



UNIVERSIDADE
LUSÓFONA

App4SHM

Trabalho Final de curso

Relatório Final

Nome do Aluno: Nuno Penim

Nome do Aluno: Paulo Oliveira

Nome do Orientador: Pedro Alves

Trabalho Final de Curso | LEI | 20 de Junho de 2021

www.lusofona.pt

Direitos de cópia

App4SHM Copyright de Nuno Penim e Paulo Oliveira, ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

A análise e monitorização da integridade estrutural é um passo importante no acompanhamento do desempenho e de segurança das infraestruturas, na área de Engenharia Civil. Esta análise é realizada por um conjunto de ferramentas que fornecem ao Engenheiro responsável pela infraestrutura dados sobre a integridade da mesma, permitindo então perceber se a estrutura se encontra estável, se precisa de reparações, ou se está à beira do colapso.

Estas ferramentas geralmente contêm um extenso conjunto de acelerómetros muito sensível e preciso. No entanto também têm um custo muito elevado.

A App4SHM pretende ser uma aplicação para smartphones capaz de monitorizar estruturas de Engenharia Civil, recorrendo ao acelerómetro presente no mesmo, aplicando algoritmos inteligentes (localmente ou remotamente) aos dados obtidos pelo acelerómetro para inferir o nível de risco de uma determinada estrutura. Estes dados vão ser posteriormente guardados para futuras comparações da mesma, no caso de catástrofes, onde é possível a ausência de tais ferramentas, ou para o utilizador comum, que pretenda apenas investigar, de forma expedita, a integridade de uma estrutura.

Abstract

The analysis and monitoring of structural integrity is an important step in monitoring the performance and safety of infrastructures in the area of Civil Engineering. This analysis is performed by a set of tools that provide the Engineer responsible for the infrastructure with data on its integrity, allowing them to understand if the structure is stable, if it needs repairs, or if it is on the verge of collapse.

These tools usually contain an extensive set of very sensitive and accurate accelerometer. However, they are very expensive.

App4SHM aims to be an application for smartphones capable of monitoring Civil Engineering structures, using the accelerometer present in it, applying intelligent algorithms (locally or remotely) to the data obtained by the accelerometer to infer the risk level of a structure, this data will be stored for future comparisons, in the case of disasters, when there is absence of such tool, or for the common user, who just wants to investigate the integrity of a structure. This application is not intended to be a replacement for Civil Engineering tools for this purpose, since the accelerometers of smartphones are not as sensitive and accurate, having a high error

Índice

1	Identificação do Problema	7
1.1	Exemplos porque esta aplicação é necessária	7
2	Viabilidade e Pertinência	9
2.1	Viabilidade	9
2.1.1	Custos	9
2.2	Pertinência	9
3	Levantamento e análise dos Requisitos	10
3.1	Casos de Uso	11
4	Solução Desenvolvida	13
4.1	Utilização	13
4.1.1	Utilização individual	13
4.1.2	Utilização em grupos	13
4.2	Tecnologias	14
4.2.1	Cliente	14
4.2.2	Canal de comunicação	14
4.2.3	Servidor	14
4.2.4	Bibliotecas	15
4.3	Arquitetura da solução	16
4.4	Desenvolvimento Inicial	17
4.5	Discussão de problemas detectados	17
4.5.1	Captura de medições	17
4.5.2	Cadência de Captura dos dispositivos	18
4.6	Solução Final	19
4.6.1	Servidor	19
4.6.2	Cliente	19
5	Resultados	21
5.1	Interface	21
5.2	Leituras da App móvel	23
5.3	Resultados do Servidor	24
5.4	Ambiente de Testes	24
6	Benchmarking	25
7	Método e Planeamento	26
7.1	Milestones	26
7.2	Mapa de Gantt	27
	Bibliografia	28
	Glossário	29
	Anexo 1 - Detalhe do Desenvolvimento Inicial	30

Lista de Figuras

Figura 1: Queda de ponte em Taiwan	7
Figura 2: O antes e depois do colapso da Ponte Morandi.....	8
Figura 3: Viaduto Kinzua, após o seu colapso.....	8
Figura 4: Diagrama de funcionamento da App4SHM	10
Figura 5: Diagrama Use-Case da App4SHM.....	11
Figura 6: Diagrama de Arquitetura da App4SHM.....	16
Figura 7: Toast de Aviso ao Utilizador	17
Figura 8: Acelerograma de um teste realizado.....	20
Figura 9: Acelerograma após a medição, com todas as opções	21
Figura 10: Acelerograma após a medição, mas filtrado	22
Figura 11: Plataforma de Testes	23
Figura 12: Captura de ecrã da loja da Google, na página de uma das aplicações de análise de vibração .	25
Figura 13: Diagrama de funcionamento da App4SHM	26
Figura 14: Mapa Gantt do projeto App4SHM.....	27
Figura 15: Implementação inicial da App4SHM	30
Figura 16: Estrutura de código da App móvel	31
Figura 17: Captura de ecrã da nova versão da App4SHM	31
Figura 18: página web demonstrativa da App4SHM	31
Figura 19: Formato exemplo fornecido pelos Professores de Engenharia Civil	31

1 Identificação do Problema

A plataforma App4SHM que vamos desenvolver vai permitir uma fácil monitorização de estruturas sem a necessidade de ferramentas extra, reduzindo assim custos de operação.

Apesar dos acelerômetros e outros sensores dos dispositivos móveis não serem tão precisos como ferramentas profissionais da área, são mais baratos e quase toda a gente tem um telemóvel. O objetivo principal da App4SHM, não é substituir na integridade as ferramentas profissionais da área, mas sim oferecer uma forma mais rápida, especialmente durante catástrofes, quando estas ferramentas não se encontram disponíveis, a fim de monitorizar a integridade de uma estrutura.

A integridade de uma estrutura pode ser definida como a habilidade de uma estrutura executar o seu propósito (se for uma ponte, permitir alguma forma de tráfego, se for um edifício residencial, permitir a sua ocupação) de uma forma segura e sem falhas, mantendo-se numa peça.

Assim, a integridade estrutural é um requisito básico de qualquer estrutura, e é o dever de qualquer engenheiro envolvido no ciclo de vida de qualquer estrutura assegurar que a mesma existe.

1.1 Exemplos porque esta aplicação é necessária

No dia 1 de outubro de 2019, uma ponte, na Ilha Formosa (Taiwan), colapsou enquanto um camião de petróleo passava a ponte, estava bom tempo, mas o colapso ocorreu horas depois de um tufão ter atingido a ilha.



Figura 1: Queda de ponte em Taiwan

A ponte Morandi, em Itália, colapsou parcialmente dia 14 agosto 2018 durante uma tempestade.

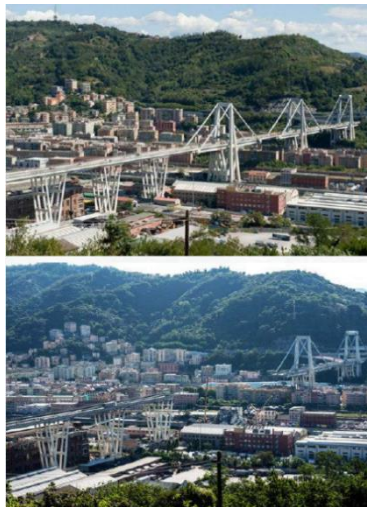


Figura 2: O antes e depois do colapso da Ponte Morandi

Em 2003, o viaduto Kinzua, na Pensilvânia, colapsou após um tornado. Este viaduto era considerado a “Oitava maravilha do mundo”. Na altura do seu colapso, o viaduto tinha 121 anos. A causa principal do seu colapso foi a fadiga dos materiais, nomeadamente de uns parafusos na base, que estavam corroídos.



Figura 3: Viaduto Kinzua, após o seu colapso

A maior parte de falhas de pontes acontece por uma combinação de vários fatores, como a fadiga dos materiais e falta de monitorização, e posteriormente, um evento natural extremo.

2 Viabilidade e Pertinência

2.1 Viabilidade

Numa fase inicial do desenvolvimento, a App4SHM vai começar por usar uma técnica matemática, baseada no método *Fast Fourier Transform*, chamado de *Frequency Domain Decomposition*, para traduzir sinais do acelerómetro para séries temporais que podemos utilizar em *Machine Learning*. Esta fórmula matemática já se encontra implementada, pelos professores de Engenharia Civil em MATLAB, podendo ser convertida para Python. Esta é uma das partes mais críticas, pois se não conseguirmos extrair os dados, é impossível analisá-los.

Foi-nos disponibilizado um *Toolbox*, em MATLAB – outra linguagem de programação mais focada em matemática, que contém todas as funções matemáticas que precisamos. Existem também diversas bibliotecas e implementações das mesmas – gratuitas e de código aberto – em Python, que mais à frente iremos abordar.

2.1.1 Custos

As ferramentas que vamos utilizar para fazer este projeto serão todas gratuitas:

- 1 (IDE) IntelliJ IDEA Community: <https://www.jetbrains.com/idea/>
- 2 (IDE) PyCharm: <https://www.jetbrains.com/pycharm/>
- 3 (IDE) Android Studio: <https://developer.android.com/studio>
- 4 (Framework) Flask: <https://flask.palletsprojects.com/en/1.1.x/>
- 5 (Framework) Spring boot: <https://spring.io/projects/spring-boot>

Os dados serão principalmente trabalhados no servidor que nas primeiras versões do trabalho poderá simplesmente ser um dos computadores dos programadores, mas chegará a um ponto que será necessário estar ligado 24/7 e usar um computador pessoal não é muito viável.

