

Projeto de Informática

UNIVERSIDADE DO MINHO

Rosetta Stone



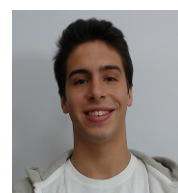
António Gonçalves
a85516



Ricardo Costa
a83870



João Fernandes
a84034



Rui Oliveira
a83610



Rita Rosendo
a84475



Gonçalo Esteves
a85731



Eduardo Conceição
a83870

February 6, 2022

Contents

1	Introdução	2
2	Primeira Componente do Projeto	3
2.1	Levantamento de Requisitos	3
2.2	Modelo de Domínio	8
2.3	<i>Mockups</i>	8
2.4	Interface do <i>Website</i>	12
2.4.1	<i>Backend</i>	12
2.4.2	<i>Frontend</i>	12
2.5	Validação de <i>feed</i> GTFS	15
3	Segunda Componente do Projeto	19
4	Trabalho Futuro	20
5	Conclusão	21
6	Referências	22
7	Anexos	23
7.1	<i>Frontend</i>	23
7.2	<i>Schema</i> do ficheiro de configuração do <i>GTFS_Val</i>	25

1 Introdução

No âmbito da Unidade Curricular de Projeto de Informática, foi-nos proposta a elaboração de um projeto, em parceria com a empresa Transportes Urbanos de Braga - TUB - que consiste, muito resumidamente, na criação de um *middleware* que permite o tratamento da informação fornecida assim como a apresentação da mesma ao cliente de forma clara e lógica em diversos sistemas, quer seja através de TOTEMs (informação estática nos abrigos da cidade), *website* ou aplicativos móveis. Para além disto, será necessário efetuar um estudo geral de forma a situar a empresa no que toca às capacidades de outros fornecedores de serviços de transporte. Para isso deveremos contactar várias empresas para identificar a sua situação, bem como o nível de conhecimento técnico dos seus trabalhadores.

Atualmente, as empresas possuem sistemas nucleares como soluções de planeamento, soluções de bilhética e soluções de apoio ao exterior que permitem materializar toda a relação comercial com o cliente, realizando um controlo em tempo real do que a empresa está a fazer, e assim consegue apresentar a informação necessária ao público de forma clara e simples.

Nos dias que decorrem, a TUB utiliza uma solução de planeamento que realiza *exports* diários, num formato *standard* GTFS, que fornece toda a informação prevista para os próximos 7 dias. Assim sendo, o *export* realizado pela plataforma possui:

- *Standard* próximo do GTFS
- Previsão dos próximos 7 dias, apresentando toda a oferta possível ao cliente;
- Segmentação dos diversos dias como dias da semana e feriados, possuindo *service id* específicos;

Posto isto, a empresa organiza esta informação presente no sistema de planeamento, integrando a mesma em soluções de informação ao público como, por exemplo, o *website* dos TUB e sistemas de informação estáticos dos abrigos da cidade (TOTEMs). Estes dados são também incluídos na plataforma *Google Transit*, permitindo assim o planeamento de rotas a qualquer utilizador na solução *Google Maps*. Esta opção está também disponível na aplicação e no *site*, sendo o utilizador posteriormente direcionado para o próprio *Google Maps*.

Sendo assim, a primeira componente deste projeto baseia-se na criação de uma solução, que através do sistema de planeamento, permita gerir e desenhar de forma clara e lógica toda a oferta dos TUB em diversos sistemas assim como o tratamento da informação para que se possa integrar a mesma de forma simples na plataforma da *Google Transit*. Por fim, a segunda componente deste projeto passará por realizar uma análise de mercado que estime o impacto desta solução verificando se esta solução é adequada tendo em conta os requisitos dos diversos operadores.

2 Primeira Componente do Projeto

Tal como foi mencionado anteriormente, parte da primeira componente do projeto baseia-se na criação de uma solução que permita gerir e desenhar de forma clara toda a oferta dos TUB em diversos sistemas quer seja através de TOTEMs, *website* ou aplicativos móveis. Atualmente os TUB disponibilizam a informação nos vários sistemas mencionados, porém esta pode não estar a ser apresentada da forma mais clara e portanto o nosso objetivo passa por propôr uma apresentação mais perceptível e simples. Mais ainda, a primeira componente também inclui a construção de um *software* de validação dos *exports* dos ficheiros em formato GTFS, que consiga apresentar os resultados de uma forma mais legível.

Para tal, e de forma a identificar alguns dos problemas já existentes, decidimos criar um questionário para recolher a opinião de diversos clientes para perceber quais as principais dificuldades na análise da informação exposta. Este foi distribuído pelo email institucional, tendo obtido um total de 101 respostas.

2.1 Levantamento de Requisitos

Como primeiro passo do nosso trabalho foi necessário fazer um levantamento de requisitos da interface do *site* dos TUB.

Em primeiro lugar, o grupo começou por reunir e discutir que aspetos podem ser melhorados no *website*, para isto foram observados outros exemplos de serviços de transportes, como por exemplo, TUG, AVIC, Transdev, CP entre outros. Posteriormente, desenvolvemos um questionário direcionado à população académica para perceber como é que a mesma interage com os serviços dos TUB.

O principal objetivo deste questionário passava por identificar as maiores dificuldades sentidas pelos utilizadores dos serviços, colocando-se também algumas perguntas com o intuito de melhor situar a amostra de pessoas que responderam (questionou-se, por exemplo, a idade e local de trabalho sendo que, tal como era expectável, a maioria das pessoas que responderam afirmaram ser estudantes da zona de Braga). Pedimos também para referir, caso já não usassem o serviço, a razão para tal, tendo obtido diversas respostas como problemas de horários, a pontualidade e rapidez do serviço, entre outros. Para além disso, fizemos várias perguntas sobre possíveis alterações no que toca à disponibilização da informação ao cliente no *website*, aplicação e TOTEMs. Foi possível ter em conta algumas das respostas obtidas na realização do trabalho, contudo algumas das sugestões obrigariam a alterações por parte dos sistemas da TUB, como por exemplo, indicar em tempo real a posição dos autocarros, permitindo saber ao certo quanto tempo demorariam a chegar a uma dada paragem. Após o fecho do formulário, procedemos à análise das respostas do mesmo, sendo as mais relevantes apresentadas de seguida.

Em relação aos autocarros utilizados, verificou-se que o autocarro 43 (Estação - Universidade do Minho) é o mais utilizado pela amostra estudada, algo facilmente explicável tendo em conta que a maioria das pessoas que responderam são estudantes.

