

DETIwall

Relatório Técnico

**Departamento de Eletrónica,
Telecomunicações e Informática
Universidade de Aveiro
Portugal**

Projeto em Engenharia Informática



Afonso Teixeira, Bernardo Barreto, Diogo Vicente, João
Oliveira, Miguel Cabral, Rodrigo Santos
Professor José Vieira, Professor Paulo Dias, Professora
Beatriz Sousa

9 de julho de 2020

DETIwall

Relatório Técnico de Projecto em Engenharia Informática do Mestrado Integrado em Engenharia de Computadores e Telemática da Universidade de Aveiro

Autoria de Afonso Teixeira (93170) apteixeira@ua.pt, Bernardo Barreto (93271) bernardo.barreto@ua.pt, Diogo Vicente (93262) dvicente@ua.pt, João Oliveira (93282) joaoroliveira@ua.pt, Miguel Cabral (93091) miguel.f.cabral@ua.pt, Rodrigo Santos (93173) rodrigo.santos18@ua.pt

Orientado pelo Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, José Vieira, pelo Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, Paulo dias e pela Professora Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, Beatriz Sousa.

Resumo

projeto DETIWall, do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, está integrado num projeto da Rede nacional de Computação Avançada e visa a instalação e configuração de um centro de visualização com uma video wall 3x3. Este centro irá ser utilizado para visualização de dados científicos e providenciar serviços de visualização à comunidade da região, permitindo que investigadores e empresas da região tenham acesso e possam usar Hardware Estado da Arte de Visualização.

Os objetivos principais foram a instalação e configuração do centro de Visualização e a identificação de potenciais necessidades de visualização, dentro do Departamento e da Universidade de Aveiro, e posterior desenvolvimento de protótipos de aplicações de visualização. Relativamente à instalação e configuração do centro era necessário assegurar que utilizadores terceiros possam usufruir da Video Wall para testar as suas aplicações de visualização sem necessitarem de configurações complexas ao nível de Hardware e Software. Foram identificados dois casos de uso dentro da Universidade, um com dados relativos a dados de tráfego relativos a *Access Points* da Universidade de Aveiro e outro com dados radioastronómicos.

Neste sentido, a equipa configurou um servidor de teste com o software para a Video Wall e desenvolveu aplicações para de forma a mostrar o potencial de um centro de Visualização destas dimensões à comunidade.

Neste momento a Universidade de Aveiro, encontra-se na fase final de aquisição do hardware.

Agradecimentos

Devido ao grande apoio que recebemos durante todo o processo de realização do projeto, o grupo tem todo o gosto de agradecer aos professores orientadores, aos professores da Unidade Curricular de Projeto em Engenharia Informática, ao Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro , ao Instituto de Engenharia Eletrónica e Telemática e aos Serviços de Tecnologias de Informação e Comunicação.

Contents

1	Introdução	8
1.1	Contexto	9
2	State of the Art	10
3	Análise Conceptual	11
3.1	Requisitos de Sistema	11
3.1.1	Levantamento de Requisitos	11
3.1.2	Atores	11
3.1.3	Diagramas de Casos de Uso	12
3.1.4	Requisitos Funcionais	16
3.1.4.1	Servidor	16
3.1.4.2	Visualização 2D	16
3.1.4.3	Visualização 3D	16
3.1.5	Requisitos Não Funcionais	17
3.1.5.1	Usabilidade	17
3.1.5.2	Performance	17
3.1.5.3	Segurança	17
3.1.5.4	Documentação	18
3.2	Arquitetura do Sistema	19
3.2.1	Modelo De Tecnologias	19
3.2.1.1	Servidor da Video Wall	19
3.2.1.2	Visualização 2D	19
3.2.1.3	Visualização 3D	20
4	Implementação	21
4.1	Repositórios de Código	21
4.2	Servidor da Video Wall	21
4.3	Visualização 2D	22
4.4	Visualização 3D	32
5	Resultados e Discussão	37
5.1	Servidor da Video Wall	37
5.2	Visualizações	37
5.3	Testagem	38

6	Conclusão	39
7	Referências	40

Índice de Tabelas

3.1	Atores do Sistema	12
3.2	Descrição de Casos de Uso do servidor da Video Wall	13
3.3	Descrição de Casos de Uso da aplicação 2D	14
3.4	Descrição de Casos de Uso da aplicação 3D	15
3.5	Requisitos Funcionais do Servidor.	16
3.6	Requisitos Funcionais da Visualização 2D.	16
3.7	Requisitos funcionais da Visualização 3D.	16
3.8	Requisitos de Usabilidade.	17
3.9	Requisitos de Performance.	17
3.10	Requisitos de Segurança.	17
3.11	Requisitos de Documentação.	18
4.1	Tabela de repositórios	21

Índice de Figuras

3.1	Diagrama de Casos de Uso do servidor da Video Wall	12
3.2	Diagrama de Casos de Uso da aplicação 2D	14
3.3	Diagrama de Casos de Uso da aplicação 3D	15
3.4	Caption5	20
4.1	Fase Inicial visualização 2D usando o D3	24
4.2	Migração do D3 para o ESRI	25
4.3	Tuplo da Tabela de dados do AP	26
4.4	Tuplo da tabela Localização	26
4.5	Tabelas da Base de Dados	27
4.6	Script em Python consumo de dados e envio para a BD	28
4.7	Batch script para cron	28
4.8	Acesso da firebase à aplicação	29
4.9	Função de criação de layers	30
4.10	Função de criação de pontos	31
4.11	Estrutura do formato FITS	33
4.12	Exemplo de um <i>Primary Header</i>	34
4.13	Exemplo de um <i>Extension Header</i>	35
4.14	Código referente à mapeação da esfera	36

Introdução

Este relatório técnico desenvolve-se em torno de representar todo o processo de implementação do projeto DETIwall, proposto na UC de PEI. O grande fator para o qual o grupo ter escolhido este trabalho deveu-se ao facto de ser algo novo e inovador, tanto para o DETI, como para o grupo. A oportunidade surgiu então o grupo decidiu encarar este projeto que irá trazer muitas mais valias para o departamento.

Este projeto, como já foi referido anteriormente está integrado num projeto da Rede nacional de computação avançada como tal um projeto completamente novo para a comunidade do departamento.

A grande inovação deste projeto seria que na conclusão deste tivessemos no DETI uma videowall 3x3 que iria ser bastante útil na ajuda a pesquisas por parte de certas investigações, que iria ser bastante importante em reuniões colaborativas e proporcionar um ambiente de trabalho muito mais comunicativo, aproximando mais as opiniões de todas as pessoas, visto que existia algo visual em que elas se poderiam apoiar. Com a implementação desta videowall o grupo também ficou responsável para fazer a criação de visualizações teste para conseguir testar todas estas funcionalidades da videowall. Sendo estas uma visualização 2D e outra 3D.

Como resultado final foi criado um servidor que irá estar colocado dentro da Universidade de Aveiro em que desta maneira será possível a comunicação com a videowall. A videowall será composta por nove televisores de 55", em que um estará ligado ao servidor sendo assim feita a comunicação entre o servidor e a videowall.

Quanto às visualizações criadas foi realizada uma com a parceria dos STIC, em que é possível fazer a visualização das pessoas penduradas em cada AP da UA. E a outra é feita através da leitura e renderização de dados radioastronómicos

Este relatório vai então ser o desenrolar de tudo aquilo que o grupo realizou, tal como a análise dos resultados obtidos pelos membros do mesmo. Também será discutido todos os problemas que foram encontrados na realização do projeto e algumas decisões que também foram alterando o rumo do mesmo.

1.1 Contexto

A visualização a uma grande escala tem sido cada vez mais uteis para estudos, monitorizações, e trabalho cooperativo.

Como tal a implementação da DETIwall, uma videowall 3x3, vai ajudar no trabalho cooperativo dentro da comunidade estudantil, e também o estímulo do uso desta parede para trabalhos de grande escala.