O maior custo, após o desenvolvimento, será o servidor para manter a aplicação operacional, e posteriormente a manutenção necessária a nível programático.

Esta aplicação vai ter como população alvo todos, mas numa primeira fase será específico para empresas como Infraestruturas de Portugal e Brisa, e para o setor da educação.

2.2 Pertinência

Este trabalho será realizado no âmbito de um estudo a ser realizado pelo departamento de Engenharia Civil da Faculdade de Engenharia da ULHT em específico pelos Prof. Eloi Figueiredo, Prof. Ionut Moldovan, Prof. Luís Silva e Prof. Hugo Rebelo com o objetivo de produzir uma solução de fácil acesso para auxiliar na monitorização do SHM.

Estes trabalhos estão enquadrados no âmbito do *Civil Research Group* (<http://civilresearchgroup.ulusofona.pt>).

Este projeto candidatou-se ao programa Fazer+, financiado pela ULHT.

3 Levantamento e análise dos Requisitos

Como mencionado anteriormente, este projeto foi pedido por terceiros, neste caso é um estudo realizado pelo departamento de Engenharia Civil, da Faculdade de Engenharia da ULHT.

De tal modo, os Professores de Engenharia Civil que sugeriram este projeto já tinham em mente um diagrama para o funcionamento da aplicação. Abaixo, na figura 4, está apresentado o diagrama sugerido pelos mesmos.

Diagrama da aplicação de smartphone para SHM

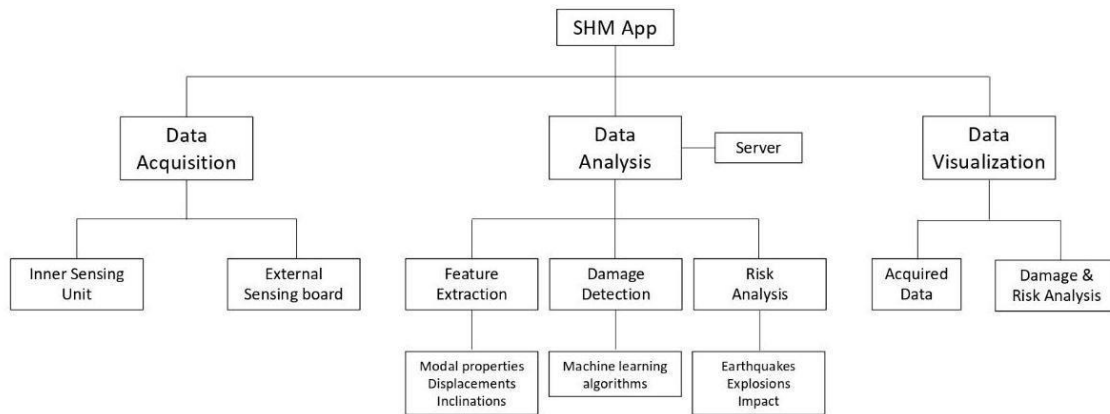


Figura 4: Diagrama de funcionamento da App4SHM

Aqui, é importante referir a existência de um servidor extra. Para tal foi importante projetar uma arquitetura para a aplicação, apresentada no capítulo seguinte.

Atualmente, encontra-se implementada a componente de *Data Acquisition* relevante à *“Inner Sensing Unit”*, que usa o acelerómetro do dispositivo móvel. No que toca a possíveis sensores externos, os Professores de Engenharia Civil relataram numa das nossas reuniões não ser uma prioridade para já.

No que toca à componente de *Data Analysis*, está implementada a componente de *Feature Extraction*, sendo que conseguimos ter todos os dados necessários sobre a integridade estrutural. A componente de *Damage Detection* ainda não se encontra implementada, por estar ainda uma discussão em aberto, com os Professores de Engenharia Civil.

Posteriormente, no que toca à componente de *Data Visualization*, já está implementada a componente de dados adquiridos.

No que toca à *Risk Analysis* em ambas as componentes de *Data Analysis* e *Data Visualization*, para já não é uma prioridade dos Professores de Engenharia Civil.

3.1 Casos de Uso

Ao analisarmos o diagrama da proposta realizado pelos Professores de Engenharia Civil, e posteriormente em conversas com os mesmos, realizamos um diagrama de casos de uso, que reflete as necessidades principais da solução. O mesmo pode ser observado abaixo, na figura 5. Este diagrama poderá ter alterações no futuro

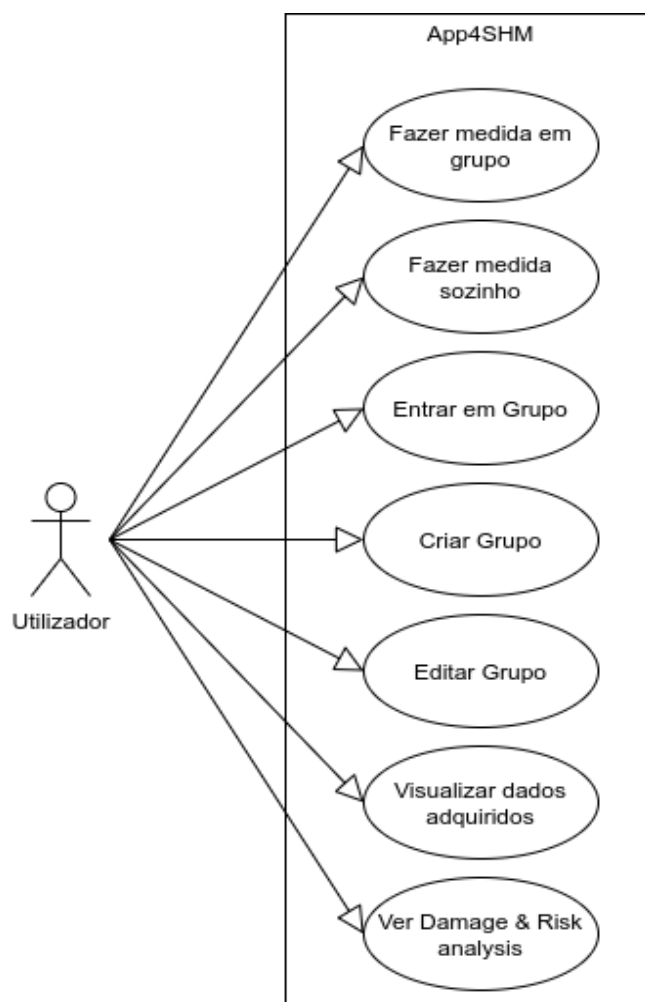


Figura 5: Diagrama Use-Case da App4SHM

Neste diagrama estão representadas as ações que os utilizadores poderão realizar. Aos utilizadores são dadas as opções de fazer uma medição sozinho, fazer uma medição em grupo, entrar em grupo, criar um grupo, editar grupo, visualizar os dados adquiridos e ver os dados relacionados com o *Damage & Risk Analysis*.

Para realizar uma medição sozinho, o utilizador encosta ou pousa o seu dispositivo móvel à estrutura, preferencialmente no centro da estrutura, para estar no ponto de maior oscilação, geralmente realizada pela ação da gravidade, e as oscilações naturais da estrutura serão medidas.

Criar um grupo, e posteriormente editar o grupo ou entrar num grupo, irá permitir aos utilizadores de mesmas equipas de medição realizar medições mais precisas, por existirem mais acelerómetros a registar os valores da mesma estrutura (eventualmente em pontos diferentes).

Para realizar uma medição em grupo, é necessário que todos os utilizadores estejam no mesmo grupo. O processo é semelhante ao de realizar uma medição sozinho, com exceção de que é aguardado até que todos os integrantes do grupo estejam prontos. Inicialmente, o “master” da sessão marca o início da sessão. De seguida, é aguardado que todos os integrantes da medição marquem o seu estado como “ready”. Esta ação é feita através do *click* num botão “ready”. Posteriormente, quando todos os dispositivos estejam marcados como *ready*, o master da sessão fica com um botão novo disponível, “iniciar”.

Todas as medições, sejam em grupo ou individuais, terão uma marca temporal fiável, para garantir a sincronização.

No final, ou se já existiam capturas, será possível visualizar os dados adquiridos, por via de acelerogramas, que representam o nível de oscilação da estrutura.

Será também possível visualizar os dados de *Damage and Risk Analysis*. Estes dados são um parecer do estado da estrutura, como por exemplo, se está segura, à beira de cair, ou vai cair a qualquer momento. Aqui entra a parte de *Machine Learning* e de Inteligência Artificial do projeto, visto que as decisões sobre o estado da estrutura irão ser cálculos realizados pelo servidor.

4 Solução Desenvolvida

Vamos desenvolver uma aplicação móvel Android nativa, onde os utilizadores podem monitorizar estruturas.

Esta aplicação não pretende substituir ferramentas profissionais da área, mas sim dar uma alternativa barata e rápida, especialmente em casos de catástrofe, quando equipamentos profissionais podem ser difíceis de transportar para o local ou foram extraviados pela catástrofe.