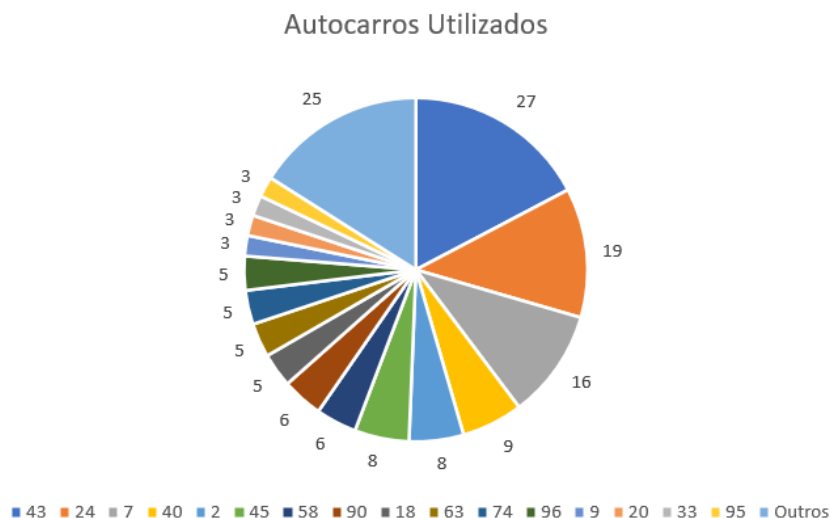


Figure 1: Respostas obtidas sobre autocarros utilizados

Outro facto interessante depreende-se no modo mais utilizado para obter a informação, sendo este a App TUB, o que faz sentido, tendo em conta que cada vez mais a população possui aparelhos móveis capazes de utilizar este tipo de programas.

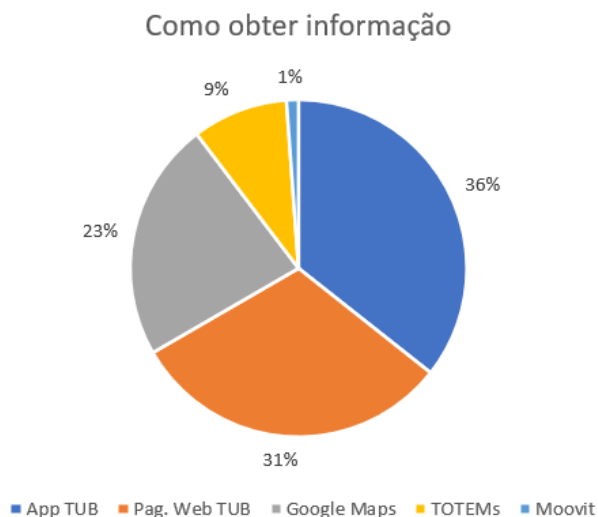


Figure 2: Respostas obtidas sobre como é obtida a informação

Relativamente aos horários de utilização dos serviços da TUB, a maioria respondeu que costuma usufruir destes durante os dias úteis, na parte da manhã. Mais uma vez, isto é algo facilmente compreensível, já que estas horas representam aquelas em que é previsível uma maior necessidade de deslocação da amostra estudada (a maior parte dos cursos da universidade têm aulas durante a manhã).

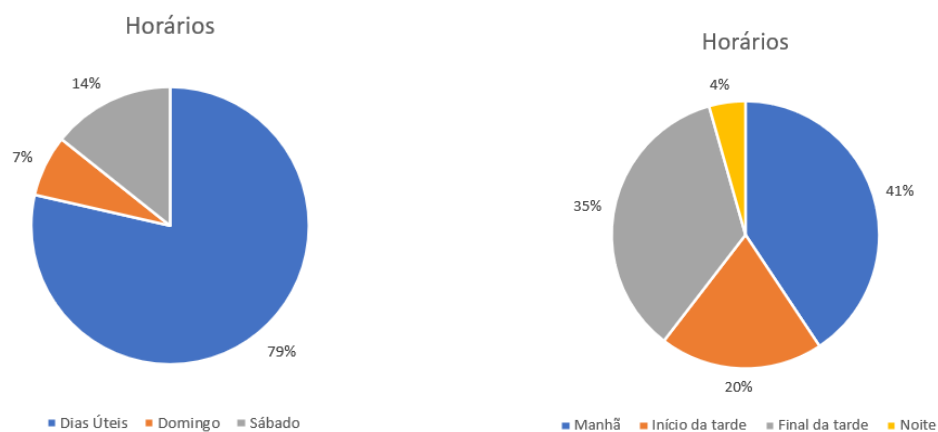


Figure 3: Respostas obtidas sobre quais os horários utilizados

Relativamente aos motivos para não utilizar os serviços dos TUB, o motivo mais mencionado é a falta de rapidez dos mesmos.

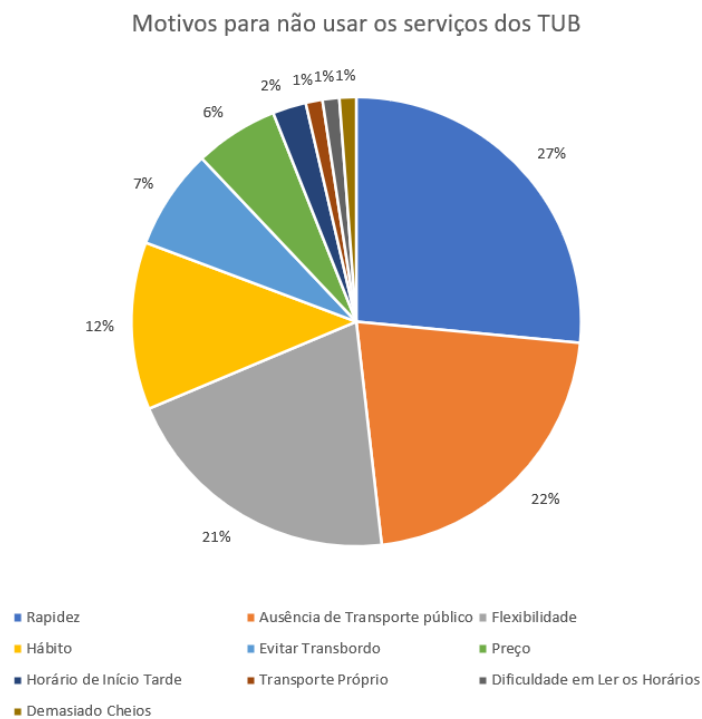


Figure 4: Respostas obtidas sobre motivos para não usar os autocarros

Por fim, no que diz respeito às dificuldades que os clientes encontram quando procuram obter informação, tanto na aplicação, como nos TOTEMs ou no *site*, obtivemos várias respostas, sendo apresentado um resumo geral das mais ditas em seguida:

