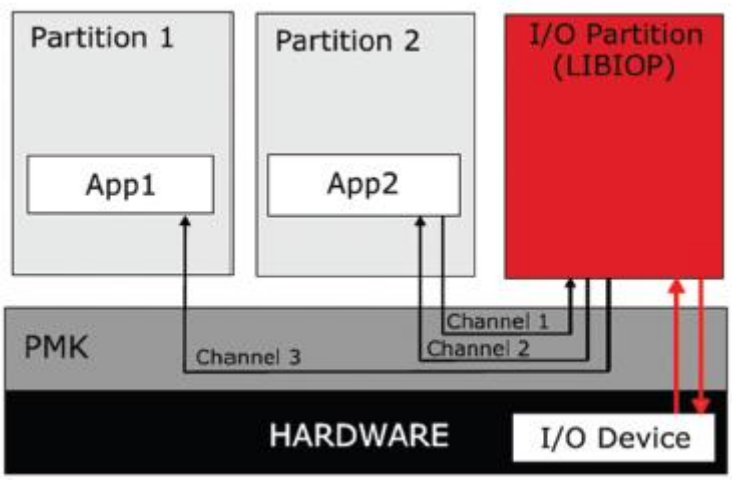


Figura 2 Aplicação AIR

Para além de permitir a sua configuração através de ficheiro de código universal, Standard na Indústria (XML), trata-se de um software OpenSource com licença GPL v2 (bastando enviar um email para a empresa GMV, formalizando a razão para aceder ao código fonte), sendo independente em termos do hardware e software utilizado.

[11]RTEMS(AIR)

- https://indico.esa.int/event/225/contributions/4307/attachments/3343/5387/OBDP2019-paper-GMV_Gomes_AIR_Hypervisor_using_RTEMS_SMP.pdf

2.3.2 Análise de benchmarking

Da Análise efetuada, o AIR é sem dúvida um pacote de software que permite a investigação, pela faculdade Open Source, tem avaliações e testes consistentes, com uma empresa multinacional e uma agência espacial que se baseiam no mesmo para evoluírem as suas soluções. Aproveita o conceito de outras soluções, de configuração por ficheiro de parametrização, podendo assim ser flexível, acompanhando a sua flexibilidade em termos de hardware e software utilizado.

Hypervisor	License	Internal Design	Development Tools	Documentation	Hardware Support	API and Guests supported	Standards	Footprint (kernel)	Performance Evaluation	Space use status
INTEGRITY Multivisor	Proprietary	Security Kernel	Wind River Workbench	Unavailable openly	(see INTEGRITY RTOS)	All guests (designed to be OS agnostic)	DO-178B, ARINC-653, EAL 6+	Unknown	No	No
VxWorks hypervisor	Proprietary	Configurable	Yes	Unavailable openly	(see VxWorks Hypervisor)	All guests (designed to be OS agnostic)	None	Depends, highly modular	No	No
XtratuM	Open-source GPL or proprietary	Monolithic kernel	No	Yes	X86, ARM, PowerPC	LithOS, paRTiKle, Linux, RTEMS	Unknown	10K lines of code	ESA	ESA
ARLX	Permissive after subscription	Xen-based	No	Some	ARM, x86	All guests supported by Xen	DO-178C	~70K	Yes	Yes
PikeOS	proprietary	Microkernel	Yes	Some	X86, MIPS, PowerPC, ARM, SPARC V8/LEON	Linux; RTEMS; POSIX, Ada	DO-178B, MILS and ARINC-653	Unknown	NASA	NASA
AIR	Open-source	Microkernel	Unknown	No	All	All guests(designed to support most OSes)	ARINC-653	Unknown	Yes (ESA) Current status unknown	Unclear
NOVA	Open-source	Separation kernel	No	Yes	X86	All guests (via emulation)	None	9k lines of code	No	No
X-hyp	proprietary	Unknown	Unknown	No	ARM-9, Cortex	FreeRTOS, Linux, RTEMS	None	Unknown	No	No
Proteus	Unknown		No	No	PowerPC	All guests (via full virtualization)	None	15 Kb	No	No
RT-Xen	Open-source	Xen-based	No	No	All Xen	Linux guests (unspecified versions)	None	Unknown	No	No

Figura 3 Resumo do Atributo Chave da Arquitetura de Virtualização

2.4 Proposta de inovação e mais-valias

O trabalho a executar irá ter uma arquitetura diferente da do trabalho do ano passado, maior possibilidade de processamento (por otimização ou por portabilidade para sistemas com maior capacidade), portabilidade, universalidade, multiplataforma e multicanal, destacando-se a capacidade de ser utilizado em dispositivos mobile, por ser completamente Web base.

Tabela 1 tabela de inovações e diferenças

AIR Anterior	Inovações
• Apenas funciona em Linux • Apenas pode ser utilizado em computadores. • Base de dados proprietária. • Não consegue validar dados, apresentando muitos parâmetros em formato de texto.	• Consegue ser usado sistemas Linux, IOS,Android e Microsoft. • Interface adaptável a qualquer dispositivo (PC, Tablet, Telemóvel). • A base de dados Opensource, multiplataforma. • Com validação de dados e a maioria dos parâmetros tem uma representação numérica.

	• Com controle de login, sessão, perfil e módulo de auditoria.

2.5 Identificação de oportunidade de negócio

A integração de open source com as soluções do GMV AIR pode abrir oportunidades de negócio significativas em Portugal, dado o crescente interesse por tecnologias acessíveis, colaborativas e sustentáveis. O uso de software ou hardware open source permite acelerar inovações, reduzir custos e fomentar parcerias com a comunidade tecnológica local, universidades e empresas. Aqui estão algumas áreas de destaque:

O GMV AIR poderia adotar plataformas open source para robótica, como o ROS (Robot Operating System), permitindo o desenvolvimento de soluções personalizadas para diferentes setores:

- **Indústria e Manufatura**: Automatização de processos com robôs colaborativos (cobots).
- **Agricultura de Precisão**: Implementação de robôs e drones para monitorar plantações e otimizar recursos.
- **Exploração Espacial**: fomentar a inovação em robótica espacial com colaborações em código aberto.

Isto também assim permite que haja:

Licenciamento de componentes adicionais ou serviços premium.

Suporte técnico e consultoria especializada.

Parcerias com universidades e empresas para desenvolvimento colaborativo.