4.1 Utilização

4.1.1 Utilização individual

Cada utilizador pode monitorizar uma estrutura, através de medições estruturais e receberá o resultado dizendo se esta se encontra em bom estado para ser utilizada, se está um pouco diferente dos valores habituais medidos anteriormente, ou se está muito afastado dos valores habituais, sendo um risco para os seus utilizadores.

Para a nossa aplicação poder fazer esta comparação precisa de valores *baseline* de quando a estrutura se encontrava em bom estado. Estes valores irão estar armazenados na base de dados do servidor, que será abordada mais à frente, na Arquitetura da Solução. Estes valores *baseline* numa primeira fase, serão carregados externamente, sendo que alguns já poderão ser conhecidos. O carregamento e atualização destes valores é feito automaticamente, com cada medição realizada com sucesso.

Na possibilidade de indisponibilidade destes dados, a aplicação não irá ter um parecer sobre o estado da estrutura, limitando-se a apresentar os valores da medição realizada.

4.1.2 Utilização em grupos

O utilizador pode convidar pessoas para se juntar à sua monitorização.

As medições podem ser feitas por vários telemóveis ao mesmo tempo. No grupo pode-se iniciar uma medição e selecionar o número de telemóveis que pretende fazer a medição.

Após a escolha, ficará à espera que outros telemóveis, com a mesma conta ou não, que estejam no grupo de monitorização, podendo-se juntar manualmente, introduzindo um código ou adicionando um utilizador pelo seu identificador, ou por código QR.

As monitorizações em grupo irão ser sincronizadas recorrendo a *Web Services REST*. Utilizar Bluetooth seria pouco viável, visto que o alcance do Bluetooth é apenas de 10-15 metros. Para estruturas maiores, isto torna-se impossível.

Depois de o número de telemóveis ser o pretendido vai existir uma confirmação para verificar se todos os telemóveis estão prontos para começar as medições. Estas são realizadas com os equipamentos no chão encostados à estrutura. Depois das medições serem efetuadas, vão ser efetuados cálculos sobre estas (no telemóvel e no servidor). Os resultados serão guardados nos smartphones dos integrantes do grupo para a visualização dos utilizadores, mais tarde, e numa base de dados no servidor, para a comparação com as próximas medidas.

Abaixo, na figura 6, está apresentado um diagrama sugestivo do funcionamento da aplicação no modo de Grupos.

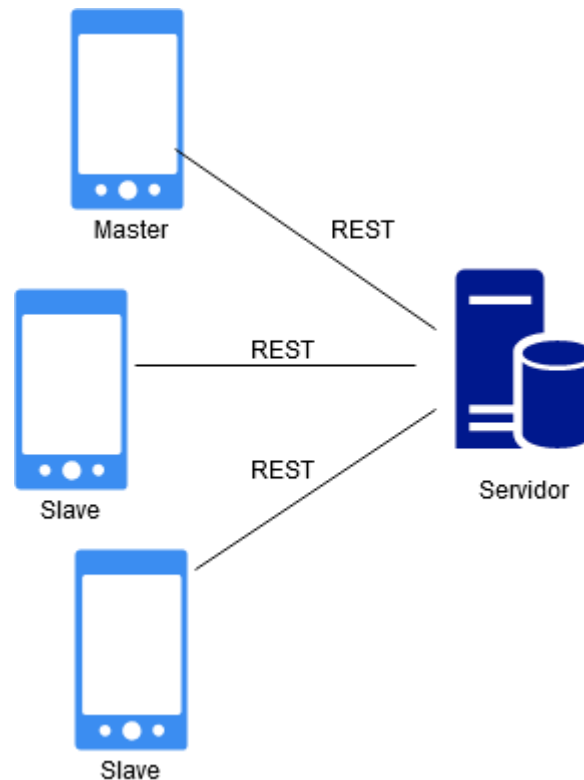


Figura 6: Diagrama sugestivo de funcionamento com dispositivos em modo de grupo

É importante reforçar neste diagrama o que foi explicado em 3.1, nas medições em grupo. Existirá sempre um Master, sendo que todos os outros dispositivos são Slaves na medição.

4.2 Tecnologias

4.2.1 Cliente

Será um dispositivo móvel, baseado em Kotlin. Inicialmente a App estará apenas disponível para dispositivos Android devido à limitação da linguagem Kotlin.

4.2.2 Canal de comunicação

São utilizados *Web Services*, em RESTful JSON, permitindo assim uma maior modularidade.

4.2.3 Servidor

No lado do servidor, inicialmente utilizamos Kotlin. O seu desempenho é semelhante ao Java, e muitos componentes Java funcionam no Kotlin. Assim, apenas necessitaríamos de usar uma tecnologia, tanto em lado servidor como em cliente, facilitando assim o diagnóstico de problemas e a resolução dos mesmos.

Na versão inicial em Kotlin, foi utilizada como *framework*, a *framework* SpringBoot. Esta *framework*, de acordo com o seu guia de *Getting Started*, permite um

desenvolvimento rápido de aplicações, e permite encurtar o código, tornando o mesmo mais simples e eficiente.

No entanto, após uma discussão com os Professores de Engenharia Civil envolvidos no projeto, e tendo em conta que muitos dos algoritmos que nós necessitamos já se encontravam feitos em MATLAB, o nosso servidor foi portado para Python.

O ambiente de desenvolvimento MATLAB permite exportar os scripts em MATLAB para C, Fortran e Python. Como destas linguagens, a mais fácil de trabalhar, e a que tem mais *frameworks* para aplicações web é a linguagem Python, esta foi a nossa escolha para a nova linguagem do servidor.

De momento, o servidor tem todas as operações matemáticas implementadas, sendo que o projeto já faz os cálculos das frequências de ressonância de uma estrutura. A obtenção destas frequências é o objetivo final, uma vez que com estas frequências, um Engenheiro já consegue dizer o estado dos materiais e a integridade estrutural do objeto de estudo. Assim, o que está em falta é apenas a componente de *Machine Learning* do servidor, que permitirá indicar a um utilizador comum o estado do objeto de estudo.

Como *framework*, e após uma conversa com o Professor Pedro Alves, orientador do nosso projeto, decidimos avançar com Flask. Flask é uma *framework* que permite o desenvolvimento rápido de aplicações web em Python, com uma componente API, que nós necessitamos.

Mais à frente no projeto, vamos utilizar uma base de dados relacional. O Sistema de Gestão de Base de Dados a utilizar será o MySQL, por ser um sistema *Open Source*, com muita documentação. Este sistema utiliza uma licença permissiva, que também permite o seu uso comercial, caso o futuro do nosso projeto evolua nesse sentido.

4.2.4 Bibliotecas

Para realizar alguns cálculos ou implementar alguns algoritmos, nomeadamente de funções e transformações matemáticas, como por exemplo as Transformadas de Fourier ou o modelo de Welch, estamos a utilizar algumas bibliotecas pré-feitas, impedindo assim realizar excesso de trabalho.

Estas operações matemáticas estão implementadas no lado do servidor. Tal é possível usando as bibliotecas para a linguagem Python *numpy* e *scipy*.

No lado do servidor, como mencionado anteriormente, serão utilizados alguns algoritmos matemáticos já existentes, em MATLAB, que posteriormente serão convertidos para Python. Esses algoritmos serão fornecidos pelos Professores de Engenharia Civil.

Em relação aos algoritmos de *Machine Learning* e Inteligência Artificial que vamos utilizar, iremos adaptar alguns dos algoritmos utilizados pelos Professores de Engenharia Civil, já programados em MATLAB, baseados em Redes Neurais e *Gaussian Mixture Models*. Tal como os algoritmos matemáticos, serão posteriormente convertidos para a linguagem Python.

4.3 Arquitetura da solução

Para a arquitetura da solução, iremos usar uma abordagem simples, do tipo Cliente-Servidor. Esta arquitetura irá conter quatro elementos principais: o cliente, o servidor, uma pequena API para interação com o servidor, e uma base de dados. Abaixo, na figura 7, é possível observar esta implementação.

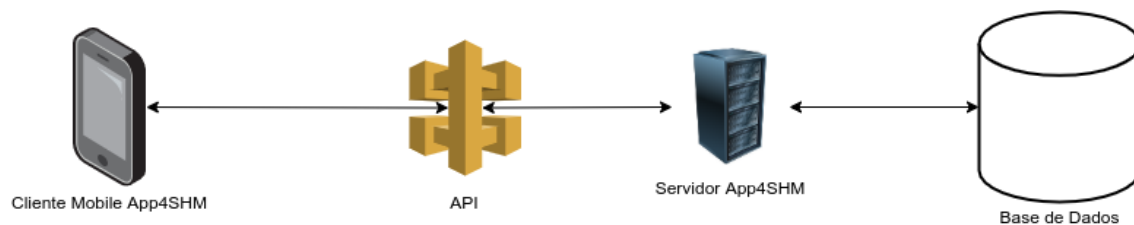


Figura 6: Diagrama de Arquitetura da App4SHM

O cliente, será o objeto de interação que o utilizador final terá. Este permitirá realizar todas as ações que serão especificadas no tópico seguinte, sobre os Casos de Uso.

A API permitirá a interação entre o cliente e o servidor, permitindo assim que o cliente e o servidor estejam em sintonia.