- No *site*, apenas aparecem alguns horários por paragem;
- Má organização da informação;
- Por vezes os horários estão desatualizados;
- Não tem explicação do roteiro;
- Falta do horário de paragens específicas nos TOTEMs;
- Horários ilegíveis e demasiado pequenos;
- Existem paragens em que o autocarro não para às vezes;
- A legenda é pouco clara;
- Horários apresentados por vezes não coincidem com os verdadeiros;
- As placas que indicam o tempo até chegar o próximo autocarro não funcionam;
- Não aparecem todas as paragens na App TUB;
- Imprevisível saber se os autocarros já passaram ou não;

Mais uma vez, algumas das sugestões foram aproveitadas para a implementação do trabalho. Mesmo as perguntas menos relacionadas com o objetivo do trabalho foram interessantes, uma vez que estas serviram para apresentar dados relevantes à empresa. Algumas das críticas aqui apresentadas, no entanto, apesar de poderem ser ideias interessantes, são impossíveis de implementar tendo em conta o estado tecnológico dos TUB, nomeadamente:

- Aspectos que envolvam saber a localização de autocarros em tempo real;
- Aspectos relacionados com as placas nas paragens;

Estes aspetos fogem do âmbito do projeto e, como tal, não foram abordados.

2.2 Modelo de Domínio

Tendo em conta a informação disponibilizada atualmente pelos TUB, definimos o seguinte modelo de domínio para a nossa solução, onde podemos sintetizar as relações entre os vários componentes que os TUB utilizam de forma a uniformizar a nomenclatura utilizada para mostrar informação:

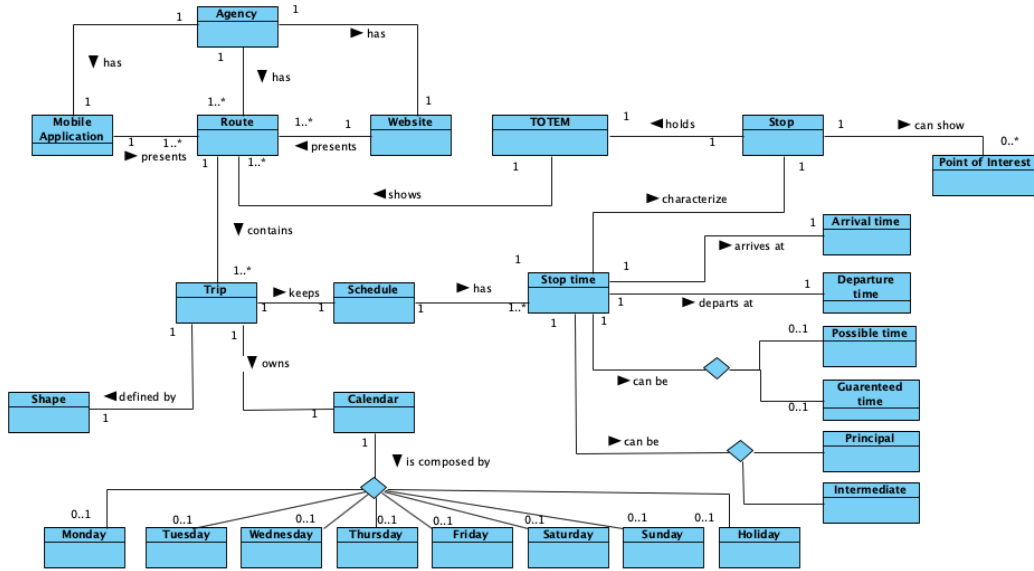


Figure 5: Modelo de Domínio da Solução

2.3 Mockups

Após analisarmos as principais dificuldades dos clientes no que toca à interpretação da informação prestada pelos TUB, decidimos propôr algumas modificações na apresentação da mesma. Para isto utilizamos a aplicação *Axure*, de modo a desenvolver *mockups* que nos permitissem ilustrar melhor as nossas ideias.

No que diz respeito aos TOTEMs (os placards informativos disponibilizados em cada paragem), propomos uma alteração na forma como é feita a distinção entre horários diurnos e noturnos, passando a utilizar símbolos ilustrativos desta diferença ao invés das mudanças de cor que eram até agora utilizados, para além da adição de nova informação que poderá permitir ao

utilizador ter uma melhor noção de onde está. Isto passa por acrescentar informação relativa à localização do utilizador tendo em conta cada um dos percursos que passa por aquela paragem bem como indicar quais os principais pontos de interesse próximos da paragem em questão. Sugere-se, também, o acrescento de informação relativa aos pontos de interesse próximos de cada percurso, isto é, pontos de interesse próximos de paragens por onde o percurso passa.

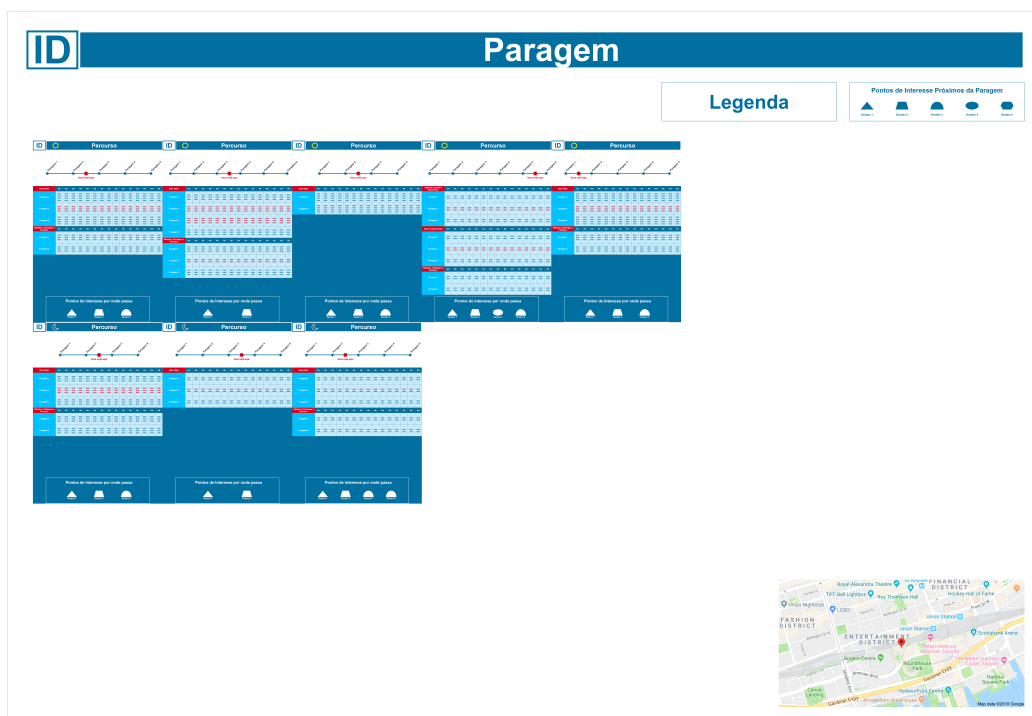


Figure 6: *Mockup* desenvolvido para os TOTEMs

Relativamente ao *website* dos TUB, tentou-se abordar uma solução mais *user friendly*, que permitisse consultar a informação de forma mais eficaz e imediata. Para isto retiraram-se os PDFs que até agora eram disponibilizados, apresentando a informação neles contida de forma mais apelativa: todas as paragens passam a estar presentes na tabela de horários, sendo que as paragens principais aparecem imediatamente enquanto que as paragens intermédias estão ocultas e são apresentadas quando o utilizador clica na