State of the Art

A inovação da tecnologia vem acompanhada com uma necessidade cada vez maior da visualização e estudo de dados, estes estando cada vez maiores e sendo necessário ser possível visualiza-los de uma forma coordenada, com grande detalhe. Sendo o DETI um departamento cada vez mais na busca desta inovação faz todo o sentido termos um centro de visualização para que seja possível realizar estes estudos de uma forma muito mais natural e sendo possível uma abstração muito melhor dos dados e do trabalho cooperativo.

Como tal foi feito o processo todo para a instalação de uma videowall. Para isto era necessário ter um servidor local que estabelecesse o contacto direto com a parede. Para contacto com o servidor os utilizadores usavam a pagina detiwall.servebeer.com (que eventualmente irá ser mudado para domain da UA).

Com a conclusão deste trabalho e apesar dos grandes esforços efetuados por todos envolvidos neste projeto, não possuímos ainda o material necessário para a criação desta parede. No entanto, todo o trabalho foi realizado para que quando seja feita toda a aquisição deste, a implementação seja o mais rápida possível.

Análise Conceptual

3.1 Requisitos de Sistema

Esta secção apresenta a especificação dos requisitos do sistema, sendo apresentado nas próximas sub-secções a descrição dos processos realizados para o levantamento de requisitos, seguida da lista de atores e diagramas de casos de utilização, assim como requisitos funcionais e não funcionais.

3.1.1 Levantamento de Requisitos

Este projeto subdividia-se em 2 categorias, a Video Wall e a criação de duas visualizações teste para mostrar o potencial de uma visualização a grande escala.

Quanto à Video Wall a pesquisa da equipa focou-se em perceber os vários tipos de software para Video Wall e nesse sentido, abordou, maioritariamente, os professores orientadores, o professor José Vieira e o professor Paulo Dias.

No que diz respeito às visualizações foi feito com os orientadores um estudo para quais seriam os melhores projetos para terem visualizações a grande escala. Após reuniões com várias pessoas com projetos interessantes, decidiu-se criar uma visualização de AP's no mapa da UA e uma com dados radioastronomicos, para o levantamento destes requisitos contamos com a ajuda então dos STIC para a primeira visualização e de membros do Square Kilometer Array para os dados radioastronomicos.

De acordo com estudos realizados sobre web-apps em outras unidades curriculares fizemos uma análise para ter requisitos não-funcionais que fossem de acordo com normas de QoS.

3.1.2 Atores

Na tabela 3.1 são apresentados os atores do sistema juntamente com o seu papel, respetivo. O Utilizador apresenta um papel de destaque, pois será o utilizador comum que irá utilizar o projeto para poder visualizar as suas aplicações e dados, uma vez que o DETIWall tem por um lado o objetivo de fornecer à comunidade uma Video Wall fácil de utilizar, e por outro aos utilizadores mais experientes uma ferramenta onde possam testar as suas aplicações em ecrãs de grande resolução.

Ator	Papel no Sistema
Utilizador	O Utilizador tem capacidade para se connectar à Video Wall e partilhar conteúdo através da partilha de ecrã ou de uma aplicação em específico.
Adminsitrador	O Administrador tem capacidade para modificar parâmetros do SAGE, fazer upload de novas aplicações na nativas e configurar novas Video Walls.

Table 3.1: Atores do Sistema

3.1.3 Diagramas de Casos de Uso

Os diagramas de casos de uso apresentados encontram-se devididos por pacotes: no diagrama 3.1 podem-se visualizar os diferentes casos de uso relativos ao servidor da Video Wall; no diagrama 4.2 podem-se visualizar os diferentes casos de uso relativos à aplicação 2D, ou seja, relativa a dados dos APs; no diagrama 3.3 podem-se visualizar os diferentes casos de uso relativos à aplicação 3D, ou seja, relativa a dados radioastronómicos.

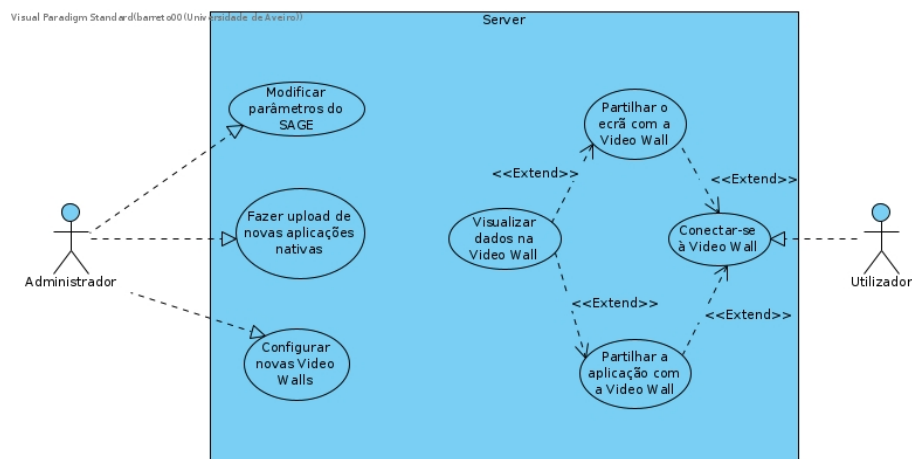


Figure 3.1: Diagrama de Casos de Uso do servidor da Video Wall

ID	Caso de Utilização	Sinopse
1.1	Modificar parâmetros do SAGE	Ao entrar nas configurações do servidor o Administrador será capaz de modificar parâmetros da Video Wall, como quantos monitores existem, a sua posição relativa e a sua resolução.
1.2	Fazer upload de novas aplicações nativas	O Administrador terá de fazer o upload do código desenvolvido terá de colocar o(s) ficheiro(s) numa determinada pasta e fazer a ligação entre a aplicação e o servidor.
1.3	Configurar novas Video Walls	Ao entrar nas configurações do servidor o Administrador será capaz de ligar novas Video Walls à existente, colocando o IP público no respetivo campo.
2.1	Conectar-se à Video Wall	Ao inserir no browser o IP público o Utilizador deverá colocar o seu nome e escolher uma cor para aceder ao controlos da Video Wall.
2.2	Partilhar o ecrã com a Video Wall	O Utilizador depois de conectado à Video Wall, utilizando os controlos desta, poderá realizar várias ações, neste caso o seu ecrã.
2.3	Partilhar a aplicação com a Video Wall	O Utilizador depois de conectado à Video Wall, utilizando os controlos desta, poderá realizar várias ações, neste caso utilizar aplicações nativas.
2.4	Visualizar dados da Video Wall	O Utilizador depois de partilhar o ecrã ou utilizar as aplicações nativas, deverá ver os dados de uma forma melhor do que veria num ecrã normal.