3 Especificação e Modelação

3.1 Análise de Requisitos

3.1.1 Enumeração de Requisitos

Tabela 2 tabela de requisitos

ID	Título	Obrigatório	Descrição
1 - Criação de arquiteturas AIR			
1.001	A leitura de um XML	Sim	O programa deve conseguir ler XML
1.002	Transferir informação da BD	Sim	O programa deve facilmente transferir informação da BD
1.003	O Export da arquitetura para um XML	Sim	O programa deve conseguir exportar o XML depois de ter sido objeto de manutenção evolutiva.
1.004	Criação de arquiteturas	Sim	Deve ser possível a criação de arquiteturas no programa e sua personalização.
2 - Time Schedule e Propriedades.			
2.001	A possibilidade da realização de um time Schedule	Sim	O programa deve permitir a realização de um time Schedule de forma correta.
2.002	Alteração do time Schedule	Sim	Deve ser possível a realização da alteração do Time Schedule depois deste ter sido já finalizado.
3 - Pagina inicial.			
3.001	A visualização da altura de criação	Sim	Deve mostrar no écran a mensagem de criação com sucesso ou razão do insucesso.
3.002	A visualização da altura em que a arquitetura foi alterada	Sim	Deve mostrar no écran o resultado da alteração de uma determinada arquitetura.
3.003	A ordenação de arquiteturas	Sim	Deve mostrar no écran as diferentes arquiteturas de forma ordenada da mais recente para a mais antiga.
4 - Utilização humano maquina.			
4.001	Apresentar uma arquitetura user friendly	Sim	Permitir configurações dos parâmetros, controlo de perfis, facilidade de criação\ alteração\importação\exportação com menus, importação. Acessível de qualquer ponto e em qualquer sistema operativo.

3.1.2 Casos de Uso/*User Stories*

- **Como um utilizador quero conseguir ler o XML**
- **Como utilizador quero exportar a arquitetura para um XML**
- **Como utilizador quero criar uma arquitetura**
- **Como utilizador quero que seja possível a realização de um time Schedule**
- **Como utilizador quero alterar o time Schedule**
- **Como utilizador quero que esteja visível a altura de criação da arquitetura**
- **Como utilizador quero que esteja visível a altura em que a arquitetura foi alterada**
- **Como utilizador quero que as arquiteturas estejam ordenadas**
- **Como utilizador quero que a interface seja de fácil perceção**