opção que lhe permite expandir o horário; é apresentada uma maior quantidade de horários estimados de chegada a paragens, estando estes devidamente assinalados, podendo servir como orientação para o utilizador apesar de não serem completamente confiáveis (atrasos no trajeto podem levar a que estas horas não sejam cumpridas); e o mapa do percurso é apresentado imediatamente, passando também a ser um mapa "dinâmico", onde podemos seleccionar o percurso a apresentar para um determinado horário (isto é particularmente útil em casos onde o percurso varia de horário para horário, devido às diferentes paragens por onde passa). São também propostas outras pequenas alterações, nomeadamente no que toca à disposição da legenda e aos símbolos utilizados, de forma a tornar a leitura da informação mais simples.

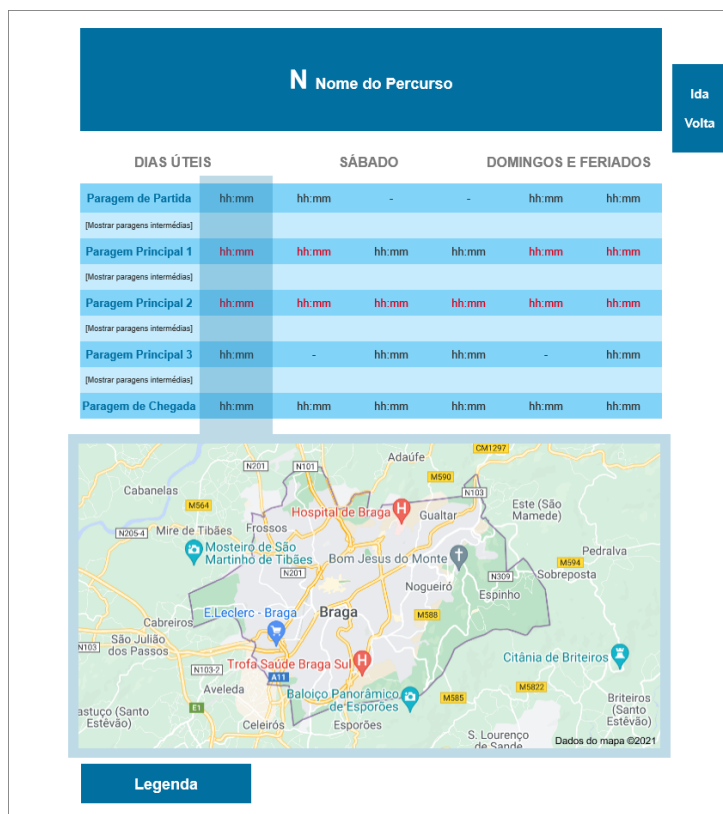


Figure 7: *Mockup* desenvolvido para o *website*

Por fim, apresentamos também alterações relativas à aplicação dos TUB.

Estas visam a aproximar mais a *app* do *website* e, portanto, são em grande medida semelhantes às apresentadas para este último: disponibilização do horário com todas as paragens existentes, estando as intermédias ocultas inicialmente; maior quantidade de horários expectáveis apresentados; e mapas relativos a cada um dos diferentes trajetos que um percurso pode ter.

TUB				
2 Ponte de Prado Bom Jesus			IDA VOLTA	
Dias Úteis	Sábado		Domingos e Feriados	
Ponte de Prado (Marginal)	06:00	-	06:45	07:15
[MOSTRAR PARAGENS INTERMÉDIAS]				
Merelim S. Roque (25 de Abril)	-	-	-	-
[MOSTRAR PARAGENS INTERMÉDIAS]				
Caires	06:20	-	-	-
[MOSTRAR PARAGENS INTERMÉDIAS]				
Liberdade (25 de Abril)	06:25	06:40	07:10	07:40
[MOSTRAR PARAGENS INTERMÉDIAS]				
João XXI (C Amarante)	-	-	-	-
[MOSTRAR PARAGENS INTERMÉDIAS]				
Bom Jesus	-	07:00	07:30	08:00
Código do Mapa	M1	M2	M3	M4
Legenda <i>"Horário em itálico"</i> com passagem nesta paragem, com horário incerto "-" sem passagem nesta paragem				



Figure 8: *Mockup* desenvolvido para a App TUB

2.4 Interface do *Website*

Para implementarmos uma interface do *website* tendo em conta aquilo a que nos propusemos inicialmente e também a algumas das sugestões obtidas através do formulário já descrito, tivemos primeiro de implementar um *backend* simples para servir de fonte de informação para o *frontend*. Inicialmente os dados estavam disponibilizados em ficheiros vindo do *4planning*, que é o sítio onde conseguimos obter o *feed* GTFS. Para melhor trabalhar com os vários dados necessários para implementar o *frontend*, decidimos então que seria melhor criar um *backend* para armazenar esta informação e conseguirmos aceder-lhes de forma mais simples.

2.4.1 *Backend*

O *backend* aplicacional foi desenvolvido com o intuito de mitigar os problemas que tivemos com o tempo que demorava a carregar a informação. Este acabou por não afetar positivamente de forma significativa, mas decidimos mantê-lo, pois, para questões de teste e demonstração, ofusca o tempo de carregamento dos dados.

Para a geração da API de dados utilizamos então a ferramenta *Strapi*, que nos poupou bastante trabalho, já que permite gerar a API de forma bastante simples. Em relação à base de dados utilizada foi o *PostgreSQL*, que é uma ferramenta *open-source* e gratuita de gestão de bases de dados orientada a objetos que se foca na extensibilidade e na conformidade com o SQL.

Para usar esta ferramenta é necessário criar um administrador que contém o acesso ao *backend* e, portanto, contém os privilégios necessários para modificar as estruturas de dados do *strapi*. Este último ponto é importante mais para a frente já que foi necessário adicionar um parâmetro adicional à estrutura de um dos ficheiros originais, para depois no *frontend* conseguirmos associar um caminho no mapa a um dado horário na tabela.