Table 3.2: Descrição de Casos de Uso do servidor da Video Wall

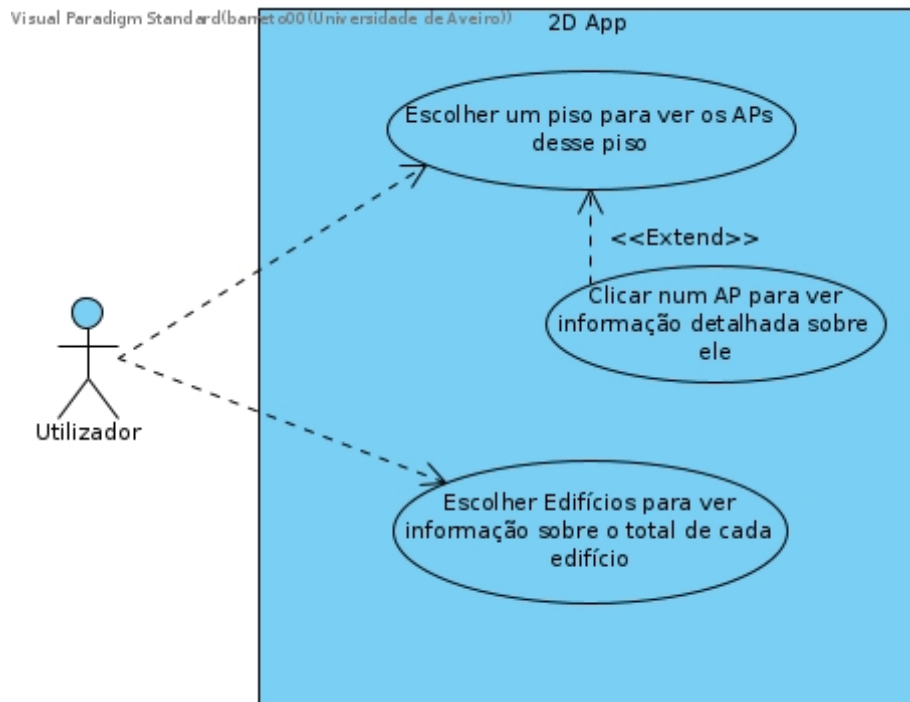


Figure 3.2: Diagrama de Casos de Uso da aplicação 2D

ID	Caso de Utilização	Sinopse
1.1	Escolher um piso para ver os APs desse piso	O Utilizador depois de entrar na Web App deverá conseguir escolher um piso para visualizar o seu mapa dos diferentes edifícios da Universidade de Aveiro e os APs localizados nesse mesmo piso.
1.2	Clicar num AP para ver informação detalhada sobre ele	O Utilizador, depois de escolher o piso que quer explorar e visualizar, poderá clicar num AP à sua escolha e ver informação detalhada sobre ele, como localização (longitude e latitude), nome do AP, número de pessoas que estão ligadas a esse AP.
1.3	Escolher Edifícios para ver informação sobre o total de cada edifício	O Utilizador depois de entrar na Web App deverá escolher a layer Edifícios para ver o mapa dos Edifícios da Universidade de Aveiro e poder ver informação do sumatório de todos os APs desse Edifício.

Table 3.3: Descrição de Casos de Uso da aplicação 2D

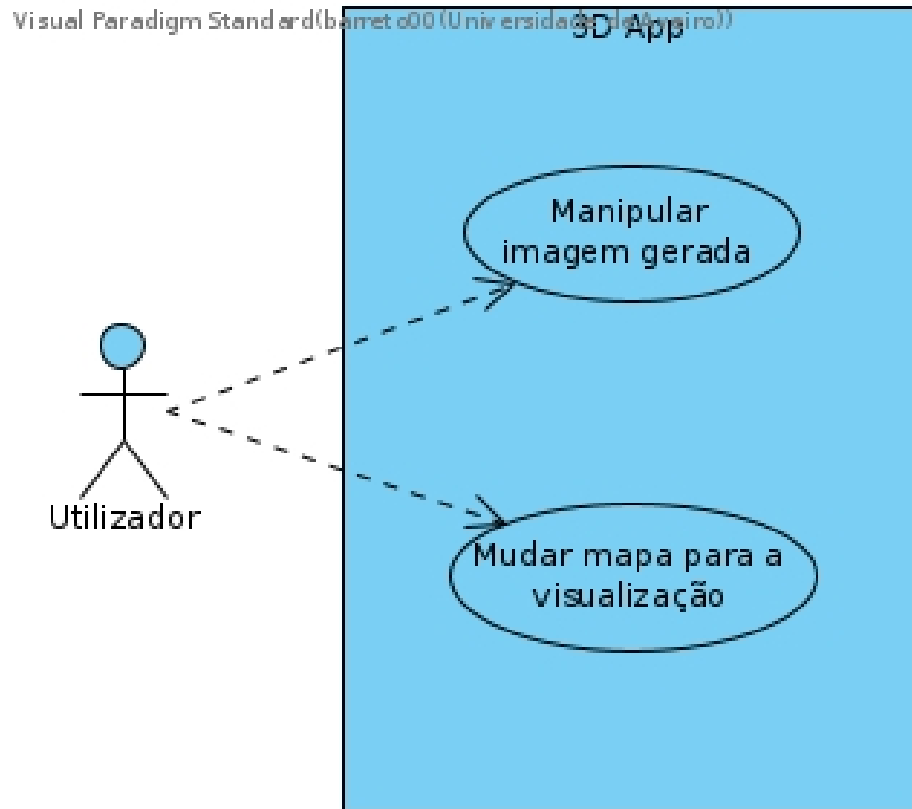


Figure 3.3: Diagrama de Casos de Uso da aplicação 3D

ID	Caso de Utilização	Sinopse
1.1	Mudar mapa para a visualização	O Utilizador depois de entrar na aplicação deverá conseguir selecionar o dataset que pretende visualizar para o mapa ser gerado.
1.2	Manipular a imagem gerada	O Utilizador, depois de entrar na aplicação e ter escolhido o dataset a visualizar, deverá conseguir manipular a imagem gerada a partir do dataset selecionado, isto é, fazer zoom in e zoom out, rodar o mapa, etc.

Table 3.4: Descrição de Casos de Uso da aplicação 3D

3.1.4 Requisitos Funcionais

3.1.4.1 Servidor

Referência	Requisito Funcional
FRS-1	Deve ser possível conectar-se ao servidor através de um ip público.
FRS-2	Deve ser possível controlar o que é visível através de vários computadores.
FRS-3	Deve ser possível alterar os parametros do servidor.
FRS-4	Deve ser possível correr aplicações nativamente.

Table 3.5: Requistos Funcionais do Servidor.

3.1.4.2 Visualização 2D

Referência	Requisito Funcional
FRsTIC-1	Deve ser possível visualizar informação em relação a densidade populacional.
FRsTIC-2	Deve ser possível interagir com a visualização.
FRsTIC-3	Deve ser possível criar gráficos e dashboards de fácil compreensão.
FRsTIC-4	Deve ser possível criar gráficos que correspondam com a verdade.
FRsTIC-5	Deve ser possível de criar gráficos/visualizações que sejam interativas com o utilizador.

Table 3.6: Requistos Funcionais da Visualização 2D.

3.1.4.3 Visualização 3D

Referência	Requisito Funcional
RF3D-1	Deve ser possível visualizar datasets de informação 3D.
RF3D-2	Deve ser possível interagir com a visualização.
RF3D-3	Deve ser possível visualizar dados de diferentes comprimentos de onda.
RF3D-4	Deve ser possível analisar os diferentes comprimentos de onda.

Table 3.7: Requisitos funcionais da Visualização 3D.

3.1.5 Requisitos Não Funcionais

3.1.5.1 Usabilidade

Referência	Requisito Não Funcional
NFRU-1	Deve ser de fácil uso.
NFRU-2	Deve ser possível mostrar informação através de um ou mais computadores.
NFRU-3	O utilizador tem de estar ligado à rede da UA.
NFRU-4	O resultado final tem de ter espaço de progressão.

Table 3.8: Requisitos de Usabilidade.

3.1.5.2 Performance

Referência	Requisito Não Funcional
NFRP-1	Deve ser capaz de exibir qualquer visualização a alta qualidade.
NFRP-2	Deve ser relativamente rápida.
NFRP-3	Deve ser capaz de suportar todos os sistemas operativos

Table 3.9: Requisitos de Performance.

3.1.5.3 Segurança

Referência	Requisito Não Funcional
NFRS-1	O servidor tem de ter protocolos que respeitam as medidas de segurança da UA.
NFRS-2	Apenas os administradores têm controlo do que é visualizado na Video Wall.
NFRS-3	Os utilizadores apenas terão acesso de partilha depois que estes estejam autenticados.
NFRS-4	Apenas os administradores podem remover quem está a usar a Video Wall.