O servidor terá um conjunto de métodos para realizar diversas operações, desde realizar cálculos mais exigentes, até guardar informações na Base de Dados. Será aqui que os algoritmos de *Machine Learning*, e toda a componente de Inteligência Artificial do projeto será implementada.

A base de dados irá guardar informações sobre os utilizadores, por exemplo os grupos em que estão registados e as estruturas que estão a ser monitorizadas, e posteriormente informações sobre as próprias estruturas, como por exemplo os resultados das últimas medições e a saúde estrutural da mesma.

Esta arquitetura irá suportar múltiplos clientes ligados em simultâneo.

4.4 Desenvolvimento Inicial

Inicialmente, foi criada uma aplicação básica que efetuava leituras do acelerómetro. Esta aplicação foi apresentada aos Professores de Engenharia Civil que gostaram, sugerindo apenas a apresentação de um gráfico para fácil interpretação. De seguida, foi implementado o gráfico, os menus na app móvel, o envio de dados, e na componente do servidor, a recepção de dados, com os devidos ajustes na sincronização. Posteriormente tivemos várias reuniões com os Professores de Engenharia Civil para acertar requisitos e perceber as melhores abordagens aos problemas detetados. As versões principais da aplicação coincidem com os *milestones* apresentados no capítulo do Método e Planeamento, apesar de pequenas versões terem sido criadas durante o desenvolvimento para demonstrar as alterações efetuadas. Esta etapa está explicada em detalhe no primeiro anexo deste relatório.

4.5 Discussão de problemas detectados

Após a implementação integral de alguns requisitos, foram detetados alguns problemas. De entre os quais, destacam-se os seguintes problemas, por ainda não terem uma solução viável testada e decidida, encontrando-se em aberto.

4.5.1 Captura de medições

O primeiro problema deve-se ao facto dos dispositivos não começarem a capturar ao mesmo tempo. Isto é problemático porque assim não temos medições exatas a nível temporal de dois locais da estrutura diferentes, podendo gerar dados errados ou inválidos, sem nenhuma utilidade.

Para este primeiro problema existem duas soluções possíveis. A primeira solução consiste na remoção da captura manual dos dados, sendo que é programado um temporizador em ambos os dispositivos para começarem uma medição a uma determinada hora. A segunda solução consiste no alinhamento posterior dos dados, usando um *Timestamp* comum. Ambas estas soluções, de uma maneira ou outra, já são aplicadas na área de Engenharia Civil, em alguns sensores.

No que toca à primeira solução, não existe uma garantia de que a hora do dispositivo está certa, podendo não começar exatamente na hora definida. No entanto, a versão atual da aplicação verifica se o dispositivo tem as definições de Data e Hora marcadas automaticamente, e irá alertar o utilizador caso não tenha. De momento é utilizada uma mensagem *Toast* (Figura 7).

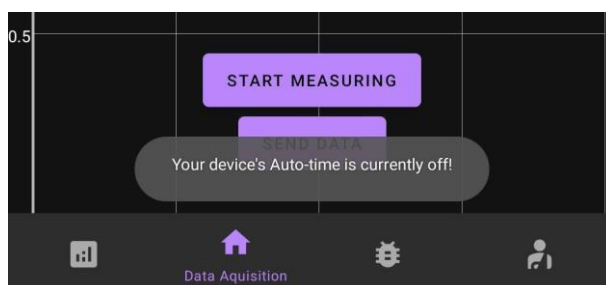


Figura 7: Toast de Aviso ao Utilizador

A apresentação desta mensagem, e a posterior ativação desta opção por parte do utilizador não nos irá garantir que o dispositivo fique com a hora exata, mas irá ajudar a aumentar as hipóteses de ter a hora correta. De tal forma, esta solução ainda necessita de mais testes da nossa parte, para perceber se de facto será viável.

Relativamente à segunda solução, nada nos garante que os dispositivos terão medições em comum, podendo ser difícil aproveitar as medições realizadas. Isto será explicado melhor no tópico seguinte, relativo ao segundo problema.

4.5.2 Cadência de Captura dos dispositivos

Os acelerómetros presentes nos dispositivos móveis, ao contrário dos acelerómetros presentes nas ferramentas profissionais de Engenharia Civil, não têm a melhor das precisões.

Um requisito das medições é que as mesmas sejam realizadas a uma frequência de 50 Hz, algo que está implementado já. Isto resulta, teoricamente, numa medição a cada 20ms. No que toca às ferramentas profissionais de Engenharia Civil, isto é assegurado pelos fabricantes. O mesmo não se poderá dizer dos acelerómetros dos dispositivos móveis.

Nos nossos testes, em vez de termos medições certas de 20ms em 20ms, tivemos, por exemplo, uma medição ao fim de 19ms, a seguinte ao fim de 17ms, a seguinte ao fim de 26ms, etc...

Para resolver este problema, será aplicada uma operação de Interpolação Linear. Inicialmente, após uma conversa com os Professores de Engenharia Civil, ficou decidido que esta operação seria realizada, numa primeira fase, manualmente, no MATLAB, daí a importância de um botão para a transferência de ficheiros. Posteriormente, um ficheiro de dados tratados do MATLAB será enviado, através de outro botão no Website, para que o servidor possa terminar todas as outras tarefas.

Esta solução, do envio manual, não será algo permanente, sendo que posteriormente iremos portar o algoritmo MATLAB correspondente para Python, a nova linguagem utilizada no servidor.

4.6 Solução Final

Atualmente, a solução implementada já produz todos os dados necessários para que um Engenheiro Civil consiga avaliar o estado da estrutura e se a mesma é segura. Isto é feito pela produção de um gráfico com as frequências de ressonância da estrutura. Geralmente, estas frequências situam-se entre os 0.5 Hz e os 20 Hz. Estas frequências atuam um pouco como a impressão digital da estrutura, sendo que se variam muito, algo de errado se passa com a mesma.

4.6.1 Servidor

Do lado do servidor foi implementado o algoritmo de interpolação para resolver o problema mencionado em 4.7.2. Esta decisão foi tomada após testes em MATLAB para verificar a viabilidade desta operação matemática. De tal modo, esta operação para a correção dos dados passou a ser feita no lado do servidor. Assim, o problema da cadência de captura dos dispositivos ficou resolvido.

Foram realizados alguns ajustes de desempenho nas funções existentes, assim como melhorias e correção de erros na interpolação, sendo que todas as operações matemáticas do servidor estão a ter os mesmos resultados das suas variantes em MATLAB usadas atualmente na área, sendo assim já uma alternativa fiel aos métodos usados atualmente na área.

Por último foi implementado e ajustado para nosso uso o algoritmo de Welch, fornecido em Python pelos Professores de Engenharia Civil. Este algoritmo permite calcular as frequências de ressonância, e pode ser considerado o motor de toda a análise estrutural. Este algoritmo está a realizar o seu dever e os resultados estão fiéis ao algoritmo original em MATLAB, a solução utilizada atualmente na área.

Estes resultados, assim como uns testes realizados com a App, foram apresentados aos Professores de Engenharia Civil, que demonstraram satisfação com os mesmos, tendo designado os mesmos de “excelentes”. Os detalhes dos testes realizados serão abordados no capítulo adequado.

Com estas funcionalidades implementadas, foi removida a funcionalidade de upload de ficheiros processados pelo MATLAB, mencionada em 4.7.2, por ser irrelevante a sua existência, dado que os resultados apresentados se demonstraram precisos. De tal modo, o Servidor encontra-se pronto para a implementação da recolha de dados inicial e a posterior implementação dos algoritmos de *Machine Learning*.

4.6.2 Cliente

No lado do cliente, foram efetuados vários ajustes na aplicação, não só para acomodar as novas funcionalidades e o novo gráfico das frequências de ressonância, como também melhorias de desempenho e das leituras, assim como a correção do problema temporal da captura das posições.

Nesta parte, o destaque vai para a apresentação de um gráfico com as frequências de ressonância da estrutura, calculadas no servidor. Um exemplo deste gráfico poderá ser visualizado abaixo, na figura 8.

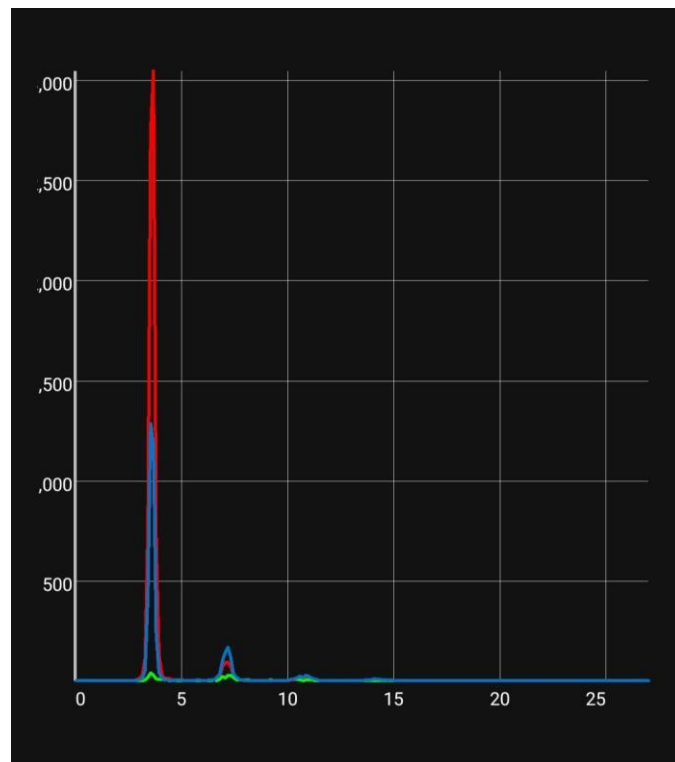


Figura 8: Acelerograma de um teste realizado

Neste gráfico é possível observar 2 frequências de ressonância predominantes, sendo que a principal para a estrutura em questão é a primeira, cerca dos 3.8 Hz. Esta medição foi realizada numa estrutura de testes, que estará descrita no capítulo dos resultados.