2.4.2 *Frontend*

Relativamente ao desenvolvimento do *frontend*, utilizamos a *framework* *Vue*, que é também *open source* e permite gerar interfaces de uma forma bastante cómoda, já que possui uma curva de aprendizagem relativamente baixa e uma documentação extensa e bem trabalhada. Para além disto, é uma ferramenta com a qual já tínhamos alguma familiaridade, o que permitiu tornar o processo de desenvolvimento mais rápido.

No *frontend*, criamos algumas estruturas necessárias para conter a informação das paragens, dos horários e de notas entre outras coisas relativas a uma dada carreira. É relevante também informar que o primeiro carregamento do *website* ainda é bastante demorado devido a enorme quantidade de dados entre as várias estruturas com que estamos a trabalhar, já que é inicialmente necessário fazer o *parse* desta informação. Uma das técnicas mais comuns utilizadas em outros *websites* como o *lazy loading* não foi possível neste caso, dado que como elaboramos apenas um protótipo, apenas temos uma única página e não é possível ir carregando informação enquanto o utilizador está numa página de *home*, por exemplo. Apesar disto, depois de a informação estar carregada da primeira vez, o *reload* da página irá ser muito mais rápida pois guardamos as informações anteriormente carregadas na *cache* do *browser*.

Em relação à interface propriamente dita, disponibilizamos novas funcionalidades que não se encontram no *website* atual dos TUB como:

- opção de mostrar paragens intermédias entre paragens principais;
- disponibilização de horas previstas para essas paragens intermédias;
- um mapa dinâmico no final da página com um caminho visível e marcadores relativos às paragens principais;
- a associação entre um determinado horário (ou seja uma coluna da tabela) e o respetivo percurso visível no mapa, através de códigos que se encontram disponibilizados no final da tabela para cada coluna;
- uma legenda mais clara, com informação também de notas adicionais relativas a um dado percurso;
- opção de carregar numa paragem principal na tabela e esta se encontrar destacada no mapa com um marcador diferente, indicando ao utilizador a localização de tal paragem;
- uma marcação na tabela com "P" ou "C" indicando claramente se para dado percurso essa paragem é a de partida ou de chegada.

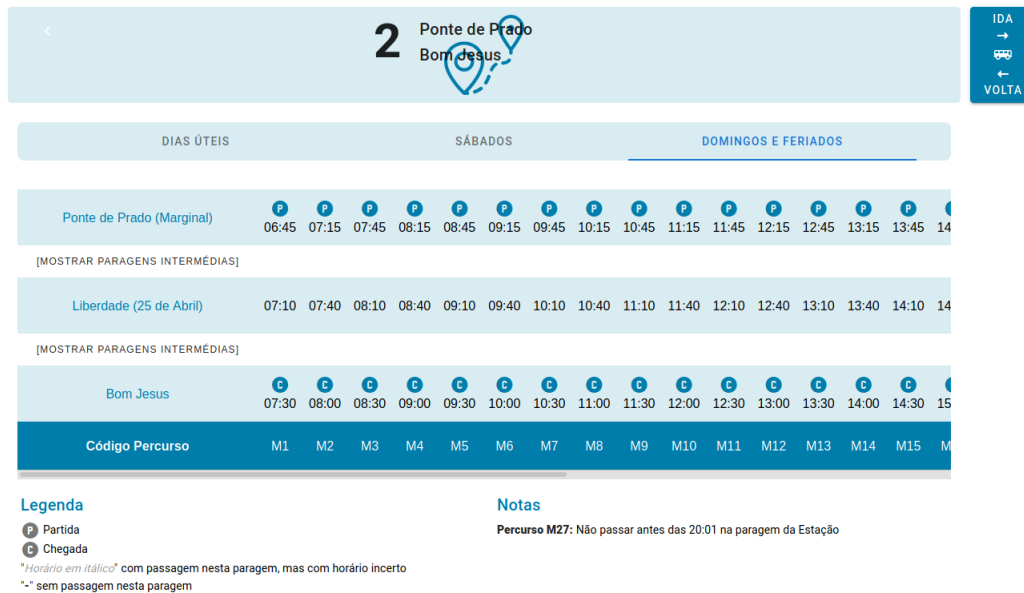


Figure 9: Primeira parte *Frontend* (Tabela dos horários)



Figure 10: Segunda parte *Frontend* (Mapa da Google)

Consideramos que a funcionalidade adicional principal é o mapa disponibilizado no final da página e a respetiva interação entre este e a tabela de horários. Além disto, também o facto de agora ser possível verificar o caminho relativo a um dado horário, já que em alguns casos, o percurso de dado

autocarro não é sempre igual em todos os horários que ele realiza, sendo este um ponto de alguma confusão para os utilizadores. Apesar disto, é importante referir que isto apenas ficou disponível para uma carreira de exemplo, pois a informação disponibilizada pela plataforma *4planning* e com a qual temos de trabalhar não possuía uma associação entre o *shape_id* e o *trip_id*, que era o que permitiria associar um caminho específico para cada coluna da tabela. Para mostrarmos esta funcionalidade, já que de facto ela é possível na nossa implementação, adicionámos uns exemplos de *shape_id* para a carreira 2. Se considerarmos que esta informação eventualmente se encontrará disponível, esta implementação irá permitir que isto fique disponível para todas as carreiras, sendo isto, como é óbvio, algo fora do alcance da nossa equipa.

2.5 Validação de *feed* GTFS

A segunda parte da primeira componente do projeto consiste no desenvolvimento de uma aplicação que permita validar *feeds* GTFS estáticos[1]. GTFS (*General Transit Feed Specification*) consiste num *standard* de representação de informação de trânsito. Duas variantes diferentes existem, estática e em tempo real, sendo que neste projeto concentramo-nos na variante estática, que será utilizada para representar toda a informação referente à oferta dos TUB, num determinado período de tempo.

A aplicação denomina-se *GTFS_Val*.

O objetivo principal desta componente será validar o *feed* disponibilizado periodicamente pelos TUB num respetivo repositório *4planning*. Depois da validação, e se o *feed* se encontrar em condições, decorre o depósito do mesmo num servidor FTP, onde poderemos agendar recolha periódica por parte dos serviços da *Google*, de modo a disponibilizar a oferta dos TUB nos próximos 6 meses no *Google Maps*. Sendo assim, esta aplicação será apenas usada num *back office* dos serviços TUB.