Table 3.10: Requisitos de Segurança.

3.1.5.4 Documentação

Referência	Requisito Não Funcional
NFRDoc-1	O setup do servidor deve estar bem documentado para utilização futura.
NFRS-2	A documentação deve cobrir todos os serviços que a Video Wall fornece.
NFRS-3	A documentação sobre como se conectar ao servidor deve ser clara.
NFRS-4	A documentação deve ser simples e de fácil leitura.

Table 3.11: Requisitos de Documentação.

3.2 Arquitetura do Sistema

Nesta secção vai ser explorada a arquitetura do sistema, dos diferentes módulos do projeto: Servidor da Video Wall; Visualização 2D e Visualização 3D. E para cada um será apresentado o seu Modelo de Tecnologia.

3.2.1 Modelo De Tecnologias

3.2.1.1 Servidor da Video Wall

No que toca ao Modelo de Tecnologia do Servidor da Video Wall:

- A rede está configurada de modo a que seja possível aceder do exterior ao IP do servidor para se utilizar a Video Wall:
 - Estão abertas duas portas, uma para a ligação não segura e outra para a ligação segura.
 - A máquina onde está o servidor, tem um IP estático.
 - O DNS dinâmico está configurado de modo a que o hostname adquirido (videowall1.ua.pt) seja traduzido para o IP da máquina.
- A máquina que funciona como servidor, tem um software desenvolvido pela University of Hawaii at Mānoa's instalado, de nome SAGE2 específico para Video Walls, que se desdobra em servidor e software para Video Wall.
- Esta máquina está ligada a 9 monitores através de um único cabo e estes estão ligados entre si, utilizando o método daisy chain.

3.2.1.2 Visualização 2D

No que toca ao Modelo de Tecnologia da Aplicação 2D:

- Um script a correr numa máquina virtual em cloud vai buscar o access token para a API do Cisco Prime.
- Com o token vai retirar os dados da API do Cisco Prime, com uma taxa de amostragem de 13 em 13 min, em que temos os dados dos Access Points de toda a UA.
- Estabelece uma conexão https com o firebase (base de dados em cloud que é uma NoSQL DataBase) e guarda os dados.
- A web App (79afonso.github.io) vai aceder à API do ArcGis para retirar as plantas da Universidade, e também acede à API do FireBase de modo a retirar os dados dos Access Points conseguidos anteriormente. Todos estes dados são retirados usando o protocolo https.

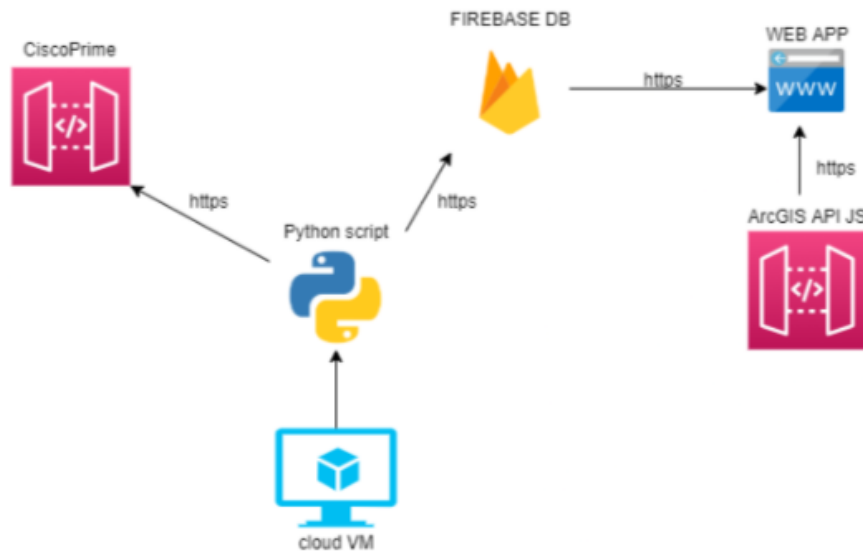


Figure 3.4: Caption5

3.2.1.3 Visualização 3D

No que toca ao Modelo de Tecnologia da Aplicação 3D:

- Os pontos são guardados em arrays com um grande poder de armazenamento devido à biblioteca NumPy;
- A biblioteca HealPy que através do algoritmo de pixelização mapeia os pontos lidos do ficheiro FITS numa esfera;
- A visualização da esfera é feita através da Visualization Toolkit (VTK).

Implementação

4.1 Repositórios de Código

Todo o código desenvolvido (das versões finais) no âmbito do projeto foi colocado em repositórios no github para facilitar o controlo de versões.

Referência	Requisito Funcional
Detiwall.github.io	Website e Documentação do projeto
79afonso.github.io	Web App 2D
detiwall.servebeer.com	Website para conexão com a videoWall (irá ser alterado para um domínio da UA)
3D Visualization	Repositório da visualização de dados Radioastronomicos

Table 4.1: Tabela de repositórios

4.2 Servidor da Video Wall

Como já referido na secção 3.2.1.1 a Video Wall funciona a partir de um software de nome SAGE2, específico para Video Walls. Este software foi-nos imposto no âmbito do projeto nacional de Rede Nacional de Computação Avançada pois permite a interligação de várias Video Walls tanto espalhadas pelo país como dentro da própria Universidade de Aveiro. No entanto, realizamos várias pesquisas para entendermos melhor o que são Video Walls e que tipos existem. Assim, foram distinguidos 2 grandes grupos:

1. Cada monitor da Video Wall tem o seu próprio computador e existem ainda um computador central que funciona como servidor.
2. Existe um único computador e os monitores encontram-se ligados diretamente à Video Wall.

Em ambas as modalidades o computador central funciona como servidor. No entanto no primeiro caso o servidor funciona para os cliente (utilizadores da Video Wall) se ligarem como para enviar as streams de bits para os respetivos

computadores de cada monitor através da rede. No segundo caso funciona apenas para os clientes se conectarem à Video Wall. Ainda no primeiro tipo de Video Wall, o computador central divide o que cada monitor irá mostrar e envia essa informação pela rede, e cada computador específico dos monitores faz essa renderização. Já no segundo tipo de Video Wall, o computador central faz todo o processo, isto é, divide e renderiza para todos os monitores. Neste sentido foram investigados vários softwares já existentes. Tais como, *WallConnect*, *HipperWall*, *Userfull*, *WallControl*, *Polywall* e finalmente *SAGE2*. O Software escolhido, *SAGE2*, não se encontra em nenhuma modalidade específica, pois permite ser utilizado com ambos os tipos. No nosso projeto, iremos optar por uma aproximação à segunda modalidade, isto é, iremos ter um computador capaz de suportar a renderização para 9 monitores que irão estar ligados entre si com daisy chain.

De modo a facilitar a posterior instalação em outros ambientes, foram feitos diversos testes locais e remotos. Iremos agora elencar os diversos testes e respectivas configurações.

- Configuração do servidor com um só monitor e apenas se poder aceder através do próprio servidor.
- Configuração do servidor de modo a aceitar clientes da mesma rede.
- Testes com clientes na mesma rede (teste local)
- Configuração do servidor para ter 2 monitores
- Configuração do servidor de modo a aceitar clientes de outras redes
 - Configuração do servidor - hostname e certificados.
 - Configuração do router - DNS dinâmico, IP estático e abertura de portas.
- Testes com clientes de outras redes. (teste remoto)
- Testes com aplicações desenvolvidas pelo grupo. (teste remoto)
- Configuração do servidor para ter 4 monitores.
- Testes com aplicações desenvolvidas pelo grupo. (teste remoto)

Tendo concluído estes testes, o servidor ficou configurado da seguinte forma: é possível aceder através de outras redes (apenas quando está ligado) e estão ligados 4 monitores.