3.2 Modelação

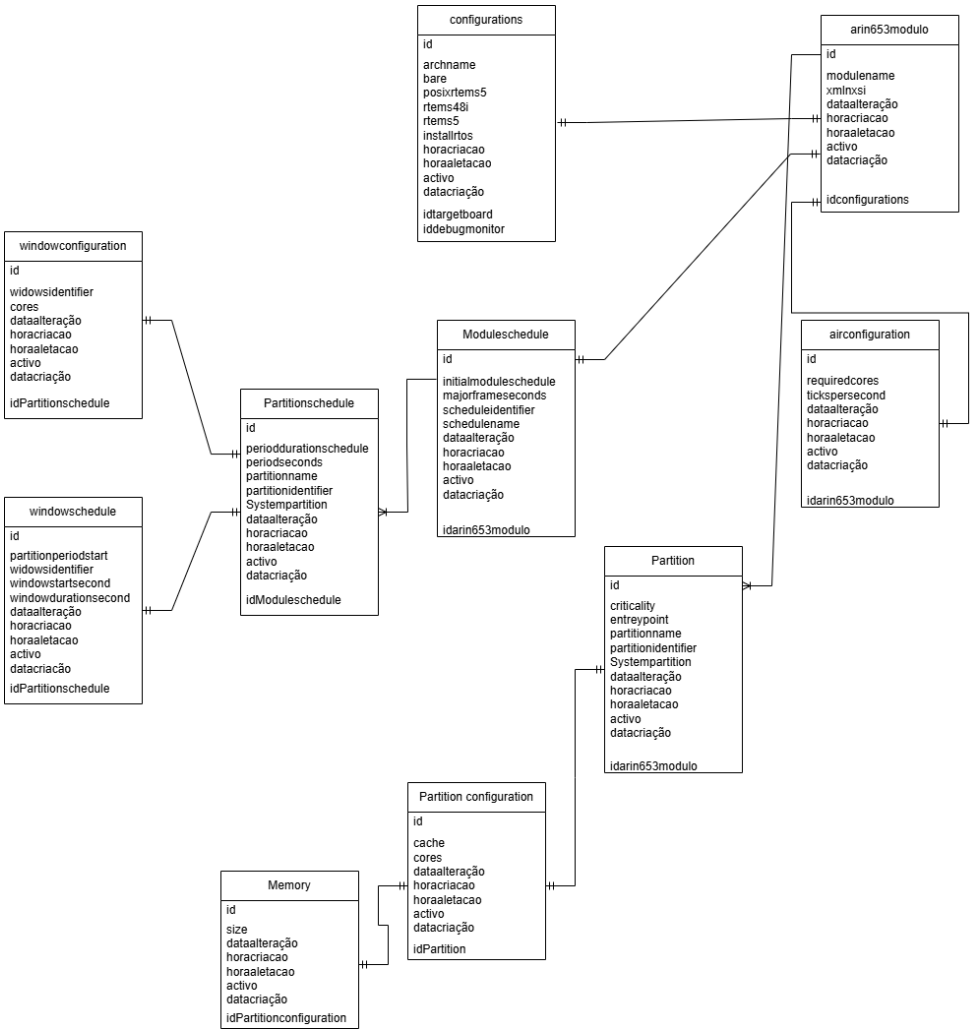


Figura 4 Diagrama de entidade-relação

3.3 Protótipos de Interface

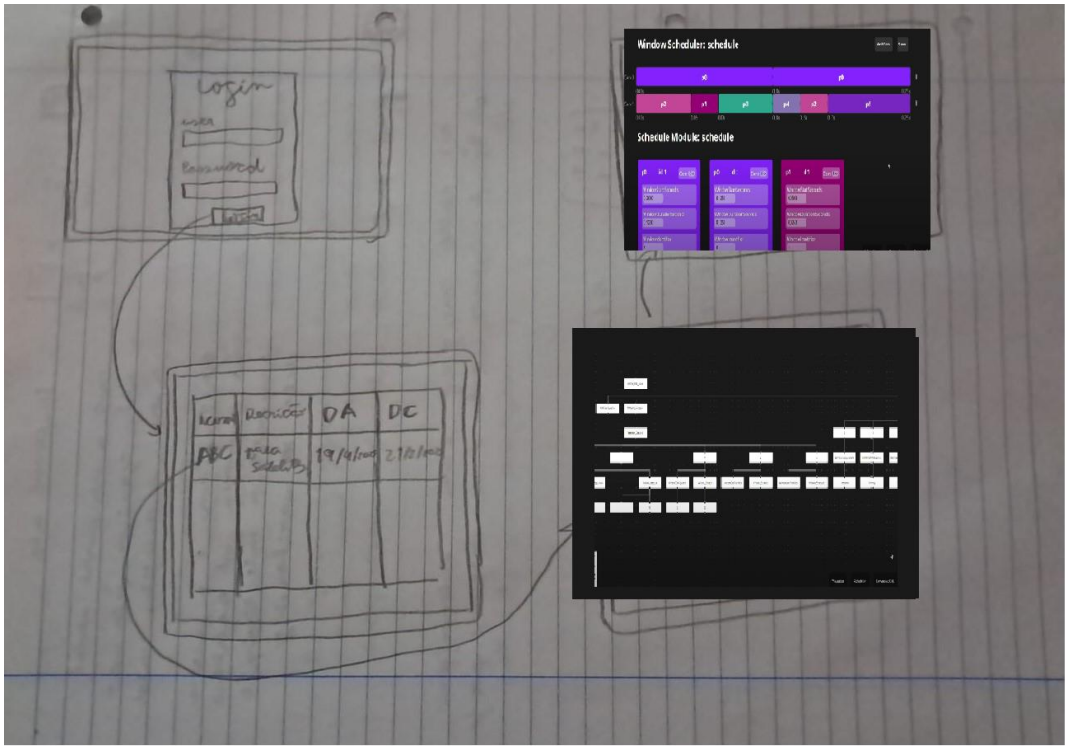


Figura 5 Protótipo da Interface

Nesta primeira fase está pensado a existência de 2 Web Windows com design apelativo. Uma com a função de login, representado o Front Office. E outra só acedida após login válido, que representa o sistema de BackOffice onde todas as atividades de utilização são desenrolar e que permitiram criar\alterar\importar e exportar o XML de configuração.

4 Solução Desenvolvida

4.1 Introdução

Neste trabalho que foi realizado a partir de outro já existente foram identificadas várias melhorias em relação à arquitetura. Estas melhorias pretendem dar continuidade ao trabalho desenvolvido e introduzir maior eficiência e qualidade no mesmo.

4.2 Arquitetura

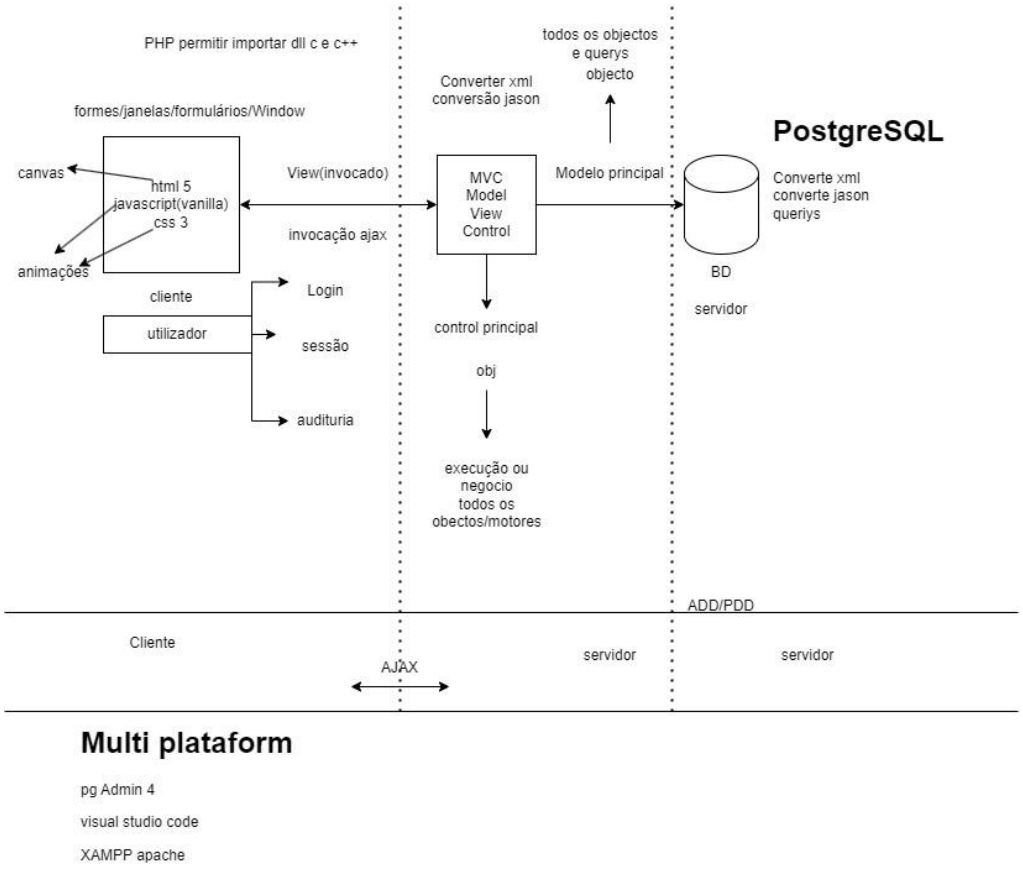


Figura 6 Modelo MVC

Utilizando um modelo MVC (Model, View e Controller), pela sua capacidade de abstração nos diversos níveis, permitirá maior eficiência na manutenção do código. Procura-se igualmente que o programa fique universal podendo ser acedido por todo o tipo de sistemas operativos, mas também facilitar a troca de informações entre a interface e o utilizador.

Este modelo já muito utilizado em termos empresariais, é constituído por três blocos principais que são: o MVC). No Model estarão contidos os dados do programa, pelo que, estamos a falar da constituição da base de dados. O Controller será a zona que irá controlar todo o funcionamento do programa, efetuando pedidos de informação ao Bloco Model, para obtenção de dados e efetuando pedidos o bloco de View para geração das interfaces (páginas Web) onde o utilizador interage.