Posteriormente foram realizados alguns ajustes nas medições realizadas pelo acelerômetro, com o objetivo de as colocar mais estáveis. No entanto, sendo que os acelerômetros utilizados nos dispositivos móveis não têm tanta precisão como os utilizados para Engenharia Civil, será sempre necessário realizar as operações implementadas no servidor.

No que toca ao problema da captura das medições, foram também realizados testes com smartphones na mesma estrutura para averiguar o intervalo temporal dos mesmos. Aqui, chegamos à conclusão que a não ser que os dispositivos estejam em zonas do país diferentes, ou até mesmo em países diferentes (algo difícil de acontecer, considerando que os dois dispositivos terão que estar na mesma estrutura para realizar a medição), a diferença temporal das medições pode ser negligenciável (cerca de 5ms no máximo), pois a operação de interpolação dos diferentes resultados, realizada no servidor, consegue corrigir com precisão até 20ms de diferença entre os dispositivos.

No que toca ao lado do cliente, os Professores de Engenharia Civil também se encontram satisfeitos com a evolução da aplicação assim como com os resultados obtidos.

5 Resultados

5.1 Interface

Relativamente aos testes realizados à interface da aplicação, obtivemos opiniões, por parte dos Professores de Engenharia Civil, para como poderíamos melhorar a interpretação do gráfico das frequências de ressonância. Os mesmos aconselharam introduzir algum seletor no gráfico que permitisse escolher o eixo das medições, para mais fácil observação. Abaixo, na figura 9, está um exemplo de uma medida, na estrutura de testes, com todos os eixos apresentados.

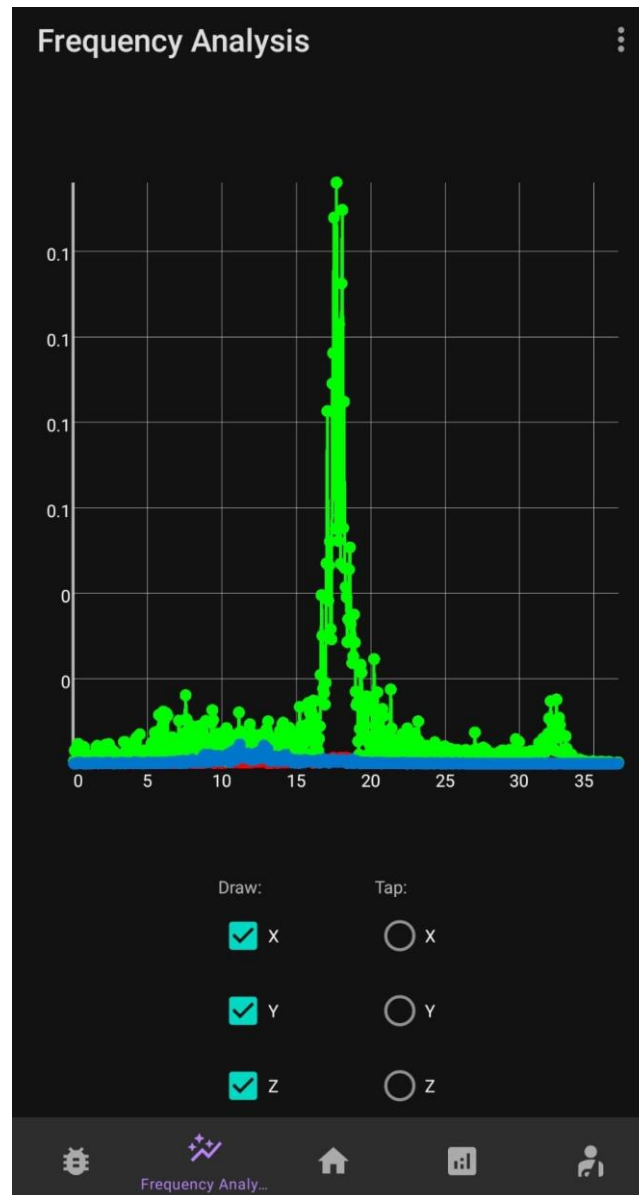


Figura 9: Acelerograma após a medição, com todas as opções

No teste em questão, o eixo das cotas do acelerómetro (a verde) é o que regista os picos da medição das frequências. Como queremos perceber qual é o valor onde

tocamos, um outro conselho dos Professores para melhorar a interpretação do gráfico, incluímos um seletor “*tap*”, para clicar apenas no gráfico pretendido.

Assim, vamos selecionar o *Tap* em Z, excluir os eixos X e Y, e clicar no ponto máximo do gráfico apresentado (figura 10).

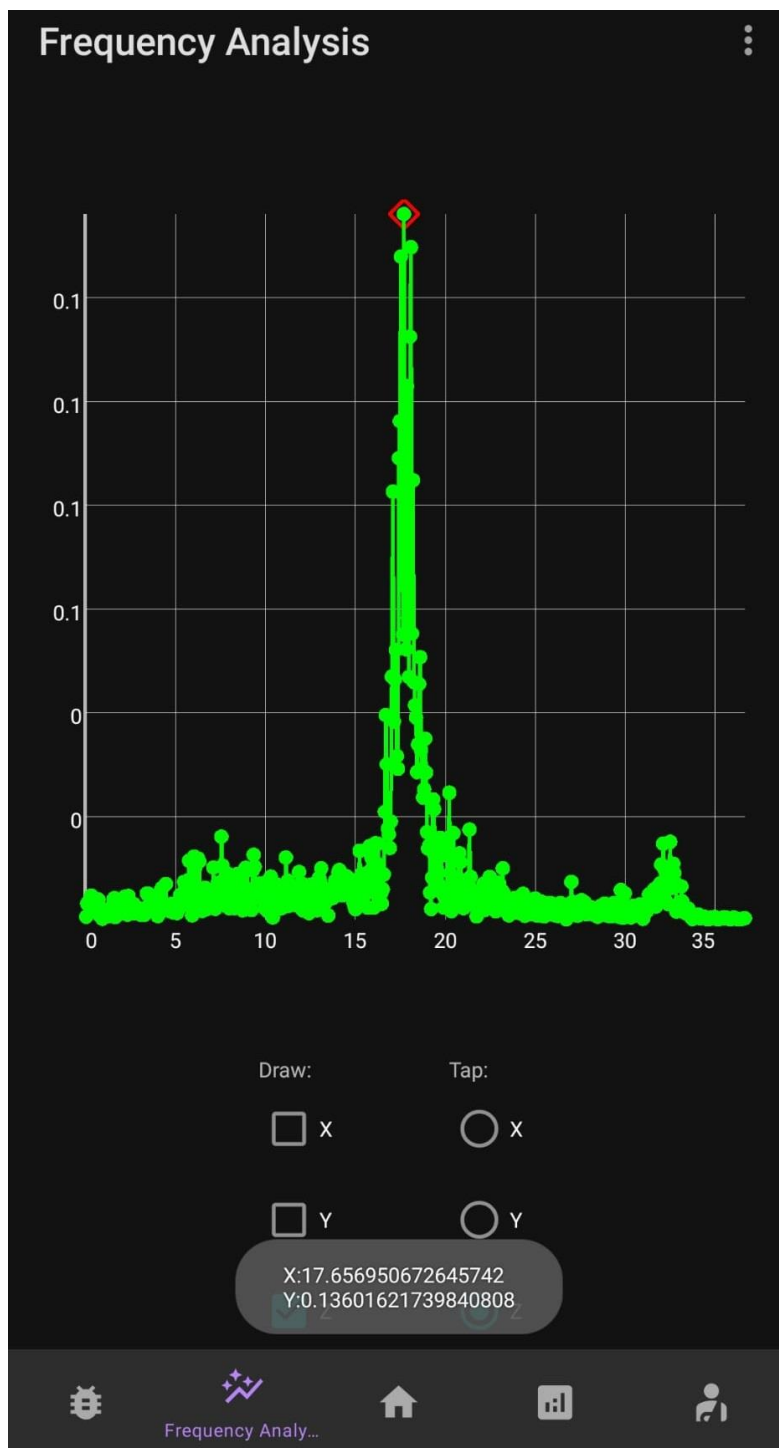


Figura 10: Acelerograma após a medição, mas filtrado

Como demonstrado, aparece apenas o gráfico que desejamos apresentar, e após um clique no ponto máximo do gráfico, temos uma mensagem que indica o valor da frequência (eixo do X), que neste caso é de aproximadamente 17.66 Hz.

Assim, a interface está a seguir os requerimentos dos Professores de Engenharia Civil, sendo que o mesmo é fácil de se usar e apresenta os resultados da análise da frequência.