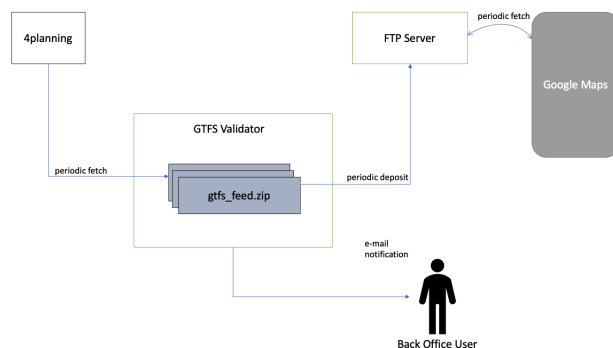


Figure 11: Arquitetura simplificada da aplicação

A implementação desta aplicação tem fortes benefícios para a produtividade da equipa dos TUB, nomeadamente:

- Substitui trabalho manual periódico de validação do *feed* através de uma aplicação auxiliar por trabalho automatizado;
- Permite que o *feed* submetido seja sempre livre de erros, o que aumenta a confiabilidade do fornecedor do *feed* GTFS aos olhos do *Google Maps*;
- Facilita a correção de erros através da automatização da correção dos erros mais frequentes, nomeadamente a codificação dos ficheiros não ser a correta.

A implementação do *software* é de relativamente simples compreensão. Para que o mesmo funcione, é necessário iniciar o programa com um ficheiro de configuração JSON com meta-dados relevantes para o programa, nomeadamente: os recetores das notificações dos relatórios de erros, o endereço de *email* utilizado para enviar as notificações, o *host* e utilizador do servidor FTP onde será depositado o *feed* validado, uma frequência de atualizações do servidor FTP, o dia de início de atualização e a fonte do *feed* GTFS. A *password* do utilizador FTP e do *email* (que tem de ser um *email* registado no *GMail*). Em anexo podemos encontrar o *schema* formal do ficheiro de configuração, e a aplicação vem com um ficheiro de exemplo de modo a facilitar o seu primeiro uso.

Periodicamente, às 23:59h dos dias no período especificado, o *software* irá fazer *fetch* dos ficheiros à fonte (que pode ser local ou remota), e validará os mesmos. A escolha da hora deve-se ao facto que o *4planning* é atualizado

às 23:00h pelos TUB. Damos uma hora de espera de modo a ter em conta possíveis atrasos.

Os erros que podem ser corrigidos são os seguintes:

- Erros de codificação;
- Campos vazios no ficheiro *agency.txt*, um dos ficheiros obrigatórios do *standard*. Apenas este ficheiro pode ser corrigido automaticamente pois é o único que contém informação que, em princípio, não mudarão ao longo do tempo.

Todos os outros erros, fatais ou não, que possa ter são reportados via uma notificação por *email*. A correção de erros é limitada apenas pela informação original. Não é possível corrigir erros de omissão de informação (que constituem a grande parte de erros nos sistemas reais), uma vez que o *software* não pode, de forma alguma, inferir a informação em falta. Informação repetida, na maior parte das tabelas, por outro lado, não constitui um erro fatal, mas apenas disparará um aviso. Como estes avisos não impedem a entrega e processamento dos dados por parte do *Google Maps*, não é necessário corrigi-los. No entanto, surgem no relatório enviado por *email*.

Caso o ficheiro não apresente erros fatais, é enviado para o servidor FTP, para que possa mais tarde ser recolhido pelo *Google Maps*. Isto requer, claro, coordenação com o serviços do *Google Maps*, mas isto é algo cujo *setup* não é pertinente ao projeto em questão, pois o mesmo já apresenta uma interface objetiva e de fácil uso no *website* do mesmo.

Em termos de *performance* do *software*, esta não é um aspeto crítico do mesmo, pois é um sistema que apenas terá de atuar, em princípio, uma vez a cada semana, ou com menos frequência ainda. Isto quer dizer que não há problema se as operações demorarem ligeiramente mais do que esperado. No entanto, mesmo assim, a operação é relativamente rápida, não demorando mais do que 5 segundos entre o pedido do ficheiro e o envio do mesmo para o servidor FTP (testado para um ficheiro de 4.4MB, utilizando um *Macbook Air 2015*[2]).

No entanto, este *software*, de modo a funcionar, precisa de estar sempre ligado, num servidor dos TUB, em *cloud*, etc. Os recursos que o sistema gasta são bastante pequenos, sendo negligenciáveis durante o *idle time* entre atualizações do *feed*. No entanto, durante a mesma, já não o serão, uma vez que necessitam de guardar dois *feeds* GTFS em memória (de modo a

comparar dois *feeds* enviados em períodos consecutivos) e, visto que um *feed* pode ser relativamente pesado, pode ocupar uma porção não negligenciável de memória. No entanto, no contexto de um servidor de um serviço como os TUB, será uma quantidade irrisória, na ordem de algumas dezenas de MB, não mais.

Em termos de interface gráfica, não apresenta nenhuma, uma vez que apenas interage com o utilizador do *back office*, que será um funcionário dos TUB, através de notificações por *email*. Caso qualquer uma das informações passadas inicialmente no ficheiro de configuração sejam alteradas, será necessário reiniciar a aplicação.

No que toca à implementação do sistema em si, esta foi feita em *Python*, uma linguagem que fornece imensos *packages* adequados ao tratamento deste tipo de dados. A interação entre os vários *packages* (pré-existentes ou implementados por nós) é descrita no seguinte diagrama:

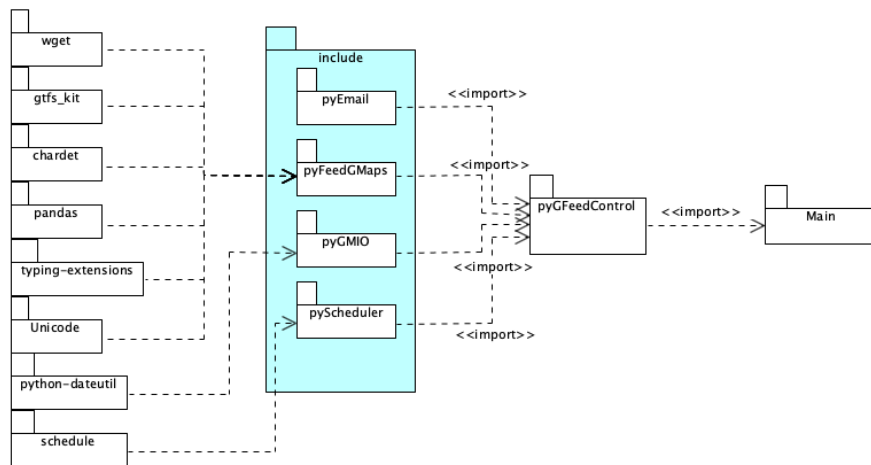


Figure 12: Diagrama de *Packages* da Implementação

Todos os *packages* do módulo *include* e os à sua direita no diagrama foram implementados por nós, sendo que os à esquerda são dependências. Todas estas dependências precisam de ser instaladas para garantir o funcionamento da aplicação, e encontram-se descritas no ficheiro de requisitos *requirements.txt*.