4.3 Tecnologias e Ferramentas Utilizadas

As ferramentas que foram selecionadas das para a realização do trabalho pretendido são as seguintes:

- **PgAdmin[12]:** Utilizado para a gestão, desenho e testes da base de dados Postgresql e todas as querys (encapsuladas em Store Procedures) necessárias para que a camada Controller consiga interagir com a base de dados.
- **Xampp Apache**[13]: Pacote de Multisoftware que proporciona a experiência de ter seu próprio mini servidor web em casa ou em alojamento, compatível com ambientes Windows (WAMP) e Linux (LAMP) e com as últimas versões do PHP.
- **Visual Studio Code[14]:** GUI de desenvolvimento de software multilinguagem com capacidade de adicionar peças de software que evoluem a sua capacidade, tornando-se uma ferramenta fundamental no desenvolvimento de software Web, nas valências de cliente e servidor. Aqui serão desenvolvidas ás várias camadas de software com as seguintes tecnologias:
  - CSS3;
  - Html5;
  - Vanilla Javascript com componente Ajax;
  - PHP 8.2;

4.4 Ambientes de Teste e de Produção

4.5 Abrangência

No decurso do trabalho proposto iram se destacar várias unidades curriculares que serão importantes á realização do mesmo das quais destaco:

- **Base dados:** Arquitetura, desenho e gestão de base de dados procurando que a informação a introduzir seja consistente, sem ambiguidades, controlada por mecanismos associados à conceção e inter-relação.
- **Interação humano máquina:** Procurar que a interface consiga ser mais intuitivo possível para o utilizador, eficiente e configurável, adaptando-se às novas necessidades sem necessidade de intervenção técnica.
- **Programação na vertente Web:** Criação de uma plataforma baseada num modelo de abstração, multicamada virtual, arquitetura Cliente-Servidor, com possibilidade do motor de Base de dados estar num servidor diferente, se no futuro assim o pretenderem transformar, bastando apenas introduzir alterações na configuração, no objeto\método de comunicação à Base dados.

[12] **PgAdmin -** https://www.pgadmin.org/

[13] **Xamp Apache -** https://www.apachefriends.org/

[14] **Visual Studio Code** - https://code.visualstudio.com/

4.6 Componentes

4.6.1 Componente 1: Model

O objeto Model será o responsável por obter, enviar o alterar informação contida na Base de dados, invocando funções ou procedimentos com validação.

4.6.2 Componente 2: Controller

O objeto Controller será a zona que irá controlar todo o funcionamento do programa, efetuando pedidos ao Bloco Model após validação da informação que chega do cliente, para interação com a base de dados e efetuando pedidos o bloco de View e retornar para o cliente as respostas necessárias, de forma eficiente.

4.6.3 Componente 3: View

O componente View será para a geração das interfaces (páginas Web) onde o utilizador interage.

4.6.4 Componente 4: Objetos de gestão de informação servidor e cliente

Estes dois componentes de gestão de dados, um do lado do cliente e outro do lado do servidor, que guardam informação, evitando acessos repetitivos à base de dados. Sendo que o objeto servidor efetua atualizações á informação se existirem alterações a determinada informação na Bases de dados e o objeto cliente atualiza a interface de forma assíncrona, de acordo com as atualizações vão sendo efetuadas na base de dados e que afetam determinados objetos visuais das várias janelas (ex.: Dropdows). Como é sabido, a interface permite a sua autoconfiguração, tornando-o flexível na introdução de nova informação. Este será a principal inovação, que sendo explorada em futuras evoluções de outros colegas de curso, poderá permitir a geração de um XML completamente adaptável. Os dados enviados para o cliente são em formato JSon (JavaScript Object Notation).

4.6.5 Componente 5: Objetos de comunicação cliente-servidor

Estes dois componentes estabelecem a comunicação entre o cliente e o servidor, um do lado do cliente e outro do lado do servidor.

4.6.6 Componente 6: Objeto CSS

Objeto onde estão formadas todas as regras dos componentes visuais e adaptabilidade a qualquer dispositivo eletrónico em termos de dimensão do ecrã.

4.7 Interfaces

A primeira pagina da interface Web (Front-office), permite que possa ser chamado o form de login, mas poderiam ser chamados outros forms, ou existir informação pública, isto é, de consulta livre. Para carregar a página de login, basta clicar na imagem, no canto superior direito.

Figura 7 Ecrã inicial

Figura 8 Ecrã inicial com login

Depois de se efetuar login, é guardada a sessão e utilizador, para controlo, pois a sessão expira após 30 minutos de utilização, garantindo a segurança da utilização do BackOffice. Esta segunda interface (de BackOffice) é responsável por garantir a integração de um menu, e de todos os forms que são disponibilizados a partir do menu, que é criado, de acordo com o perfil do utilizador. Significa que o menu varia a sua apresentação, de acordo com o perfil que foi atribuído ao utilizador. As operações apresentadas a partir do menu, disponibilizam forms para executar as tarefas definidas para este trabalho.

Figura 9 Menu Principal

Este menu esta dividido da forma apresentada:

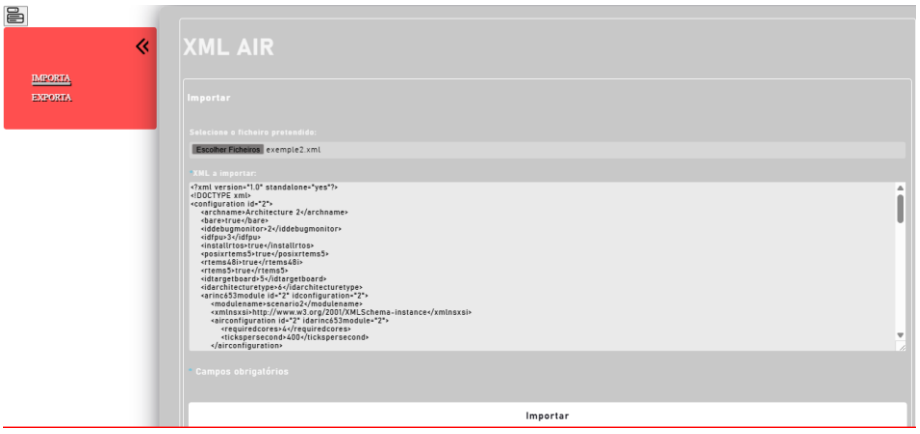
- AIR
  - Configurar – Encontrasse em desenvolvimento o form de configuração da arquitetura XML.
  - Visualizar – Form para o visionamento gráfico da arquitetura. – Ainda em desenvolvimento
- XML AIR
  - Importar – Form para importar xml para a Base dados
  - Exportar – Form para exportar xml da base dados
- Gestão de tabelas AIR – Form para efetuar a alteração e adição das opções apresentados nas dropdowns.
- Gestão de sistemas
  - Perfil – Form para alteração e visualização do perfil, disponível para perfil de administração.
  - Utilizadores – Form para criar e alterar os dados do utilizador, disponível para perfil de administração.
  - Autenticação –para ativar e desativar login de utilizadores, disponível para perfil de administração.
  - 