4.3 Visualização 2D

Um dos objetivos deste projeto era a criação de uma visualização 2D capaz de produzir uma representação, bastante fiel, do mapa da universidade em que em cima desta conseguiríamos ver os access points da Universidade. Cada um dos

access points foi localizado fisicamente e a sua localização foi colocada numa base de dados, e através desta conseguir fazer um estudo demográfico sobre a quantidade de pessoas em diversas localizações da universidade.

Como tal, no início do projeto tivemos um elevado trabalho de estudo de ferramentas e bibliotecas para a criação de visualizações.

Foi feito um estudo para que os dados que obtíamos das API's fosse representado por dashboards. Para isso fizemos um estudo introdutório no grafana, uma ferramenta open source de visualização e processos analíticos. Permite fazer queries, alertar, visualizar métricas, basicamente permite ter uma representação visual de o que se está a passar dentro de uma base de dados. Isto seria uma implementação muito interessante para dados sobre tráfego de rede, mas depois de uma análise melhor e com o avançar do projeto decidiu-se optar pelos dados demográficos. Assim, descartou-se a possibilidade de uma dashboard.

Começando a trabalhar então na visualização a nossa próxima aproximação foi com o D3.

O D3 é uma biblioteca em javascript que permite manipular documentos baseados em dados. Esta ferramenta com a ajuda de HTML, SVG e CSS faz a transformação de dados para uma forma visual mais apelativa ao utilizador.

O grupo começou o seu estudo desta biblioteca com exemplos simples e a começar a fazer representações com mapas estáticos da universidade. Os testes sobre esta biblioteca tiveram resultados iniciais muito promissores.

Esta imagem seria uma representação inicial daquilo que tínhamos planeado para o D3, mas após pesquisa mais intensiva sobre a biblioteca, a conclusão tirada foi que não seria a melhor ferramenta a utilizar.

Depois desta aproximação, mudou-se de rumo e começou-se a trabalhar com o ArcGis, pois na API do ArcGis existem as plantas da universidade em formato svg. Tomou-se esta direção pois daria um resultado final muito mais aproximado com o objetivo que tínhamos inicialmente. Visto que a parede tem uma resolução elevada, a imagem do mapa da universidade tinha que ter uma resolução também elevada de modo a que quando se fizesse zoom não ficasse com baixa definição. Isto é conseguido através de imagens svg, deste modo, as imagens são desenhadas através de vetores, e por muito zoom que o utilizador faça ela vai manter sempre uma resolução elevada.

Depois de verificarmos que as imagens encontravam-se no formato de svg, vimos que estas também se encontravam georeferenciadas, tornando-se assim o formato ideal para a nossa implementação.

Utilizando a biblioteca do esri, a migração do D3 para o ArcGis foi bastante fácil, e conseguiu-se um resultado final mais apelativo e no geral melhor.

Dados dos APs

De maneira a obtermos os dados que necessitávamos, que neste caso seria a localização de cada AP e retirar deles os utilizadores pendurados em cada um, precisávamos de um sítio onde pudessemos juntar isso tudo.

campus
universitário

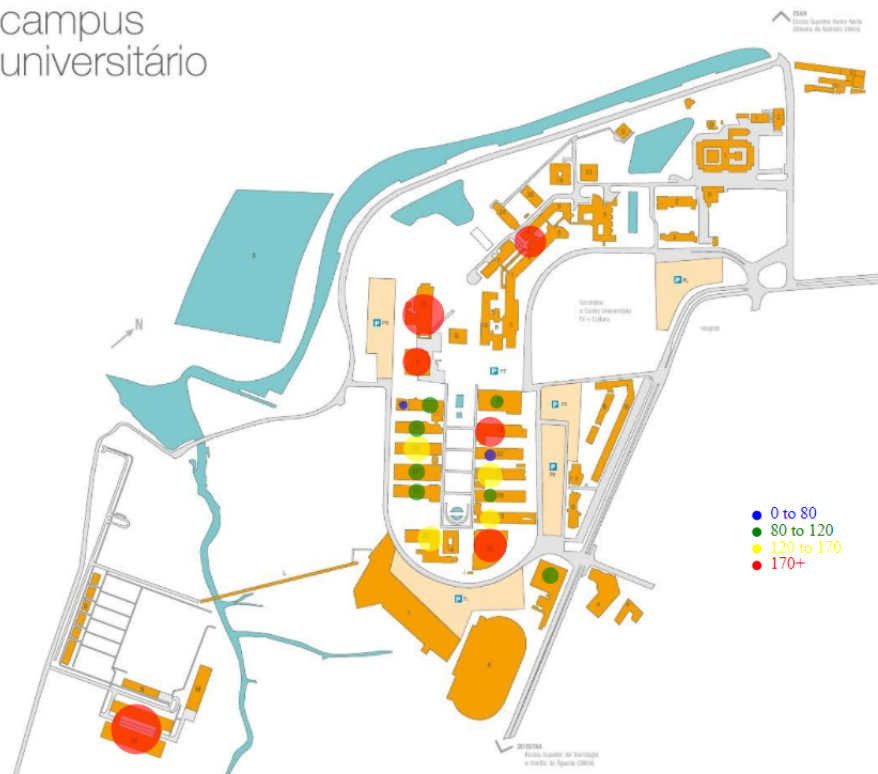


Figure 4.1: Fase Inicial visualização 2D usando o D3

No que diz conta à localização dos APs, com a ajuda dos Stic que nos disponibilizaram as plantas com a localização de cada AP da rede da universidade, isto engloba todos os departamentos, ESTGA, ESAN, residências, e usando o ArcGis, conseguimos marcar individualmente os cerca de 780 APs.

Para os dados dos utilizadores, foi utilizada a API do Cisco Prime que possuía dados como utilizadores ligados ao AP, MAC address do mesmo, upTime, entre outros como conseguimos visualizar na figura 4.5

Numa fase inicial, foi testado com que a web app consumisse os dados da API em tempo real, isso foi logo posto de parte, pois a API da Cisco Prime não o permite. Como tal, foi aí introduzida a ideia da criação da base de dados. Esta base de dados foi implementada no FireBase, visto que sendo uma DB implementada em cloud com a linguagem NoSql, permitiu-nos focar muito mais na visualização em si do que na criação de API para consumir dados da DB. Nesta base de dados foi feita uma divisão de duas tabelas em que uma apontava para a outra através do atributo ID.

A segunda tabela, como foi mencionado anteriormente, é a da localização dos dispositivos. Esta possuía o id de cada AP, tal como a sua localização (longitude



Figure 4.2: Migração do D3 para o ESRI

e latitude) todas marcadas individualmente pelo grupo, encontra-se também o nome do AP e o piso onde se encontra.



..... 1

..... clientCount: 0

..... clientCount_2_4GHz: 0

..... clientCount_5GHz: 0

..... id: 10055300

..... location: "Univ. Aveiro * ESTGA - Edif. principal"

..... macAddress: "003a9804d620"

..... model: "AIR-AP1131AG-E-K9 "


..... name: "estga-sap11.core.ua.pt"

..... status: "CLEARED"

..... type: "AP1130"

..... upTime: 194980912

Figure 4.3: Tuplo da Tabela de dados do AP



..... 0

..... id: "1857374445"

..... latitude: "40.6238405"

..... longitude: "-8.6575435"

..... name: "aauav-ap01"

..... Piso: "1"