Por último, é importante referir a informação que o *software* pode realmente adicionar ao *feed*. Esta informação é composta pela oferta dos TUB

para os próximos seis meses, e é gerada com base na informação no ficheiro *calendar_dates.txt* (outro ficheiro que faz parte do *feed*). Sendo que é esperado que este carregue a informação para uma semana, a contar do dia em que é submetido, apenas é necessário estender essa informação para os próximos seis meses. A informação mais relevante é um par, constituído pelo dia do serviço e o identificador único do serviço.

3 Segunda Componente do Projeto

A segunda componente do projeto seria suplementar à primeira, passando por um estudo de mercado das várias empresas que oferecem serviços de transportes, avaliando os sistemas que os mesmos têm, de modo a entender como é que poderíamos melhor adequar as nossas soluções ao panorama geral das operadoras de transportes e não exclusivamente aos TUB.

Para este efeito, contactamos várias empresas de transportes através de *email*, de modo a podermos obter respostas às seguintes perguntas:

- Como é que a informação relativa à oferta de transportes é armazenada e curada?
- Como é que é apresentada ao utente dos serviços de transportes?
- Em relação à(s) equipa(s) encarregue(s) destas funções, qual seria o nível de conforto à introdução de um esquema de apresentação, armazenamento e/ou atualização novo? Por outras palavras, qual é o nível de literacia tecnológica que essas equipas apresentam?

Toda esta informação poderia influenciar a nossa solução, de modo a ter em conta vários tipos de fontes de dados, por exemplo, para além de *feeds* GTFS, como, por exemplo, folhas de cálculo.

No entanto, apesar de repetidas tentativas de contacto, nenhuma empresa com que contactamos mostrou interesse em fornecer respostas às perguntas que fizemos. Isto impossibilitou que esta parte do projeto trouxesse frutos significativos que influenciassem a solução. No entanto, na secção relativa ao trabalho futuro iremos elaborar sobre o que é a informação que as operadoras nos poderiam fornecer influenciaria a nossa solução.

4 Trabalho Futuro

Nesta secção, descreveremos que potenciais alterações a adições podem ser feitas no futuro ao *software* que produzimos.

Em relação ao *website*, seria interessante fornecer alguma forma de, no *back office*, desenhar as *shapes* de um percurso, utilizando uma interface mais simples, que não requeira apenas acessos à Base de Dados. Esta funcionalidade seria particularmente interessante em contextos como se poderiam encontrar noutras operadoras de transportes, em que não temos uma forma de exportar ficheiros GTFS, que contêm ditas *shapes*.

Em relação à aplicação móvel, não a conseguimos implementar, pelo que trabalho futuro passaria pela sua implementação. A mesma seria apenas uma adaptação do *front end* que construímos, utilizando *frameworks* de *Android* para adaptar o que já temos. Acreditamos que *Capacitor*, uma *framework* de *Android* para adaptar *Vue*. Nós experimentamos utilizá-la para demonstrar a aplicação, mas não conseguimos que funcionasse a tempo.

No que toca ao *GTFS_Val*, existem vários pontos no sistema em que o mesmo poderia ser melhorado.

Em primeiro lugar, de modo a tornar-se numa solução mais genérica, poderia ser implementado um método de construção de um *feed* GTFS, com base numa fonte de informação diferente. Esta funcionalidade não foi implementada pois, para o caso de uso dos TUB, o *feed* já é produzido com base na informação na Base de Dados dos mesmos. No entanto, dependendo das possíveis respostas das operadoras de transportes públicos que não obtivemos, poderíamos implementar funcionalidades de leitura de folhas de cálculo, bases de dados relacionais, etc. Apesar de sabermos que existem operadoras que utilizam folhas de cálculo em vez de bases de dados para guardar os dados da oferta, não sabendo a estrutura desses ficheiros, leitura dos mesmos não poderia ser implementada. Não sabendo também o *schema* das Bases de Dados que as mesmas usam, também não poderíamos gerar o *feed* com base numa BD.

Por outro lado, a interface é bastante simples. Uma versão mais *user friendly*, que não utilizasse apenas a linha de comandos e um ficheiro escrito à mão para passar as informações necessárias poderia ser interessante. Não implementamos isto pois não achamos necessário tendo em conta o nível de literacia tecnológica da equipa dos TUB.

5 Conclusão

Em suma, acreditamos ter atingido os principais objetivos deste projeto. Conseguimos desenvolver uma interface nova, com uma apresentação, nos nossos olhos, simples e objetiva da informação da oferta dos TUB. Mais ainda, conseguimos desenvolver a ferramenta pedida para validar o *export* que os mesmos fazem periodicamente, potencialmente facilitando o trabalho da equipa do *back office* e aumentando produtividade.

Todo este projeto teve como objetivo representar a informação que os TUB possuem para o cliente. Ao longo do ciclo de desenvolvimento, o maior entrave foi sempre a limitação dos dados, muitas vezes apresentando-se incompletos, o que também acabou por limitar a nossa solução. Contudo, dentro dos possíveis, acreditamos ter atingido uma excelente solução que satisfaz todos os requisitos que a empresa inicialmente propôs.

Em relação ao contacto que tivemos com a empresa, acreditamos que esta tenha sido bastante positiva, assim como a relação com os seus representantes, tendo eles fornecido toda a ajuda que nós necessitamos e estando sempre disponíveis. Gostaríamos de deixar aqui o nosso agradecimento pela mesma ajuda.

Concluindo, acreditamos que o trabalho foi muito bem sucedido, tendo ido de encontro às expectativas dos TUB, tendo em conta o período que tivemos.

6 Referências

- [1] *General Transit Feed Specification Reference*. URL: <https://gtfs.org/reference/static>.
- [2] *MacBook Air, Early 2015 model*. URL: https://support.apple.com/kb/SP714?locale=en_US.