Figura 10 Importar

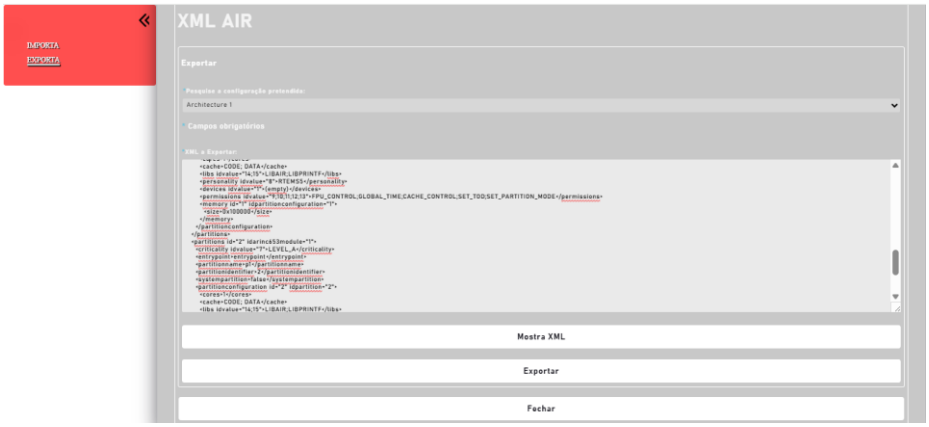


Figura 11 Exportar

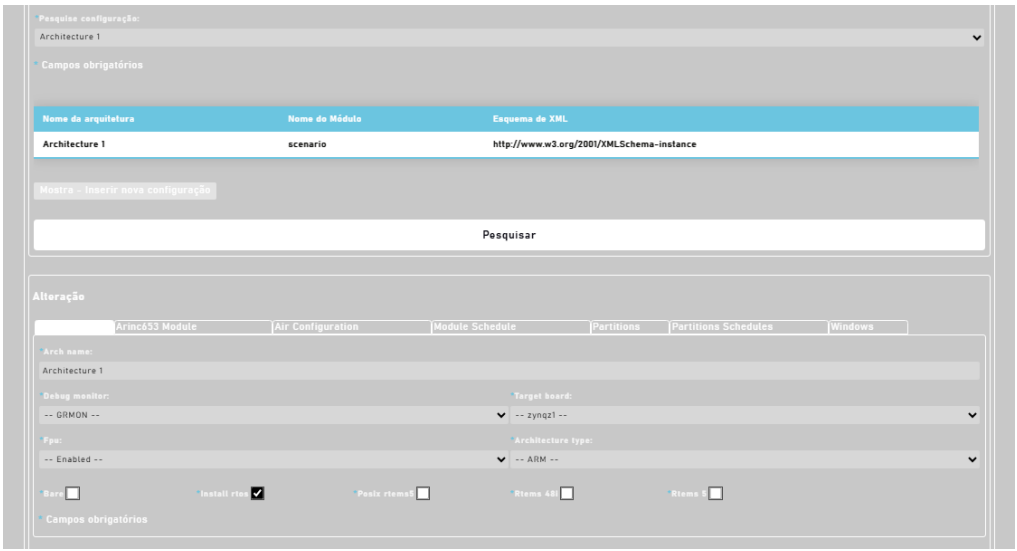


Figura 12 Configurar

Figura 13 Gestão das tabelas AIR

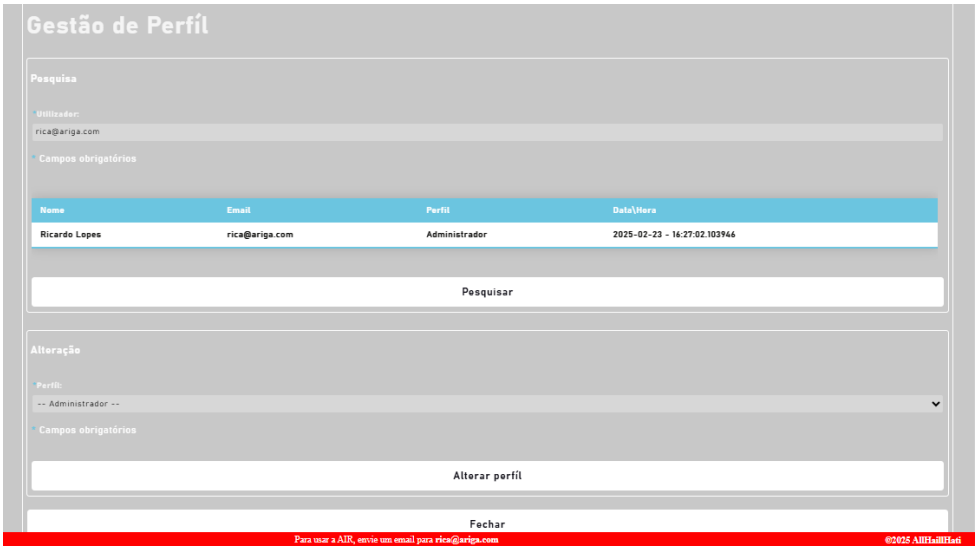


Figura 14 Perfil

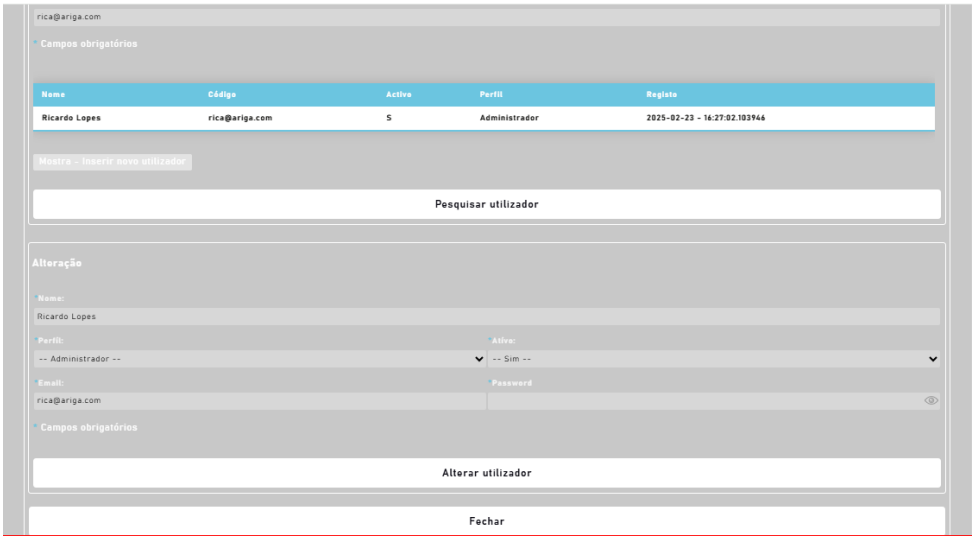


Figura 15 Gestão de utilizadores

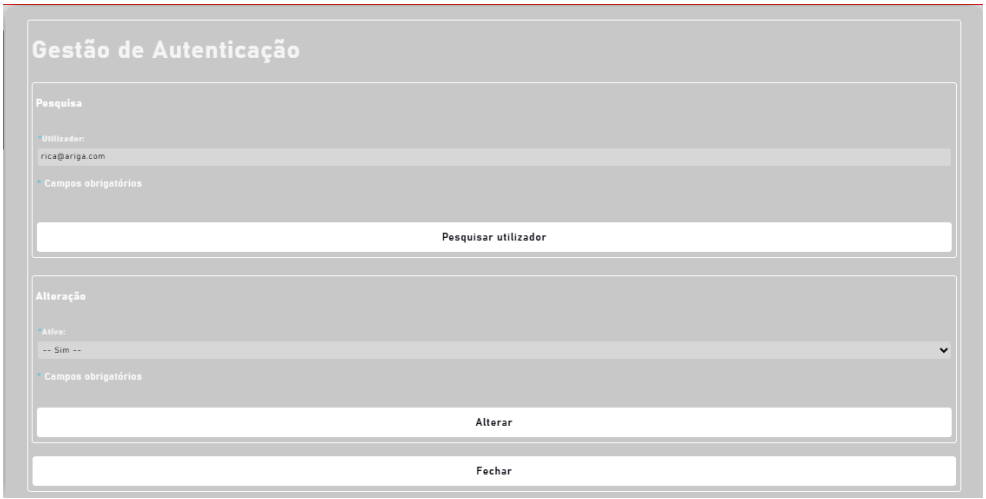


Figura 16 Autenticação

5 Testes e Validação

Para garantir que a solução desenvolvida cumpre os objetivos propostos e resolve problemas reais, foi delineado um plano de testes e validação que abrange vários cenários de uso e critérios de desempenho. Este plano inclui testes funcionais, de desempenho, de usabilidade e de segurança.

5.1 Testes Funcionais

- Objetivo: verificar se todas as funcionalidades principais da IGA operam conforme esperado.
- Método: Testes de violação onde os casos de uso são executados e os resultados observados são comparados com os resultados esperados.
- Cenários de Teste:
    1. Criação\alteração de configuração.
    2. Criar XML, gerado externamente, validando o formato e dados enviados.
    3. Geração de XML com as configurações feitas na interface e guardado na BD.

5.2 Testes de Desempenho

- Objetivo: avaliar o tempo de resposta da aplicação e a eficiência da Interface.
- Método: colocar vários utilizadores a trabalhar na interface e verificar o desempenho das operações associadas à geração do XML.
- Cenários de Teste:
    1. Teste de verificação de tempo de resposta da importação e exportação do XML.
    2. Teste de verificação de tempo de resposta configuração e visualização da arquitetura representada pelo XML.

5.3 Testes de Usabilidade

- Objetivo: garantir que a interface é intuitiva e fácil de usar, mesmo para utilizadores com conhecimentos técnicos limitados.
- Método: Testes de usabilidade com utilizadores reais, observando a interação com a interface e recolhendo feedback.
- Cenários de Teste:
    1. Interação com a interface para criar, configurar, importar e exportar XML.
    2. Navegação e acessibilidade da interface.

5.4 Testes de Segurança

- Objetivo: Assegurar que a interface protege adequadamente os dados e operações contra acessos não autorizados e outras ameaças.