Figure 4.4: Tuplo da tabela Localização

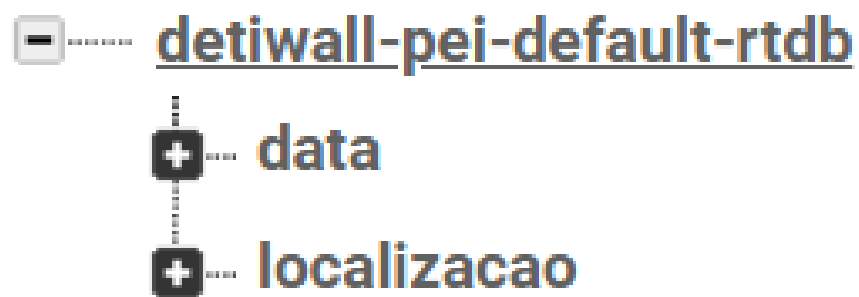


Figure 4.5: Tabelas da Base de Dados

Para consumo de dados da API do Cisco Prime foi criado um Script em python (figura 4.6), este script consumia dados de uma API para obter um access token, de forma a conseguir ler os dados fornecidos pela API do Cisco Prime. Após o consumo de dados da API do Cisco Prime, era feita a ligação para a nossa DB, a taxa de amostragem era de treze em treze minutos onde os dados eram guardados então no FireBase.

```
def access_points():  
    url = 'https://wso2-gw.ua.pt/primecore_primecore-ws/1.0.0/AccessPoint?maxResults=1000'  
    authorization = "Bearer " + token  
    x= requests.get(url, headers={"Authorization" : authorization})  
    response= x.json()  
    result= firebase.post("/data",response)  
    f = open("backup.log","w")  
    f.write("data sent!")  
    f.close
```

Figure 4.6: Script em Python consumo de dados e envio para a BD

Esta taxa de amostragem foi possível, fazendo com o script corresse com o Cron de acordo com o tempo especificado. O Cron é um serviço que executa comandos agendados em sistemas operacionais tipo Linux. Este serviço foi então utilizado numa máquina virtual em cloud, que estava colocada nos servidores do Azure.

```
*/13 * * * * python3 /home/detiwall/script.py > /home/detiwall/backup.log
```

Figure 4.7: Batch script para cron

O modo como a aplicação 2D vai buscar os dados é através de funções já existentes da própria firebase. Onde primeiro teríamos de ligar a firebase à aplicação e de seguida realizar uma série de tarefas que nos daria acesso à pasta da base de dados pretendida.

```
var firebaseConfig = {
  apiKey: "AIzaSyAGlltLGv62NuK7A0Af1TUE2SrE2x1pW4A",
  authDomain: "detiwall-pei.firebaseio.com",
  databaseURL: "https://detiwall-pei-default-rtdb.firebaseio.com",
  projectId: "detiwall-pei",
  storageBucket: "detiwall-pei.appspot.com",
  messagingSenderId: "251921800960",
  appId: "1:251921800960:web:ae6613c0a2c38220041de5",
  measurementId: "G-K70JYWB1R3"
};
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
var firebaseRef = firebase.database().ref("localizacao");
firebaseRef.once("value", function(snapshot){
  var data = snapshot.val();
});
```

Figure 4.8: Acesso da firebase à aplicação

ArcGIS/Esri

De modo a projetar um mapa georeferenciado da Universidade de Aveiro, fomos sugerido utilizar a API do ArcGIS. O ArcGIS é um serviço com bastantes funcionalidades, sendo uma delas partilhar mapas interativos, baseado na biblioteca Esri.

O principal objetivo da utilização de um mapa georeferenciado foi a facilitação na marcação de cada access point, visto que a localização deles era dada em coordenadas geográficas. Um outro ponto positivo que a utilização deste mapa nos trouxe foi uma visualização bastante mais apelativa.

Inicialmente tivemos algumas dificuldades a perceber como seria possível efetuar a projeção do mapa da universidade através da API do ArcGIS, mas depois de uma reunião com o técnico Luís Galiza dos serviços de gestão técnica ficou bastante mais claro os passos a seguir. Na reunião ficámos a perceber que o melhor método a seguir seria usar o mapa já anteriormente criado pela universidade e usar um MapImageLayer (um MapImageLayer é uma função da biblioteca Esri que nos permite projetar uma camada/mapa apenas necessitando do endereço do mapa que será usado como layer) e através desta projetar o mapa da universidade sobre o mapa mundo. Cada um dos pisos tem a sua própria layer sendo que todas estas são carregadas no início do programa para acelerar o processo de visualização.

Tanto para a projeção do mapa como para a projeção de cada ponto e dos respetivos pop-ups foi usada uma biblioteca de JavaScript chamada Esri, que é especializada na produção de soluções para a área de informações geográficas. O uso desta tornou-se uma mais valia pois permitiu que a marcação da localização

```
function createLayers(){
  layer = new MapImageLayer({ url: "https://sgt-websig.ua.pt/server/rest/services/Edifícios_UA/MapServer" });
  map = new Map({basemap: "dark-gray-vector", layers: [layer] });

  layer_p1 = new MapImageLayer({ url: "https://sgt-websig.ua.pt/server/rest/services/Piso_1/MapServer" });
  map_p1 = new Map({basemap: "dark-gray-vector", layers: [layer_p1] });

  var layer_p2 = new MapImageLayer({ url: "https://sgt-websig.ua.pt/server/rest/services/Piso_2/MapServer" });
  map_p2 = new Map({basemap: "dark-gray-vector", layers: [layer_p2] });

  var layer_p3 = new MapImageLayer({ url: "https://sgt-websig.ua.pt/server/rest/services/Piso_3/MapServer" });
  map_p3 = new Map({basemap: "dark-gray-vector", layers: [layer_p3] });

  var layer_p4 = new MapImageLayer({ url: "https://sgt-websig.ua.pt/server/rest/services/Piso_4/MapServer" });
  map_p4 = new Map({basemap: "dark-gray-vector", layers: [layer_p4] });

  var layer_p5 = new MapImageLayer({ url: "https://sgt-websig.ua.pt/server/rest/services/Piso_5/MapServer" });
  map_p5 = new Map({basemap: "dark-gray-vector", layers: [layer_p5] });
}
```

Figure 4.9: Função de criação de layers

de cada access, ou seja cada ponto fosse feita de modo imediato, o que nos acabou por salvar bastante tempo. Também com o uso desta, a própria visualização ficou, de um modo geral, muito mais apelativa.

```

// Add Points according to an array that contains coordinates
function addPoints(aps,mapa){

    graphicsLayer = new GraphicsLayer();
    mapa.add(graphicsLayer);
    for (i = 0; i < aps.length; i++) {
        var colorTest = changeColor(aps[i].people);

        const point = { //Create a point
            type: "point",
            longitude: aps[i].long,
            latitude: aps[i].lat
        };
        const simpleMarkerSymbol = {
            type: "simple-marker",
            color: colorTest,
            size: (aps[i].people/2)+4,
            outline: {
                color: "black",
                width: 0.4
            }
        };

        const attributes1 = {
            title: aps[i].apName,
            NUM_PES: aps[i].people,
            NUM_PISO: aps[i].Piso
        }

        const pointGraphic = new Graphic({
            geometry: point,
            symbol: simpleMarkerSymbol,

            attributes : attributes1,
            popupTemplate: popupOpenspaces
        });

        graphicsLayer.add(pointGraphic);
    }
}

```

Figure 4.10: Função de criação de pontos

4.4 Visualização 3D

Este módulo do sistema cumpre outro dos objetivos deste projeto que é mostrar outro tipo de visualização com impacto numa videowall. Esta visualização pretende mostrar dados de radioastronomia potencializando os benefícios para este tipo de visualização.

No início do projeto, o essencial foi a procura de informação sobre as ferramentas que iríamos usar bem como as bibliotecas para a criação das visualizações. A linguagem de programação foi moldada de acordo com aquilo que íamos analisando, chegando à conclusão de usarmos Python3.

Começámos então o estudo de um software Open Source, o VTK (Visualization Toolkit), para modelação, computação gráfica 3D, processamento de imagem, plots 2D e, no caso que mais interesse tinha para nós, na visualização de dados científicos. Após a construção de programas teste familiarizamo-nos com a arquitetura deste software apesar de conter um número vasto de classes e funções.

