

Antenor Moreira de Barros Leal

**Aplicativo web de auxílio à
navegação aérea**

PROJETO FINAL

DEPARTAMENTO DE INFORMÁTICA
Programa de Graduação em Engenharia da
Computação

Rio de Janeiro
Junho de 2024

Antenor Moreira de Barros Leal

Aplicativo web de auxílio à navegação aérea

Relatório de Projeto Final I

Relatório de Projeto Final, apresentado ao Programa de Engenharia da Computação, do Departamento de Informática da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Prof. Adriano Francisco Branco

Rio de Janeiro
Junho de 2024

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

Antenor Moreira de Barros Leal

Graduando em Engenharia da Computação na PUC - Rio

Ficha Catalográfica

Leal, Antenor Moreira de Barros

Aplicativo web de auxílio à navegação aérea / Antenor Moreira de Barros Leal; orientador: Adriano Francisco Branco. – 2024.

?? f: il. color. ; 30 cm

Projeto Final - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2024.

Inclui bibliografia

1. Informática – Trabalho de Conclusão de Curso (Graduação). 2. Aviação. 3. Navegação. 4. Aplicativo. 5. Algoritmo. 6. Web. 7. Internet. I. Branco, Adriano Francisco. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Resumo

Leal, Antenor Moreira de Barros; Branco, Adriano Francisco. **Aplicativo web de auxílio à navegação aérea**. Rio de Janeiro, 2024. ??p. Projeto Final – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

É um aplicativo web de código aberto com o objetivo de auxiliar usuários de simuladores de voo que não possuem acesso à ferramenta (Electronic Flight Bag) que um piloto de linha aérea teria. Ao acessar o aplicativo, o usuário se depara com a lista de aeroportos cadastrados e, após escolher um, são exibidas as informações da pista, frequências do aeroporto (torre, solo, ATIS, etc.), e frequências de navegação (ILS, VOR, etc.). Também são apresentadas as informações das condições meteorológicas atuais do aeródromo (vento, visibilidade, temperatura, etc.), tanto no formato oficial (.ETAR), obtidas a cada hora de uma API externa, como em um texto em linguagem natural para melhor entendimento do jogador iniciante. Um usuário com permissão de administrador pode adicionar e editar aeroportos. Módulos adicionais estão disponíveis como o cálculo da pista em uso (a partir de informações do vento) e do perfil de descida.

Palavras-chave

Aviação; Navegação; Aplicativo; Algoritmo; Web; Internet.

Abstract

Leal, Antenor Moreira de Barros; Branco, Adriano Francisco (Advisor).
Aerial navigation aid web application. Rio de Janeiro, 2024. ??p.
Projeto Final – Departamento de Informática, Pontifícia Universidade
Católica do Rio de Janeiro.

It is an open source web application to auxialiate the flight simulator's users that don't have access to the tool (Electronic Flight Bag) that an airline pilot would have. When accessing the app, the users encounter the list of registered airports and, after choosing one, runway information, airport frequencies (tower, ground, ATIS etc) and navigation frequencies (ILS, VOR etc) are showed. Also provided are the current meteorological conditions of the aerodrome (wind, visibility, temperature, etc.), both in the official format (METAR), obtained hourly from an external API, and in natural language text for better understanding by novice players. An administrator user can add and edit airports. Based on the current wind information, the active runway is calculated. Additional modules are available such as calculation of the runway in use (from wind information) and the descent profile.

Keywords

Aviation; Navigation; Application; Algoritm; Web; Internet.