- Método: Testes de penetração e análise de segurança para identificar vulnerabilidades.
- Cenários de Teste:
    1. Testes de autenticação e sessão expirada, de utilização do link apresentado, noutro browser, computador ou sessão.
    2. Análise de vulnerabilidades para identificar possíveis pontos de entrada para ataques.

6 Método e Planeamento

6.1 Planeamento inicial

Descrição do Método de Trabalho Seguido no Desenvolvimento do Projeto O desenvolvimento deste projeto foi estruturado de acordo com um calendário proposto nos relatórios preliminares, o qual delineou duas fases principais: Estudo e Desenvolvimento da Solução.

Cada fase foi cuidadosamente planeada e executada para garantir a entrega de uma Interface Gráfica Avançada (IGA) que atenda aos requisitos definidos e resolva os desafios identificados na configuração de sistemas aviónicos espaciais.

Fase 1: Estudo da solução

Compreender a arquitetura IMA, procurar criar um XML com maior qualidade (reduzir o erro), aplicar o modelo MVC, montar um sistema multiplataforma e de aceso por Multidispositivos.

Atividades:

- Seleção de linguagens de programação cliente e servidor, pertinentes para o desenvolvimento do projeto multiplataforma.
- Análise do trabalho anterior de forma a identificar zonas de melhoria do mesmo.
- Avaliação e escolha de estratégias que possam ser utilizadas na elaboração do projeto.
- Desenho da arquitetura tendo como ponto de referência o modelo MVC.
- Desenho da base de dados forma a que esteja em conformidade ao XML a importar e que permita a parametrização e redução do erro.
- Pensamento conceptual da interface gráfica.
- Pensamento conceptual sobre a forma de comunicação cliente servidor.
- Escolha de motor de base dados e servidor web multiplataforma.

Fase 2: Estruturação do bloco de dados

Objetivo: Adequar a BD de dados para permitir parametrizações, controlo de acessos e auditoria.

Atividades:

- Efetuar alterações na arquitetura de BD, da definição dos campos em termos de tipos de dados e dependência entre tabelas, garantindo a robustez dos dados trabalhados.

Fase 3: Desenvolvimento da Solução

Objetivo: Desenvolver a Interface MultiLayer, multicanal dispositivo e universal.