Posteriormente, mais ajustes e testes de usabilidade serão realizados, para garantir a aceitação desta app por parte de um leque mais extenso de utilizadores.

5.2 Leituras da App móvel

No que toca às leituras efetuadas pela App móvel, as mesmas são realizadas com alguma cadência, como explicado anteriormente no tópico dos problemas detetados. Como mencionado anteriormente, este problema foi corrigido no lado do servidor, usando uma interpolação matemática.

Estes resultados foram apresentados aos Professores de Engenharia Civil, assim como alguns exemplos de medições realizadas. Os mesmos referiram parecer medições válidas. Posteriormente, foram realizados testes numa maquete de uma estrutura construída na propriedade de um dos integrantes do grupo, como apresentado abaixo, na figura 11.



Figura 11: Plataforma de Testes

Esta maquete foi apresentada aos Professores de Engenharia Civil, para verificar se seguia as recomendações dos mesmos. O “tabuleiro” deveria ser algo flexível para termos bons resultados com a curta dimensão da maquete. Posteriormente, os pilares não podiam estar presos à respetiva estrutura, de forma a poder variar a frequência de ressonância da mesma, e ter resultados diferentes nos testes.

Todos os testes foram realizados nesta maquete. Estão planeados testes posteriores num laboratório de Engenharia Civil, para efetuar comparações mais rigorosas com as ferramentas atualmente usadas na área.

5.3 Resultados do Servidor

Inicialmente, após a implementação da componente de interpolação do servidor, foram realizadas várias leituras de teste, na plataforma oscilante acima referida.

De seguida, e recorrendo à ferramenta que permite descarregar os dados do servidor, os dados foram carregados para um script de MATLAB fornecido pelos Professores de Engenharia Civil, que realizava a interpolação dos dados. O ficheiro resultante foi guardado. Posteriormente, descarregamos os ficheiros interpolados do servidor, guardando o ficheiro também.

Estes dados foram colocados numa folha de cálculo, de forma a facilitar a análise. Aqui, foi observado que os intervalos temporais e os valores da aceleração coincidiam em ambos, o que era de esperar, dada a natureza da interpolação matemática. Assim, esta componente foi aprovada e testada.

Posteriormente, de uma forma semelhante, testamos o cálculo da frequência de ressonância, com a função Welch fornecida. Os resultados foram igualmente satisfatórios. Com esta componente realizada, um Engenheiro já consegue utilizar o seu smartphone para realizar medições numa estrutura com a nossa App.

5.4 Ambiente de Testes

Atualmente, a aplicação foi testada numa maquete, sendo que todos os testes foram bem-sucedidos. Posteriormente, de forma a calibrar rigorosamente a aplicação com as ferramentas usadas em Engenharia Civil, serão realizados testes em laboratório próprio do departamento.

Por último, após terem sido implementadas as duas componentes principais finais do projeto, serão realizados testes para acertar os algoritmos de *Machine Learning*. Neste caso, o objetivo do ambiente de testes será detetar falhas graves no algoritmo. Tais falhas podem ocorrer se, por exemplo, o algoritmo disser que uma ponte não vai cair, ou é segura, e os valores serem obviamente inválidos para tomar qualquer decisão ou até mesmo perigosos, podendo colocar vidas em perigo.

6 Benchmarking

Da maneira como a App4SHM está pensada, não existe nenhuma solução, comercial ou não, publicada nas respetivas lojas de aplicações móveis idêntica. No entanto existem diferentes aplicações que fazem algumas tarefas que estarão presentes na versão final.

Tais apps incluem analisadores de vibração (ver exemplo na figura 12), de frequência, de aceleração, de amplitude de oscilação e de amplitude de aceleração.

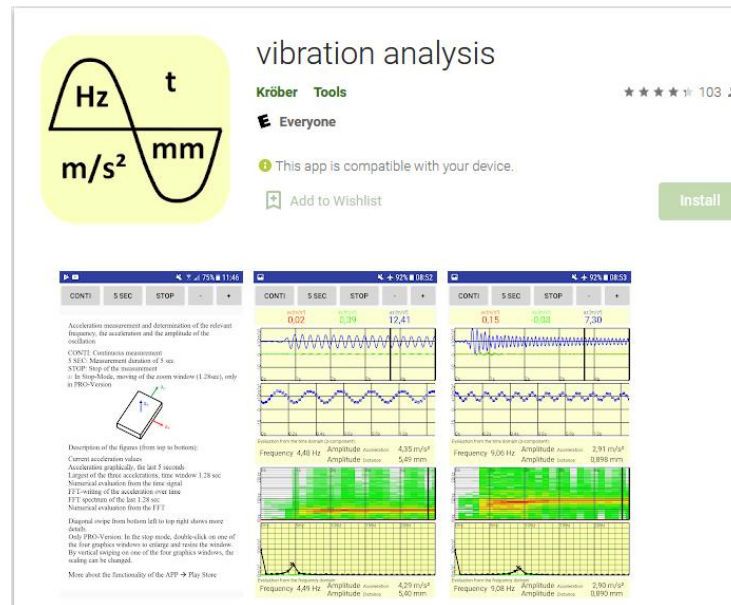


Figura 12: Captura de ecrã da loja da Google, na página de uma das aplicações de análise de vibração

Existe uma diferença principal destas apps para a app que nós estamos a planear. A maior parte das apps disponíveis no mercado apenas fazem leituras dos sensores, limitando-se apenas a apresentar os resultados, sem qualquer informação acrescida.

A App4SHM irá utilizar algoritmos de *Machine Learning* para prever futuros comportamentos da estrutura e avisar o seu utilizador de uma possível falha estrutural iminente. Estes cálculos, devido à sua complexidade, não serão realizados localmente, no dispositivo móvel, visto que tal iria ser muito exigente a nível energético.

Esta aplicação não pretende ser uma substituição de ferramentas próprias da Engenharia Civil para este propósito, visto que os acelerómetros dos smartphones ainda não são tão sensíveis e precisos.

7 Método e Planeamento

7.1 Milestones

Diagrama da aplicação de smartphone para SHM

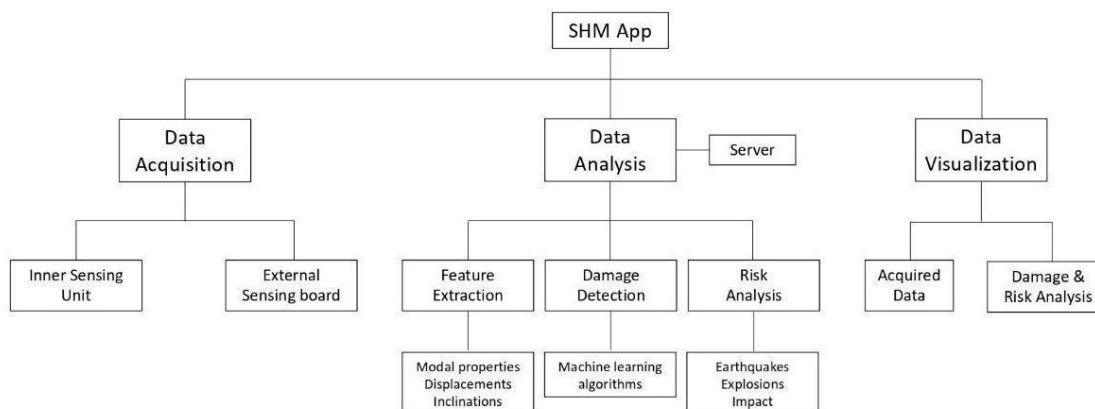


Figura 13: Diagrama de funcionamento da App4SHM

Vamos dividir o desenvolvimento da aplicação em vários *milestones*.

Os *milestones* são para a apresentação à equipa de Professores de Engenharia Civil.

Em que cada *milestone* iremos entregar uma versão do programa com as seguintes funcionalidades:

- **1º Milestone** – liga telemóvel à estrutura, lê os valores através da aplicação com o acelerómetro do telemóvel e mostra no ecrã.
- **2º Milestone** – envia acelerogramas obtidos para o servidor.
- **3º Milestone** – Medições de grupo sincronizadas e enviadas para o servidor.
- **4º Milestone** – Escrita de ficheiros, download dos mesmos do servidor e implementação das leituras de ficheiros do MATLAB
- **5º Milestone** – Implementação dos algoritmos matemáticos de MATLAB, recolha de dados e implementação da componente de *Machine Learning*, verificando se os valores medidos são considerados normais para determinada estrutura. Procedimento matemático para verificar o que aconteceria à estrutura no caso de uma catástrofe.