O próximo passo foi reconhecer o tipo de dados que iríamos tratar e este foi um dos processos que demorou mais tempo, visto que a física e a radio astronomia não está muito presente no domínio do nosso conhecimento. Posto isto, os ficheiros que contêm os datasets que posteriormente analisámos são do formato FITS. Este formato é particularmente mais difícil de analisar visto ser bastante flexível por ser facilmente editável por cientistas.

Já obtidos os dados, o próximo passo foi combiná-los para a visualização usando o VTK e outras bibliotecas. Mais uma vez a procura foi demorada com várias tentativas falhadas: usámos a biblioteca AstroPy que integra ferramentas que facilitam a interligação dos dados radioastronómicos com a linguagem Python permitindo a sua leitura usando funções I/O mas não foi possível a sua interligação com o VTK; usámos a biblioteca Matplotlib para a criação de uma esfera mas a sua visualização final era muito lenta. Portanto, a solução final foi a biblioteca HealPy que faz uso de um algoritmo de pixelização de uma esfera e mapeia os seus pontos sendo que cada um corresponde a um pixel além de permitir também a leitura dos ficheiros FITS em questão usando funções de I/O. Foi possível conjugar o VTK e esta biblioteca num mesmo ficheiro de forma a criar a visualização pretendida.

Além destas bibliotecas, ainda estudámos o NumPy e a sua função é dar suporte para arrays e matrizes multidimensionais pois os pontos mapeados rondam os milhões.

FITS

Flexible Image Transport System trata-se do formato standard para datasets relacionados com astronomia. Este standard foi desenvolvido inicialmente para imagens mas evoluiu para estruturas de dados mais avançadas. O FITS guarda a sua meta data em formato ASCII e os dados em bits. O formato também armazena dados em formato de tabelas ASCII ou binárias sendo que estas tabelas possuem named columns e linhas multidimensionais.

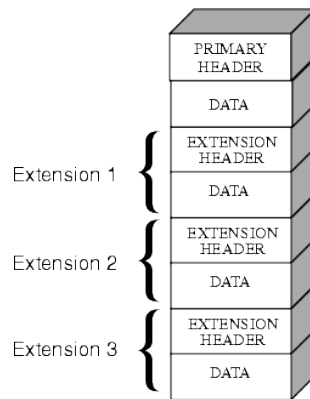


Figure 4.11: Estrutura do formato FITS

O formato está dividido em Header Data Units (HDU), cada um composto por um cabeçalho e um array de dados, o primeiro HDU é obrigatório estar definido, sendo os HDU seguintes opcionais. Este formato é bastante flexível o que leva a que haja bastantes formas diferentes de armazenar dados. Por exemplo, em muitos dos datasets que encontramos, o campo de dados do primeiro HDU servia como um dummy, estando os dados definidos no HDU seguinte. Os campos, geralmente, com mais importância neste formato tratam-se do BITPIX que indica o número de bits utilizado por cada valor do campo de dados, o NAXIS que indica o número de eixos associados ao array de dados. Na Figura 2 podemos também ver o primeiro campo “SIMPLE” que indica se o ficheiro está de acordo com o standard do FITS. (Até agora só trabalhamos com ficheiros que seguem o standard). Por último é também especificado o campo NSIDE que corresponde à resolução da imagem. Este campo é bastante importante para a pixelização da esfera, feita através do algoritmo HEALPix.

Em exemplos em que o ficheiro codifica uma imagem geralmente são usados $NAXIS = 2$ e é então tratado os dados como um array bidimensional (com a biblioteca Numpy). Em exemplos mais específicos para codificação de dados na esfera deparámo-nos mais uma vez na flexibilidade do formato figura 4.12 O ficheiro apresentado na Figura 2 e 3 trata-se de dados relativos à esfera, na figura abaixo é possível ver que se trata de uma tabela binária com 2 eixos e a especificação dos mesmos ($NAXIS1/NAXIS2$), está também especificado o tipo de dados desses eixos ($TTYPE1 = 'TEMPERATURE'$ e $TTYPE2 = 'N_OBS'$). Mais à frente nos campos é possível observar em comentário que o campo ‘N_OBS’ é apenas utilizado para consistência do formato, tratando-se, assim, estes dados, no fundo, de um array com 3145728 entradas ($NAXIS2$).

```
Terminal - diogo@diogo-pc: ~/Documents/PEI
>>> m[0].header
SIMPLE      =          T / file does conform to FITS standard
BITPIX      =          8 / number of bits per data pixel
NAXIS       =          0 / number of data axes
EXTEND      =          T / FITS dataset may contain extensions
COMMENT     FITS (Flexible Image Transport System) format is defined in 'Astronomy
COMMENT     and Astrophysics', volume 376, page 359; bibcode: 2001A&A...376..359H
DATE        = '2008-01-16T00:00:00' /file creation date (YYYY-MM-DDThh:mm:ss UT)
TELESCOP    = 'WMAP'       / Wilkinson Microwave Anisotropy Probe
OBJECT      = 'ALL-SKY'    / Portion of sky given
EQUINOX     =      2000.0 / Equinox of coordinates
COMMENT     This file contains the WMAP Internal Linear Combination (ILC) map
COMMENT     based on the first 5 years of observations.
REFERENCE   = 'WMAP Explanatory Supplement: http://lambda.gsfc.nasa.gov/'
RELEASE     = 'DR3'        / WMAP Release
VERSION     = 'PASS 3'     / Internal WMAP Version
STARTGMT    = '2001222000000000000' / Start time of TOD used
ENDGMT      = '2006221235900000000' / End time of TOD used
STOKES      = 'Intensity'  / i, q, or u
FREQ        = 'K-W bands combined' / Frequency band
STOKES      = 'Intensity'  / i, q, or u
RESOLUTN    =          9 / Resolution index
SKYCOORD    = 'Galactic'   / Coordinate system
PLANETS     = 'MJSUN'      / Planets avoided
COMMENT     Planet codes: [M]ars, [J]upiter, [S]aturn, [U]ranus, [N]eptune
SHIELDS     = 'S'          / Shield angle avoidance
COMMENT     Shield codes: [S]un, [E]arth, [M]oon
PIXTYPE     = 'HEALPIX'    / Pixel algorithm
ORDERING    = 'NESTED'     / Ordering scheme
NSIDE       =      512 / Resolution parameter
NPIX        =     3145728 / # of pixels
FIRSTPIX    =          0 / First pixel (0 based)
LASTPIX     =     3145727 / Last pixel (0 based)
>>>
```

Figure 4.12: Exemplo de um *Primary Header*

```
Terminal - diogo@diogo-pc: ~/Documents/PEI
>>> m[1].header
XTENSION= 'BINTABLE' /binary table extension
BITPIX = 8 /8-bit bytes
NAXIS = 2 /2-dimensional binary table
NAXIS1 = 8 /width of table in bytes
NAXIS2 = 3145728 /number of rows in table
PCOUNT = 0 /size of special data area
GCOUNT = 1 /one data group (required keyword)
TFIELDS = 2 /number of fields in each row
TTYPE1 = 'TEMPERATURE' /label for field 1
TFORM1 = '1E' /data format of field: 4-byte REAL
TUNIT1 = 'mK, thermodynamic' /physical unit of field
TTYPE2 = 'N_OBS' /label for field 2
TFORM2 = '1E' /data format of field: 4-byte REAL
TUNIT2 = 'counts' /physical unit of field
EXTNAME = 'Archive Map Table' /name of this binary table extension
DATE = '2008-01-16T00:00:00' /Table/MAP creation date
PIXTYPE = 'HEALPIX' /Pixel algorithm
ORDERING = 'NESTED' /Ordering scheme
NSIDE = 512 /Resolution parameter
FIRSTPIX = 3145727 /First pixel (0 based)
LASTPIX = 3145727 /Last pixel (0 based)
COMMENT = Temperature is in thermodynamic units
COMMENT = N_obs is set to 1 for all pixels. The N_obs field
COMMENT = is filled only to provide consistent map format MERANG and WMAP. Those
>>>
```

Figure 4.13: Exemplo de um *Extension Header*

HEALPix

Hierarchical Equal Area isoLatitude Pixelisation é um algoritmo de pixelização de uma esfera. Este algoritmo em si possui alguma matemática por trás, pelo que não será aqui abordado. É no entanto importante perceber que se trata de um algoritmo que divide a esfera em pixels de forma a que a área de superfície desses pixels seja sempre a mesma, o algoritmo enumera também os pixels de forma única. Nas imagens apresentadas acima é possível observar os campos associados a este algoritmo, estes são: PIXTYPE - Algoritmo HEALPIX ORDERING - Esquema de numeração de cada pixel. NSIDE - Que corresponde à resolução da imagem. FIRSTPIX e LASTPIX - Número associado ao primeiro e último pixel.