Atividades:

- Desenvolvimento do nível View;
- Desenvolvimento do nível Model;
- Desenvolvimento do nível Controller;
- Desenvolvimento do nível cliente (JavaScript);
- Desenvolvimento do objeto de comunicação cliente\servidor (Ajax);
- Desenvolvimento do objeto de atualização assíncrona das Views, multiutilizador;
- Desenho da interface gráfica;
- Testes de usabilidade;
- Testes semânticos e sintáticos.

Numa primeira fase, foi usado o método Agile para desenvolvimento do relatório:

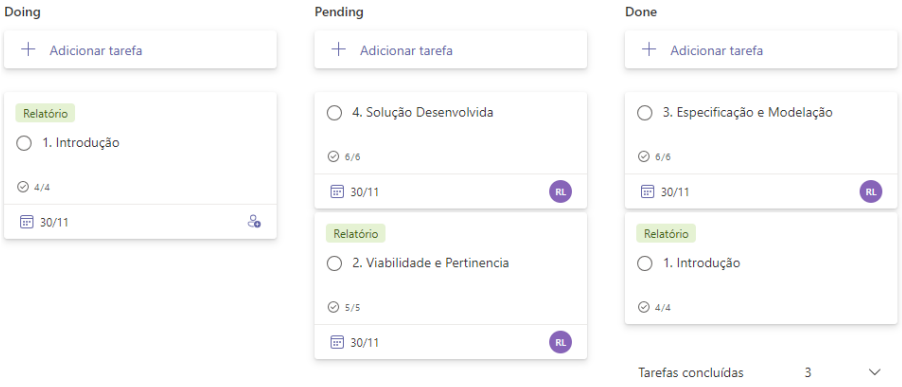


Figura 17 Planeamento em abordagem Agile

Apresenta-se o gráfico Gantt, que demonstra temporalmente as várias fases do projeto. De acordo com a sua evolução poderá ser objeto de adequação\melhoria, com o intuito de se atingir os objetivos definidos.

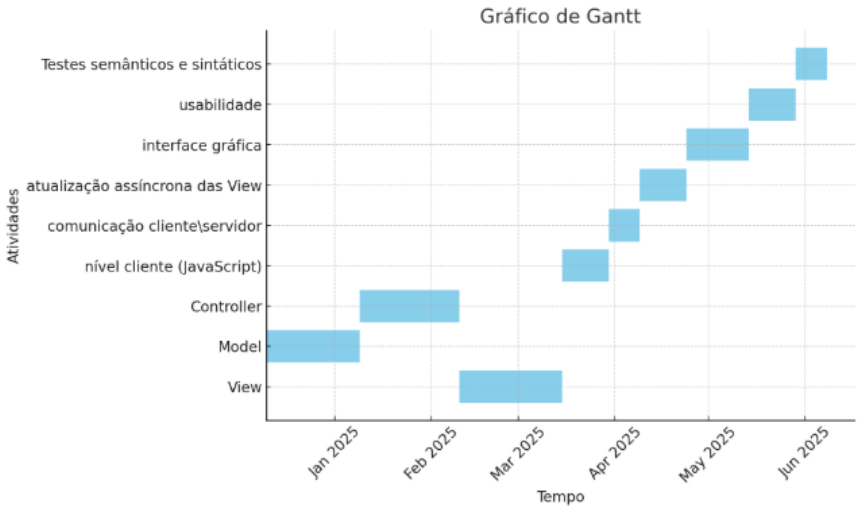


Figura 18 Gráfico de Gantt

6.2 Análise Crítica ao Planeamento

Atualmente, face ao planeado o estudo da solução está concluído, tendo-se desenhado uma arquitetura concreta do que se pretende desenvolver. Foi concluída a arquitetura de base de dados, faltado introduzir uma melhoria a modelo, resultante análise do XML e de discussão da solução com o orientador que me acompanha neste projeto. No entanto, já foram introduzidos dados de teste, verificando o controlo e consistência de dados. Foi efetuado um desenho preliminar do Front-End, mas o desenvolvimento ficou parado, tendo em conta a necessidade de conceber este relatório preliminar. Face ao trabalho a realizar, convém existir um acompanhamento continuo sabendo que existem outras tarefas, que não se limitam a este trabalho.

7 Resultados

7.1 Resultados dos Testes

O projeto culminou na criação de uma Interface Gráfica Avançada (IGA) para a configuração de sistemas aviónicos espaciais, especificamente integrados com o sistema AIR da GMV. Os resultados obtidos foram avaliados em relação aos critérios de sucesso estabelecidos no levantamento de requisitos e mediante a análise dos test cases definidos no segundo relatório intercalar.

7.1.1 Resultados e Outputs

1. Interface Gráfica Avançada (IGA)
    a. Descrição: A IGA desenvolvida permite aos utilizadores criar, modificar e gerir configurações de sistemas aviónicos espaciais de maneira intuitiva e eficiente.

2. Funcionalidades Principais:
    a. Criação e Configuração de Arquiteturas: Esta aplicação permite a criação e alteração das arquiteturas, nos seus vários componentes (configuration, arinc653module, airconfiguaration, partition, partionconfiguration, memory, moduleschedule, partitionschedule, windowschedule e windowconfiguaration).

    b. Exporte Automático de XML: Esta aplicação permite o exporte da arquitetura em formato xml.

    c. Importe Automático de XML: Esta aplicação permite o importe da arquitetura em formato xml.

    d. Validação Automática: Incorpora ferramentas de validação que verificam a consistência e correção das configurações, reduzindo significativamente o risco de erros, na importação do XML e na configuração, visualização com possibilidade de alteração do partitionschedule da aquitectura.

3. Acessibilidade e Usabilidade
    a. Descrição: A interface centrada no utilizador, com um design intuitivo, permitiu que utilizadores com conhecimentos técnicos limitados pudessem configurar sistemas complexos. Evita-se a necessidade de escrita, procurando que os campos a preencher sejam na grande maioria de seleção ou formatados de forma especifica para evitar enganos.

4. Redução de Erros
    a. Descrição : As ferramentas de validação automática incorporadas na IGA (funções de validação de formatos, campos de seleção, validações adicionais de consistência da informação e preenchimento automático de campos, analisando os valores inseridos anteriormente), procuram

diminuir os erros de configuração, melhorando a confiabilidade e segurança das configurações desenvolvidas.