Sumário

Lista de figuras

Lista de tabelas

Lista de Abreviaturas

METAR – *ME*Teorological Aerodrome Report (Informe Meteorológico de Aeródromo)

EFB – *Electronic Flight Bag* (Maleta Eletrônica de Voo)

ICAO – *International Civil Aviation Organization* (Organização Internacional da Aviação Civil)

IATA – *International Air Transport Association* (Associação Internacional de Transporte Aéreo)

ATIS – *Automatic Terminal Information Service* (Serviço Automático de Informação Terminal)

PK – *Public Key* (Chave Pública em banco de dados)

FK – *Foreign key* (Chave Estrangeira em banco de dados)

SSH – *Secure SHell* (Shell Segura)

VPS – *Virtual Private Server* (Sevidor Virtual Privado)

ISP – *Internet Service Provider* (Provedor de Serviços de Internet)

API – *Application Programming Interface* (Interface de Programação de Aplicações) REST – *Representational State Transfer* (Transferência de Estado Representacional)

1

Introdução

Com o aumento da capacidade de passageiros e carga e a necessidade de uma maior segurança, começou a se fazer necessário trazer ao cockpit vários documentos como checklist de procedimentos; log book; cartas de navegação, de saída, de aproximação, do aeródromo; tabelas de performance da aeronave etc.

Para levar tudo isto costumava-se usar uma maleta (a Flight Bag), obviamente esta ficava muito pesada.

Com a miniaturização dos computadores e surgimentos dos tablets, começaram a ser desenvolvidos programas que substituíam partes ou todos estes documentos, é a chamada maleta de voo eletrônica, mais conhecida pela sigla em Inglês EFB (*Electronic Flight Bag*).

Atualmente existem hardware dedicados para esta função, mas é mais comum se usar um tablet com um aplicativo disponibilizado pela companhia aérea. Normalmente, o tablet escolhido é um iPad da Apple, mas algumas companhias optaram pelo Microsoft Surface. [??]

O uso do EFB trouxe uma série de benefícios para os pilotos e para as companhias aéreas. Além de reduzir o peso e o volume de documentos físicos a serem transportados, o EFB permite uma rápida atualização das informações, garantindo que os pilotos tenham sempre acesso às versões mais recentes das cartas de navegação. [??]

Além disso, a capacidade de armazenamento do EFB possibilita o acesso a uma vasta quantidade de informações adicionais, como manuais de operação da aeronave, regulamentações atualizadas e até mesmo dados meteorológicos em tempo real, o que contribui para uma tomada de decisão mais informada e segura durante o voo.

2

Sistemas Similares

Os EFBs possuem funções variadas como cálculo de combustível, de performance, etc. Para aeronaves mais novas, como o Airbus A320 é difícil realizar cálculos de performance, porque não é disponibilizado ao público como este cálculo é feito. Ferramentas encontradas na Internet [??] normalmente fazem engenharia reversa, e portanto, podem apresentar resultados diferentes de um cálculo oficial.

Nos simuladores de voo para computador pessoal, algumas aeronaves simulam este equipamento como o Airbus A320neo desenvolvido pela *FlyByWire Simulations*. Apesar de ser uma aeronave *freeware*, ela é bem sofisticada chegando ao nível de realismo da *Fenix Simulations* ou da *ToLiss Simulations*, duas produtoras com modelos pagos do A320.