Em Python foi usada a biblioteca HealPy que implementa o algoritmo HEALPix. Esta biblioteca possui também funções de I/O com ficheiros FITS e algumas funções de visualização.

NumPy

NumPy trata-se de uma biblioteca matemática para python que oferece suporte para *arrays* e matrizes multidimensionais com grande volume. Esta biblioteca foi necessária para a manipulação dos dados que analisámos pois estamos a de dados na casa dos milhões. A utilização desta biblioteca acabou por se resumir a criar *arrays*, onde foram mapeados os pontos da esfera, e *grids*, com milhares de pontos, para as quais foi mapeado o valor escalar associado a esse ponto, que mais tarde é associado uma cor.

VTK

VTK foi a escolha inicial para visualização de dados 3D, pela sua relativa facilidade em relação a outros software de visualização mas sem comprometer nas possibilidades e na qualidade do produto final. Esta biblioteca utiliza um modelo de *pipeline* o que torna a sua utilização bastante sistematizada aquando de um devido domínio do *toolkit*. Esta biblioteca é responsável por toda a parte visível deste módulo, desde a realização de uma legenda, a esfera em si e a colorização da mesma. A esfera consiste em milhares de pontos mapeados conforme o algoritmo HEALPix.

```
for i in range(len(x)):
    for j in range(len(x[i])):
        array = np.array([x[i][j],y[i][j],z[i][j]])
        id = points.InsertNextPoint(x[i][j],y[i][j],z[i][j])
        cells.InsertNextCell(1)
        cells.InsertCellPoint(id)
        Colors.InsertNextTuple1(grid_map[i][j])
```

Figure 4.14: Código referente à mapeação da esfera

Resultados e Discussão

De uma maneira geral podemos afirmar que os objetivos deste projeto foram atingidos, tendo sido estudado e implementado, em material de teste, uma solução de *software* para o centro de visualização, bem como a realização de duas visualizações que exploram as capacidades deste centro de visualização.

5.1 Servidor da Video Wall

No que toca aos resultados obtidos do servidor, foram cumpridos todos os objetivos ficando apenas a faltar a configuração do computador final. O servidor está totalmente funcional e, assim, também a Video Wall.

5.2 Visualizações

Visualização 2D

No que toca aos resultados obtidos da visualização 2D, de um modo geral, estamos bastantes satisfeitos com estes pois todos os pontos mais cruciais foram realizados e só não foram atingidos os objetivos num único ponto. Sendos os que atingimos:

- mapa funcional da universidade georeferenciado
- pontos representando AP's na sua localização correta
- consumo de dados por parte da web app
- criação da base de dados
- alocação de serviços de cloud
- segurança de dados (através do serviço de BD usado)

E o que não atingimos:

- criar uma visualização com dashboard que fosse representativo dos dados fornecidos por cada access point, onde estes dados seriam analisados em mais detalhe.

Alguns dos pontos mais positivos que penso que podemos levar da visualização 2D realizada é o facto de toda esta visualização ser bastante interativa onde ao carregar em cada um dos pontos, é-nos fornecida informação sobre esse ap. Outro ponto também muito positivo é o facto de esta visualização poder servir de ponte para um maior controlo de fluxo demográfico. Com este controlo será possível ter prececção dos espaços da universidade com maior movimento populacional e também o horário a que esse fluxo ocorre com maior frequência. Mais tarde este controlo poderá ser usado para tentar distribuir este fluxo demográfico de uma maneira mais equilibrada, tentando reaproveitar os espaços com menos densidade populacional.

Visualização 3D

Os resultados obtidos na visualização 3D foram os definidos inicialmente, de salientar:

- A leitura de grandes volumes de dados;
- A pixelização de milhões de pontos numa esfera;
- A manipulação 3D desta esfera;
- A interação com a esfera.

Os pontos que gostaríamos de melhorar no futuro centram-se na interação entre a visualização e o utilizador, de salientar:

- Maior dinamização entre o ficheiro com os dados e a visualização;
- Melhor interação com os dados;
- Uma eventual conexão com um repositório de dados radioastronómicos.

5.3 Testagem

Visto que a realização deste projeto estava diretamente ligado à montagem e criação de um centro de visualização no Departamento de Eletrónica, Telecomunicações e Informática, sempre existiu a possibilidade de eventuais atrasos na aquisição do material para o centro de visualização devido principalmente a burocracias relacionadas com administrações públicas pois a compra de equipamentos nestas instituições necessita de aprovações e concursos públicos.

Já sabendo da possibilidade deste eventual precalço foi determinado que, em último caso, os testes seriam realizados num ambiente temporário de teste, que consistia em vários ecrãs/monitores de menor dimensão. De qualquer forma, todo o material desenvolvido e as suas implementações foram testadas no ambiente temporário.

Conclusão

Para concluir, o objetivo do nosso projeto era a configuração, suporte e desenvolvimento de ferramentas de visualização para uma videowall 3x3, e exemplos de aplicações que iam beneficiar de uma visualização com as grandes resoluções que esta permite.

Nesse sentido foi criado um protótipo semelhante ao que será implementado no Departamento. Ainda que numa escala muito menor, todas as funcionalidades do nosso prototipo vão poder ser usadas mesmo quando for implementada a videowall.

Para comprovar que o uso desta ferramenta era bastante útil ainda desenvolvemos duas visualizações, uma 2D e outra 3D que tentavam potencializar a utilidade da grande resolução do ecrã.

Como resultados, temos todo o processo inicial pronto, simplesmente encontra-se à espera do material, e assim que este chegue pensamos que a sua montagem seja o mais breve possível para que toda a gente possa começar a usufruir deste equipamento.

Como objetivos futuros, a implementação desta videowall e como esta está inserida na rede nacional de computação avançada, seria muito útil existir algum tipo de comunicação para que duas videowalls em sítios diferentes tivessem a comunicar e a demonstrar a mesma coisa.

Referências

- SAGE2 - Scalable Amplified Group Environment : <https://sage2.sagecommons.org/>
- SAGE2 - Instructions : <https://sage2.sagecommons.org/instructions/>
- SAGE2 - Videos : <https://sage2.sagecommons.org/videos/>
- Documentação ArcGis: : <https://developers.arcgis.com/javascript/latest/>
- Documentação Cisco Prime: <https://solutionpartner.cisco.com/media/primeinfrastructure>
- Documentação Firebase: <https://firebase.google.com/docs>
- Tese de um software de visualização : <https://ria.ua.pt/bitstream/10773/8617/1/tese.pdf>
- Documentação VTK : <https://vtk.org/doc/nightly/html/index.html>
- Documentação HealPy: <https://healpy.readthedocs.io/en/latest/>
- Documentação do Cron: <https://www.raspberrypi.org/documentation/linux/usage/cron.md>
- Site Fits : https://fits.gsfc.nasa.gov/fits_standard.html
- Mapa da Universidade - API ArcGIS : <https://sgt-websig.ua.pt/server/rest/services>