7.2 Cumprimento de requisitos

Os critérios de sucesso foram definidos no levantamento de requisitos e revistos ao longo do desenvolvimento do projeto. Abaixo estão os principais critérios e a análise e o seu cumprimento:

7.2.1 Criação de arquiteturas:

Critério: Deve ser possível a criação de arquiteturas no programa e sua personalização.

Cumprimento: Verificado. Os testes mostram que é consegue realizar a criação de uma arquitetura a alterar a mesma sempre que for necessário

7.2.2 Transferir informação da BD:

Critério: O programa deve facilmente transferir informação da BD.

Cumprimento: Verificado. Os testes mostram que se consegue realizar a transferência de informação da BD sem dificuldade, existindo para isso, uma interface e funções na BD para esse efeito.

7.2.3 O Export da arquitetura para um XML:

Critério: O programa deve conseguir exportar o XML depois de ter sido objeto de manutenção evolutiva.

Cumprimento: Verificado. Os testes mostram que se consegue realizar a exportação de qualquer arquitetura, em qualquer fase da sua construção, na sua última alteração.

7.2.4 A leitura de um XML:

Critério: O programa deve conseguir ler XML.

Cumprimento: Verificado. Os testes mostram que programa se consegue ler XML, tendo em conta que a função de Base de dados que pesquisa a arquitetura devolve sempre um XML que depois é lido e convertido na informação que é carregada na interface.

7.2.5 A possibilidade da realização de um time Schedule:

Critério: O programa deve permitir a realização de um time Schedule de forma correta.

Cumprimento: Verificado. Os testes mostram que é possível criar o Schedule (partitionschedule) na operação de configuração. Pode criar as Windowschedules na operação de configuração ou na operação de visualização, que vão permitir definir o time Schedule para todas as partitions consideradas, nos cores definidos.

7.2.6 Alteração do time Schedule

Critério: Deve ser possível a realização da alteração do Time Schedule depois deste ter sido já finalizado.

Cumprimento: Verificado. Os testes mostram que se consegue fazer a alteração do Time Schedule depois deste ter sido já finalizado nas operações de configuração e visualização.

7.2.7 A visualização da altura de criação

Critério: Deve mostrar no écran a mensagem de criação com sucesso ou razão do insucesso.

Cumprimento: Verificado. A criação de uma configuração, em modo de importação ou em modo de configuração devolve sempre a indicação ao utilizador do sucesso ou razão do insucesso.

7.2.8 A visualização da altura em que a arquitetura foi alterada

Critério: Deve mostrar no écran o resultado da alteração de uma determinada arquitetura.

Cumprimento: Verificado. Os testes mostram que as alterações de arquitetura, nos seus diversos módulos recebem sempre uma mensagem visível para o utilizador com o sucesso ou razão do insucesso.

7.2.9 Apresentar uma arquitetura user friendly:

Critério: Permitir configurações dos parâmetros, controlo de perfis, facilidade de criação\ alteração\importação\exportação com menus, importação. Acessível de qualquer ponto e em qualquer sistema operativo.

Cumprimento: Verificado. Os testes mostram que estas funcionalidades estão presentes, pode aceder em qualquer browser e em PC, tablet ou mobilePhone (com algumas restrições se os equipamentos tiverem écrans com valor inferior a 600 de width, no modo de visualização do time schedule). Os campos são de seleção, checkbox, com formatos bem definidos, ajudando o utilizador. Menu de fácil interação. Utilização de Tabs e janelas modal na visualização da configuração, evitando que a página ultrapasse o tamanho do ecran, com necessidade de efetuar o scroll vertical. Modo de visualização com opção de alteração do Time Schedule.

8 Conclusão

8.1 Conclusão

A implementação da Interface Gráfica Avançada (IGA) para a configuração de sistemas aviónicos espaciais representa uma inovação significativa no campo da engenharia de sistemas distribuídos. Este projeto abordou e solucionou diversos desafios inerentes ao processo de configuração por XML, utilizando uma base de dados relacional, o que representou um desafio relevante, não se tratando apenas de uma tabela que guarda configurações num campo, mas que ofereceu uma interface de procura, criação e alteração em árvore com eficiência, mas também garantindo a qualidade dos dados guardados, pelas validações que implementou.

8.2 Trabalhos Futuros

Extender o modo de visualização em formato de edição (já existe a consulta), para a configuration, arinc653module, moduleschedule, partitionschedule. Já faz nos outros módulos. Adaptar a visualização para formatos móveis inferiores a width < 600, para permitir uma melhor utilização. Adaptar a configuração, para formatos móveis inferiores a width < 600, para os botões disponibilizados nos tabs (tudo o resto se adapta sem problemas).

Permitir que a BD não só guardo em formato de BD relacional em árvore, como igualmente possa funcionar, guardando a configuração em formato XML e criando histórico de alterações.

Procurar que os campos numéricos, nos casos de definição de tempos sejam barras deslizantes com valor máximo, mínimo e intervalos de valores parametrizados, evitando assim a validação posterior do formato e compatibilidade de valores de intervalo.

Bibliografia

[DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.

[DEISI24b] DEISI, www.deisi.ulusofona.pt, Out. 2024.

[TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.

[ULHT21] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt, acedido em Out. 2024.

[Pikeos2] Pikeos, https://www.sysgo.com/fileadmin/user_upload/data/flyers_brochures/SYSGO_PikeOS_Product_Note.pdf , 18.11.2024

[XtratuM] XtratuM, https://www.fentiss.com/wp-content/uploads/2022/11/XM_for_Safety_Critical_Embedded_Systems.pdf , 18.11.2024

[01] https://www.earthdata.nasa.gov/s3fs-public/2023-11/newspace_nasa.pdf 18.11.2024

[02] https://github.com/air-gmv/air 18.11.2024

[03] https://www.easa.europa.eu/en/downloads/48264/en 18.11.2024

[04] https://www.aviationtoday.com/2007/02/01/integrated-modular-avionics-lessis-more/ 18.11.2024

[PgAdmin] https://www.pgadmin.org/ 21.11.2024

[Xampp Apache] https://www.apachefriends.org/ 21.11.2024

[Visual Studio Code] https://code.visualstudio.com/ 21.11.2024

Glossário

LEI Licenciatura em Engenharia Informática

LIG Licenciatura em Informática de Gestão

TFC Trabalho Final de Curso

API Interface de Programação de Aplicações

GMV Grupo Mecanica e Voo Advanced and Intelligent Robotics

XML Extensible Markup Language

IMA Integrated Modular

IGA interface gráfica avançada.