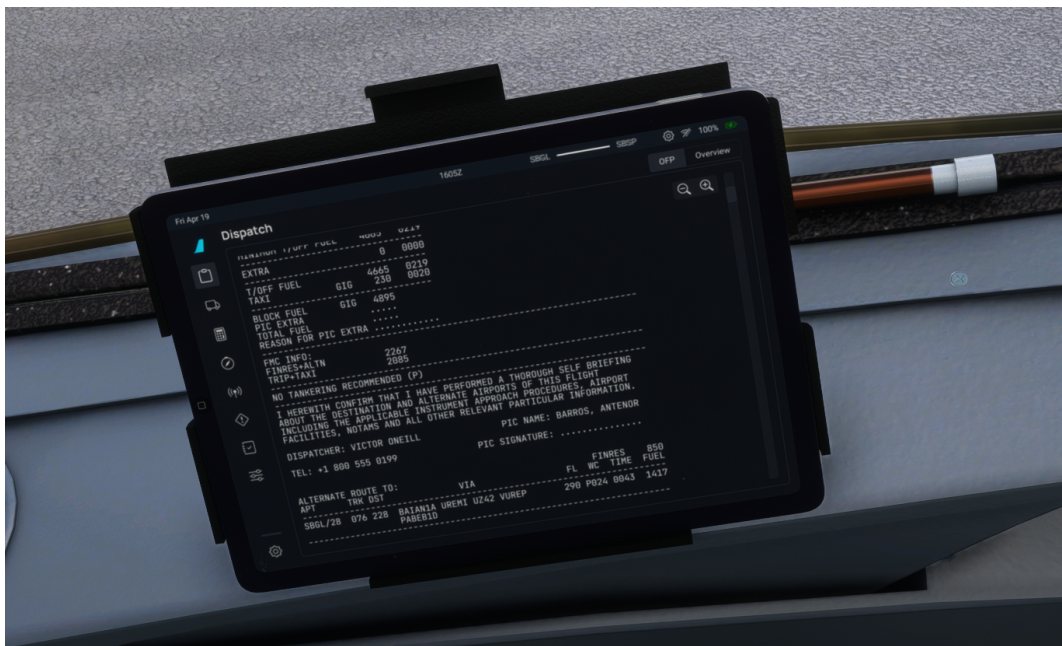


Figura 2.1: Exemplo de um EFB no Flight Simulator 2020 na aeronave A320neo

Contudo, o METAR do aeródromo não se encontra disponível no EFB. É possível usar o computador de bordo da aeronave (FMC) e conseguir esta informação. Também é possível sintonizar na frequência do ATIS, mas isto só funcionará se o avião já estiver perto do aeródromo.

O que muitos jogadores fazem é acessar o AISWEB (<https://aisweb.decea.mil.br/>), sistema oficial brasileiro de informações aeronáuticas.

É um site extremamente completo, podendo ser usado em operações reais, mas para o jogador iniciante seria de valia uma interface mais simples.



The screenshot shows the AISWEB website interface. At the top, there is a navigation bar with links for 'Abreviaturas', 'Central de Ajuda', and a language selector set to 'Portuguê'. Below this is a header with the AISWEB logo and a menu with 'INÍCIO', 'AERÓDROMOS' (selected), 'ESPAÇO AÉREO', and 'NOTAM'. The main content area is titled 'Santos Dumont (SBRJ)' with the location 'Rio de Janeiro/RJ' and the CIAD 'RJ0002'. A horizontal menu below the title includes 'ROTAER', 'NOTAM', 'Suplementos AIP', 'Cartas', 'Metar/TAF', and 'Rotas Preferenciais'. The 'ROTAER' section is active, displaying airport details for Santos Dumont (SBRJ) in Rio de Janeiro, RJ, including coordinates (22 54 36S/043 09 45W), elevation (3 (10)), and runway information (02L, 02R, 20L, 20R). It also lists communication frequencies for COM (Tower, Solo, Traffic, ATIS) and other services (CMB, SER, RFFS).

ROTAER D-AMDT 9/24 [Ver mudanças](#)

Santos Dumont (SBRJ) / RIO DE JANEIRO, RJ 22 54 36S/043 09 45W
 AD PUB/MIL INFRAERO 2E UTC-3 VFR IFR L21 , L23 , L26 3 (10)

02L - L9(3.23) [1] [12] [14] , L12 - (1260x30 ASPH 39/F/B/X/T L14 , L15) - L9(3.12) SBCW (CRCEA-SE)
 [11] [14] , L12 - **20R**

02R - L9(3.23) [1] [2] [14] , L12 - (1323x42 ASPH 65/F/B/X/T L14 , L15) - L9(3.12)
 [4] [14] , L12 - **20L**


COM -TORRE RIO [9] [14] 118.700 [10] 121.500 [3]
 SOLO RIO [10] [14] 121.700
 TRÁFEGO RIO [10] [14] 121.050
 ATIS RIO [8] [10] [14] 132.650

CMB- TF [5] **SER** - S1 **RFFS** - CAT CIVIL - 7

Figura 2.2: AISWEB com informações de pista, frequências de comunicação e navegação para o Santos Dumont

O AISWEB exibe o METAR no aeroporto, mas não explica para o que cada campo serve.