7.2 Mapa de Gantt

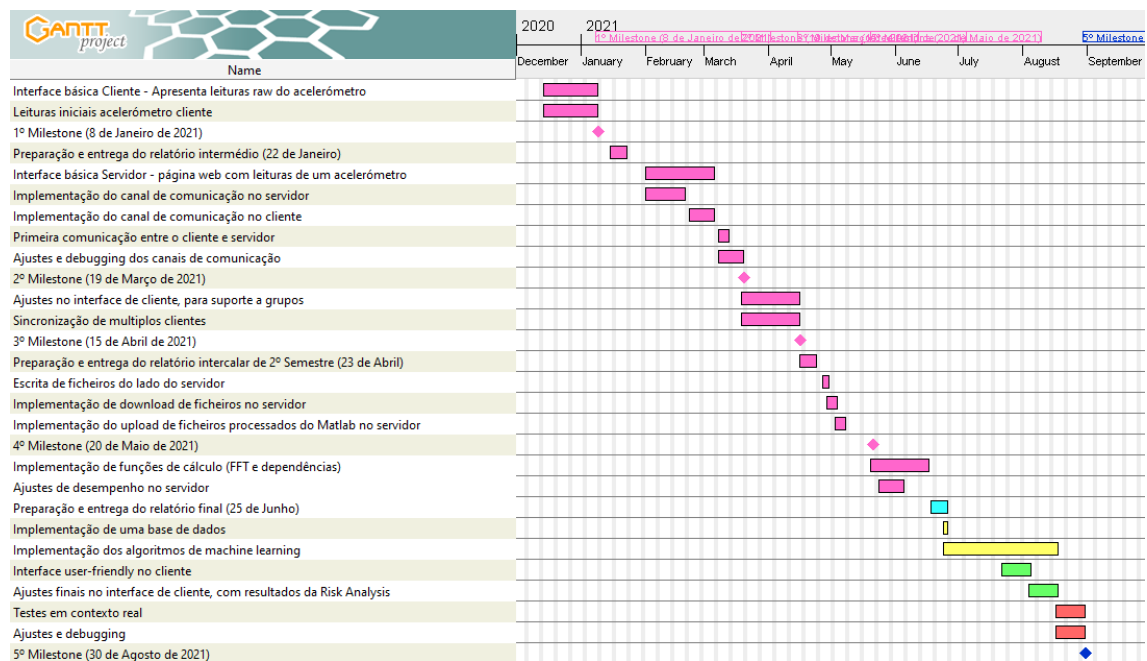


Figura 14: Mapa Gantt do projeto App4SHM

Acima está apresentado o calendário do planeamento das etapas do desenvolvimento do nosso projeto. É importante referir que foi utilizado um código de cores, para distinguir que componente está a ser focada em cada ponto:

- Verde: aplicação do lado do cliente
- Amarelo: aplicação do lado do servidor
- Vermelho: aplicações de ambos os lados
- Azul escuro: *milestones* do projeto
- Azul claro: Períodos de planeamento e entrega dos relatórios
- Magenta: Tarefas e *milestones* atingidos.

No calendário estão também marcadas as estimativas dos prazos dos *milestones* definidos no ponto 7.1 deste relatório. É importante referir que a estrutura deste calendário pode ser ajustada.

Atualmente, encontram-se realizadas as tarefas até ao 4º *milestone*, como apresentado no capítulo da Solução Desenvolvida.

Com este *milestone* concluído, é importante referir que tivemos algumas reuniões com os Professores de Engenharia Civil, de modo a realizar um *brainstorm*, validação de soluções dos diferentes *milestones*, e perceber quais as melhores abordagens para os diferentes problemas que tivemos até agora.

Até à data, todas as tarefas realizadas foram cumpridas dentro das datas inicialmente estabelecidas, e não faltamos ou tivemos atrasos em nenhuma entrega.

Bibliografia

- [1] DEISI, Regulamento de Trabalho Final de Curso, Set. 2020.
- [2] Universidade Lusófona, www.ulusofona.pt, acedido em Nov. 2020.
- [3] *What is Structural integrity*, AeroSIFT, <http://www.aerosift.com/blog/structural-integrity/>, acedido em Nov. 2020.
- [4] Yoo,B., *Failrule is an Option*, <https://www.midasoft.com/bridge-library/failure-is-an-option>, acedido em Nov. 2020.
- [5] Kate Day Sager, Planet Smethport Project, *Tornado blew Kinzua Viaduct down*, <http://www.smethporthistory.org/kinzuaviaduct/viaductcollapse2003/kinzuacollaps e2003.html>, acedido em Nov. 2020.

Glossário

ULHT Universidade Lusófona de Humanidades e Tecnologias

LEI Licenciatura em Engenharia Informática

TFC Trabalho Final de Curso

SHM Structural Health Monitoring

REST Representational state transfer

Anexo 1 - Detalhe do Desenvolvimento Inicial

Solução Inicial

De acordo com o nosso calendário, já entregamos uma primeira versão da aplicação. Esta versão inicial é equivalente ao primeiro *milestone* definido no planeamento.

De uma forma genérica e sucinta, o objetivo deste *milestone* era ligar o smartphone a uma estrutura e apresentar no ecrã os valores da leitura do acelerómetro, que flutuariam dependendo da estabilidade da estrutura.

De uma forma mais técnica, foi realizada uma aplicação móvel básica, que apresentava estas leituras no ecrã do dispositivo móvel, em tempo real. Esta aplicação foi desenvolvida usando a linguagem de programação Kotlin. As leituras retiradas do acelerómetro são apresentadas no ecrã, em tempo real, com um ArrayList de Strings. Deste modo, mantemos sempre um breve registo do registo do acelerómetro.



Figura 15: Implementação inicial da App4SHM

Acima, na Figura 15, está uma captura de ecrã parcial da App enquanto a mesma efetuava medições.

Na captura, cada linha contém informação sobre o tempo de captura, a aceleração medida em relação ao eixo das abcissas (x), a aceleração medida em relação ao eixo das ordenadas (y) e a aceleração medida em relação ao eixo das cotas (z).

As unidades das medições (segundos no caso do tempo, e metros por segundo ao quadrado, no caso das acelerações) foram incluídas também nestas linhas de

informação, para coerência científica. Nesta demonstração, as medições estão a ser efetuadas a uma frequência de 50Hz, correspondendo a uma medição a cada 20 ms.

Mais abaixo, está presente o botão dinâmico responsável por iniciar e parar a medição. O texto neste botão varia dinamicamente dependendo do estado da medição.

Esta versão da aplicação foi apresentada à equipa de Professores de Engenharia Civil, que se encontra a acompanhar o projeto. De uma forma geral, os comentários foram bastante positivos e satisfatórios, sendo que as únicas alterações que eles gostavam de ver realizadas foi a apresentação dos dados, sendo que para uma análise mais fácil, é preferível um gráfico, e a frequência de medição, pois esta estava elevada. A mesma foi, entretanto, reduzida para 50Hz.

Nesta etapa, nós apresentamos os dados em texto corrido, que é passado de uma forma muito rápida, o que se torna difícil de analisar. Para uma análise mais fácil, os Professores de Engenharia Civil gostariam de visualizar os dados num gráfico de acelerações. Foram realizados testes, na passagem pedonal superior junto às instalações da ULHT, utilizando os *smartphones* dos Professores de Engenharia Civil, para demonstrar que qualquer dispositivo compatível pode efetuar medições.

Atualmente, a aplicação está organizada em dois ficheiros, um que contém as funções auxiliares, e outro que faz as ações principais da aplicação. Abaixo, na figura 16, é possível observar a estrutura da aplicação.

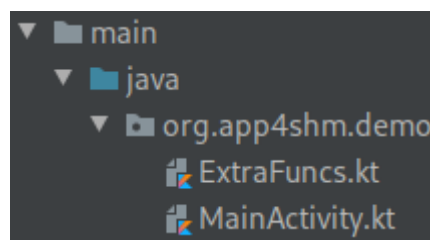


Figura 16: Estrutura de código da App móvel

O código da aplicação encontra-se disponível na plataforma GitHub, acessível no seguinte link: https://github.com/nunopenim/App4SHM_Mobile. Na página das *releases* está disponível o ficheiro APK, executável em dispositivos Android. É possível também visualizar toda a história dos *commits*.

Alterações posteriores da componente Mobile

Posteriormente à apresentação do projeto inicial aos Professores de Engenharia Civil, foram realizadas algumas alterações à app inicial. O aspeto estético da aplicação móvel foi melhorado. Foram também adicionados alguns menus, mas ainda existem alterações pendentes.

Abaixo, na figura 17, está uma captura de ecrã da nova versão da aplicação móvel.

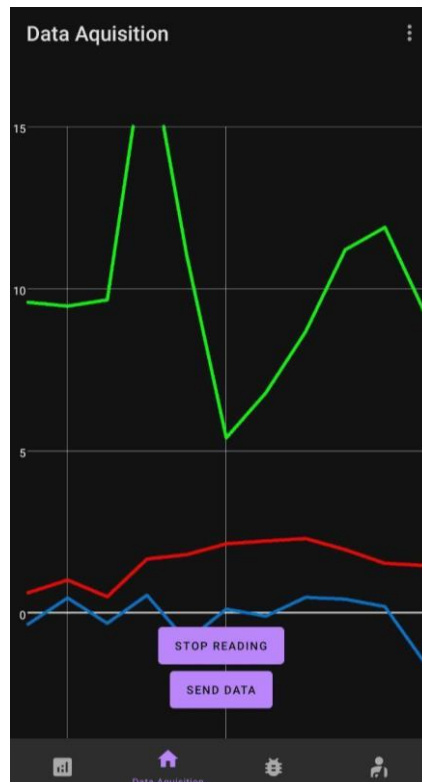


Figura 17: Captura de ecrã da nova versão da App4SHM

Neste novo menu, localizado na barra inferior, foram adicionadas algumas etapas que iremos necessitar para o nosso projeto, enumeradas nos requisitos, assim como a opção de criação de grupos, para realizar medições.