7 Anexos

7.1 *Frontend*

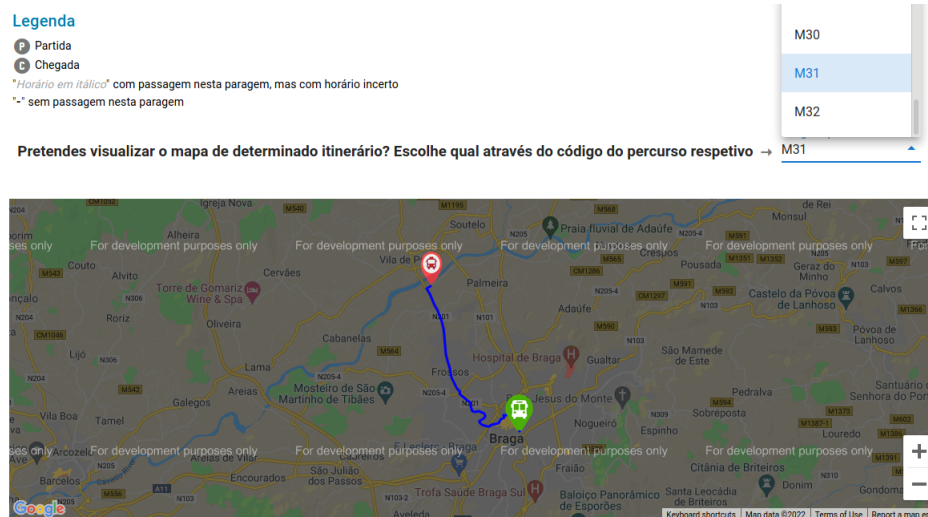


Figure 13: Mapa com paragem destacada (a verde) e a seleção do código de mapa

DIAS ÚTEIS				SÁBADOS										DOMINGOS E FERIADOS									
Ponte de Prado (Marginal)	<div><div></div><div>06:45</div></div>	<div><div></div><div>07:15</div></div>	<div><div></div><div>07:45</div></div>	<div><div></div><div>08:15</div></div>	<div><div></div><div>08:45</div></div>	<div><div></div><div>09:15</div></div>	<div><div></div><div>09:45</div></div>	<div><div></div><div>10:15</div></div>	<div><div></div><div>10:45</div></div>	<div><div></div><div>11:15</div></div>	<div><div></div><div>11:45</div></div>	<div><div></div><div>12:15</div></div>	<div><div></div><div>12:45</div></div>	<div><div></div><div>13:15</div></div>	<div><div></div><div>13:45</div></div>	<div><div></div><div>14:15</div></div>	<div><div></div><div>14:45</div></div>						
[MOSTRAR PARAGENS INTERMÉDIAS]																							
Liberdade (25 de Abril)	07:10	07:40	08:10	08:40	09:10	09:40	10:10	10:40	11:10	11:40	12:10	12:40	13:10	13:40	14:10	14:40							
[FECHAR PARAGENS INTERMÉDIAS]																							
João XXI (31 de Janeiro I)	-	-	-	-	-	-	-	-	11:11	-	-	-	-	-	-	-							
João XXI (C Amarante)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-							
João Paulo II (Piscinas)	-	-	08:13	-	-	-	-	-	-	-	-	-	-	-	-	-							
João Paulo II (Parque Rodovia)	-	-	-	-	-	-	-	-	-	-	-	12:44	-	-	-	-							
João Paulo II (Inl)	-	-	-	-	09:16	-	-	-	-	-	-	-	-	-	-	-							
Lusiadas I	-	-	-	08:48	-	-	-	-	-	-	-	-	-	-	-	-							
Lusiadas II	-	-	-	-	-	09:50	-	-	-	-	-	-	-	-	-	-							
Lusiadas III	-	-	-	-	-	-	-	-	-	-	-	-	-	-	13:52	-							
Lusiadas IV (Tenões)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-							
República I	-	-	-	-	-	-	-	-	-	-	12:26	-	-	-	-	-							
República II	-	-	-	-	-	-	-	-	11:28	-	-	-	-	-	-	-							
Bom Jesus	<div><div></div><div>07:30</div></div>	<div><div></div><div>08:00</div></div>	<div><div></div><div>08:30</div></div>	<div><div></div><div>09:00</div></div>	<div><div></div><div>09:30</div></div>	<div><div></div><div>10:00</div></div>	<div><div></div><div>10:30</div></div>	<div><div></div><div>11:00</div></div>	<div><div></div><div>11:30</div></div>	<div><div></div><div>12:00</div></div>	<div><div></div><div>12:30</div></div>	<div><div></div><div>13:00</div></div>	<div><div></div><div>13:30</div></div>	<div><div></div><div>14:00</div></div>	<div><div></div><div>14:30</div></div>	<div><div></div><div>15:00</div></div>							
Código Percurso	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16							

Figure 14: Tabela dos horários com o ”mostrar paragens intermédias” aberto

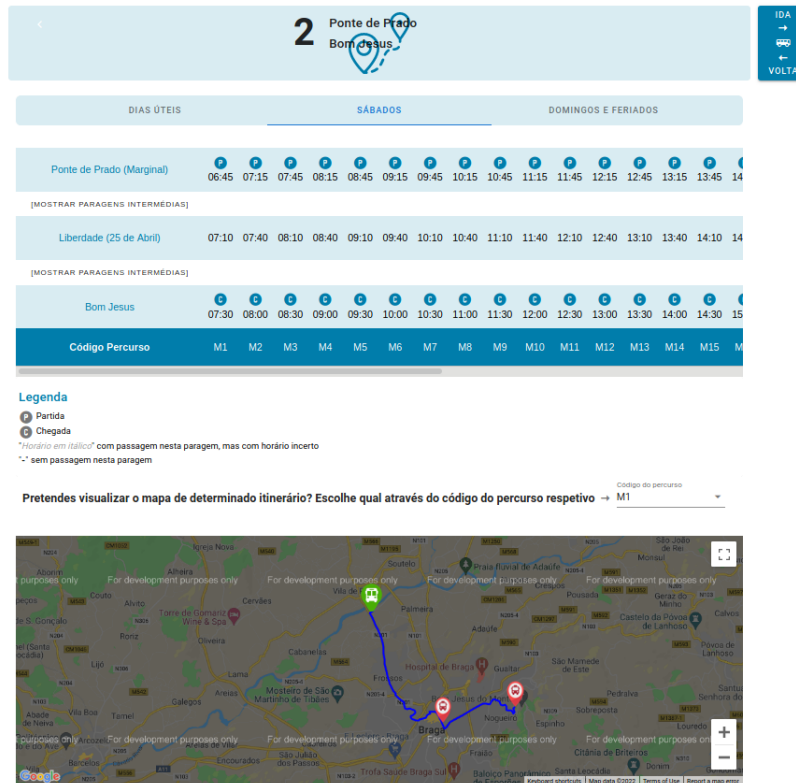


Figure 15: *Frontend* completo

7.2 *Schema* do ficheiro de configuração do *GTFS_Val*

```
{
  "file_source": String,
  "notification_receivers": [String],
  "notification_sender": String,
  "update_frequency": Number,
  "starting_date": String,
  "ftf_host": String,
  "ftp_user": String
}
```

Listing 1: *Schema* do Ficheiro de Configuração