Botões neste menu:

1º- *Risk Analysis*, que após uma conversa com os Professores de Engenharia Civil, será alterado para *Feature Extraction*, numa versão inicial da App.

2º- *Data Aquisition*

3º - *Debug* de momento, mas será alterado para "Mudar Grupos" no futuro

4º - *Damage Detection*.

Na parte de *Data Acquisition*, no lugar da antiga lista de texto, existe agora um gráfico com os valores medidos da aceleração do dispositivo móvel. Esta alteração foi efetuada após uma reunião com os Professores de Engenharia Civil, que referiam que para eles era mais fácil a interpretação de um gráfico, do que uma lista de texto corrido. No que toca a este gráfico, temos uma demonstração de uma versão anterior (que não contém o envio de dados), que facultamos aos Professores de Engenharia Civil, para uma apresentação da área realizada pela empresa Estradas de Portugal. Este video de demonstração pode ser acedido na plataforma YouTube, usando o link <https://www.youtube.com/watch?v=a1Wd4jVv7VY>.

Por último, foi adicionado um botão extra, que poderá ainda ter a sua posição final alterada, para o envio de dados para o servidor. Este botão irá gerar um JSON e enviar para o servidor com o endereço de IP configurado nas definições da aplicação, acessível pelo menu dos 3 pontos, situado no canto superior direito.

A aplicação encontra-se disponível no GitHub, no repositório referido em 4.4.

Servidor

Nesta etapa do projeto, já se encontra também implementada uma versão inicial do servidor da aplicação. A longo prazo, não está planeado o acesso, sem ser por razões administrativas, ao servidor fora da aplicação. No entanto, para algumas demonstrações aos Professores de Engenharia Civil, foi implementada uma página web básica.

Para tal, é ainda importante referir que de momento o servidor da App4SHM está hospedado localmente, na casa de um dos alunos responsável por este TFC. Assim, a disponibilidade do servidor poderá variar, podendo não estar disponível na altura da avaliação deste relatório.

De qualquer modo, a página de diagnósticos pode ser consultada no endereço <http://nunopenim.bounceme.net/diag>. Como este endereço também poderá sofrer alterações, tendo sido criado por conveniência, o servidor poderá ser acedido, usando o seu IP, na página <http://95.94.8.193/diag>.

No que toca à página web do servidor, a mesma é bastante simples de utilizar, sendo que apenas contém um campo onde irão aparecer as leituras e um botão para limpar as leituras, que as limpa universalmente (Figura 18).

A sincronização ao milissegundo em todos telemóveis é impossível com o hardware atual, a solução que encontramos para este problema é agrupar intervalos de *timestamps* e processá-los como se fossem o mesmo *timestamp*

App4SHM Diagnostic Page

Clear screen

ID: nunopenim	group: test	Timestamp: 1619013905522	X: 0.031124622	Y: 3.7469256	Z: 5.8801594
ID: nunopenim	group: test	Timestamp: 1619013905542	X: 0.7230489	Y: 3.715801	Z: 4.745308
ID: nunopenim	group: test	Timestamp: 1619013905562	X: 1.2808979	Y: 3.3207579	Z: 4.733337
ID: nunopenim	group: test	Timestamp: 1619013905582	X: 1.8052281	Y: 2.8395233	Z: 6.1004257
ID: nunopenim	group: test	Timestamp: 1619013905602	X: 1.0366893	Y: 2.0159178	Z: 10.371682
ID: nunopenim	group: test	Timestamp: 1619013905622	X: -0.50278234	Y: 1.2473791	Z: 19.589357
ID: nunopenim	group: test	Timestamp: 1619013905642	X: 0.093373865	Y: 0.9983821	Z: 23.015461
ID: nunopenim	group: test	Timestamp: 1619013905662	X: 0.46447513	Y: 1.3239936	Z: 20.887016
ID: nunopenim	group: test	Timestamp: 1619013905682	X: 1.1204864	Y: 1.7549498	Z: 16.40507
ID: nunopenim	group: test	Timestamp: 1619013905701	X: 0.260968	Y: 1.970428	Z: 12.698846
ID: nunopenim	group: test	Timestamp: 1619013905722	X: 0.36152446	Y: 2.6623523	Z: 8.882488
ID: nunopenim	group: test	Timestamp: 1619013905741	X: 0.61052144	Y: 3.4093432	Z: 6.5218053
ID: nunopenim	group: test	Timestamp: 1619013905762	X: 0.4357447	Y: 4.036624	Z: 4.63278
ID: nunopenim	group: test	Timestamp: 1619013905782	X: -0.09816227	Y: 4.285621	Z: 2.4277205
ID: nunopenim	group: test	Timestamp: 1619013905801	X: -0.32800564	Y: 5.0637364	Z: 0.6751649
ID: nunopenim	group: test	Timestamp: 1619013905821	X: -0.44532153	Y: 5.233725	Z: -0.6368576
ID: nunopenim	group: test	Timestamp: 1619013905841	X: -0.7685388	Y: 4.8458643	Z: -0.93373865
ID: nunopenim	group: test	Timestamp: 1619013905862	X: -0.34476504	Y: 4.663905	Z: 0.3854665
ID: nunopenim	group: test	Timestamp: 1619013905881	X: 0.5674258	Y: 4.5657425	Z: 2.2625206
ID: nunopenim	group: test	Timestamp: 1619013905901	X: 0.47165772	Y: 4.3143516	Z: 4.4939165
ID: nunopenim	group: test	Timestamp: 1619013905921	X: 0.31364042	Y: 3.8187518	Z: 7.156269
ID: nunopenim	group: test	Timestamp: 1619013905941	X: 0.58657944	Y: 3.241749	Z: 10.354922
ID: nunopenim	group: test	Timestamp: 1619013905963	X: 0.81402856	Y: 2.492364	Z: 12.672509

Figura 18: página web demonstrativa da App4SHM

Esta página é propositalmente simples, visto que na versão final do projeto será removida.

O código fonte do servidor está disponível na plataforma GitHub, através do link https://github.com/nunopenim/App4SHM_Server. É importante referir que a versão funcional atual do servidor ainda é em Kotlin, encontrando-se na página das *releases* do respectivo repositório.

A aplicação de servidor só foi testada numa distribuição do sistema operativo Linux. É necessária pelo menos a versão Java 8 para a sua execução. A porta de comunicação usada pelo servidor é a 80, não configurável (para já), sendo que num sistema operativo Linux, poderá necessitar de ser executada como *root/superuser*. De futuro, isto será alterado por se tratar de um risco elevado de segurança.

No entanto, a operação de portabilidade deste código para Python, um novo requisito do nosso projeto, está de momento a decorrer. De tal modo, o código em Kotlin deverá ser consultado na diretoria *Legacy_Kotlin_Code*, presente na raiz do repositório. Esta operação de portabilidade está a decorrer de forma a ser mais fácil posteriormente integrar com scripts do MATLAB, que exportam diretamente para Python, tal como mencionado em 4.2.3

Os dados por defeito, na aplicação mobile, serão enviados para o endereço IP especificado acima, podendo ser alterado usando a página das definições como mencionado anteriormente.

A receção de dados por REST no servidor está completamente implementada, sendo que o próximo passo será escrever ficheiros num formato exemplo (observável na figura 19), disponibilizado pelos Professores de Engenharia Civil, permitir a transferência do mesmo ficheiro, e posteriormente permitir o upload de um ficheiro com resultados do MATLAB, algo que será explicado no tópico seguinte.

```
Station_code  GSGMS
Sampling_rate 100.000000
Start_date    24.02.2018
Start_time    18:00:00.000
Time:sec      11X,g      11Y,g      11Z,g
0.000000000e+00 4.7821462940e-02 -4.1248528426e-02 2.2512009310e-03
1.000000000e-02 4.7507810092e-02 -4.2319292625e-02 1.0977849680e-02
2.000000000e-02 4.7287935207e-02 -4.0145975087e-02 -6.7366485850e-03
3.000000000e-02 4.8184388955e-02 -4.2005639777e-02 3.5126999700e-03
4.000000000e-02 4.6410025124e-02 -4.1537809600e-02 5.4105116280e-03
5.000000000e-02 4.7151241905e-02 -4.1456747293e-02 4.5617415900e-04
6.000000000e-02 4.7336148736e-02 -4.2561949727e-02 8.9867898780e-03
7.000000000e-02 4.7139056068e-02 -4.0394990017e-02 -1.2720954190e-03
8.000000000e-02 4.7507280273e-02 -4.0784406982e-02 -2.2957057270e-03
9.000000000e-02 4.7442112536e-02 -4.1755565209e-02 3.9826494230e-03
1.000000000e-01 4.8027032712e-02 -4.1403235574e-02 4.9903651610e-03
1.100000000e-01 4.6953619418e-02 -4.1353432588e-02 4.5326015450e-03
1.200000000e-01 4.6687650280e-02 -4.0868648203e-02 -1.0543398100e-04
1.300000000e-01 4.6984878739e-02 -4.1192367612e-02 -1.0755325700e-03
```

Figura 19: Formato exemplo fornecido pelos Professores de Engenharia Civil

No Servidor depois de receber os dados vamos utilizar funções em Python transformadas automaticamente de MATLAB fornecidas pelo departamento de Engenharia civil para interpolação linear e posteriormente análise de frequências dos dados