O site METAR-TAF (<https://metar- taf.com/>) é o decoder mais conhecido, possui uma interface gráfica bem construída e muito fácil de entender, mas não possui a lista de frequência dos aeroportos e de radionavegação.



METAR
192000Z 17006KT 9999 FEW030 BKN050 23/17
Q1018=

TAF
191500Z 1918/2006 23005KT 9999 FEW020
TX25/1918Z TN22/2006Z BECMG 2000/2002
27005KT FEW030 BECMG 2004/2006 32005KT
SCT017 SCT025 RMK PGY=

Figura 2.3: METAR do Santos Dumont no AISWEB

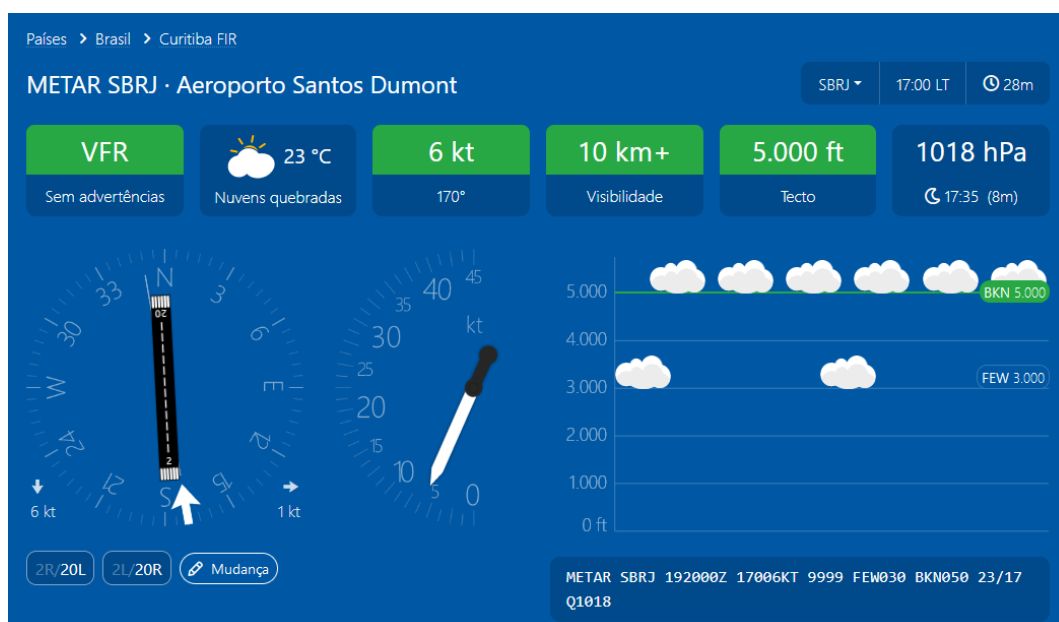


Figura 2.4: Interface gráfica do METAR-TAF

3

A Proposta

A ideia do trabalho seria unir as funcionalidades do METAR-TAF com o AISWEB em uma interface gráfica que o usuário iniciante consiga usar sem dificuldades.

Pelo fato de aviação necessitar ter um ambiente seguro e bastante regulado, considerando que meu projeto é apenas um protótipo, prefiro restringir o caso de uso apenas para jogadores de simuladores de voo que desejam que a simulação seja parecida com o real. Nas páginas do sistema conterà um aviso de que o sistema **não deve ser usado para um voo real**.

Dito isto, o sistema possui backend escrito na linguagem Python, fazendo uso da framework FastAPI. Até a entrega do Projeto I, o backend era feito com Flask. A escolha foi feita pela minha familiaridade com a framework. Porém, depois, descobri a framework FastAPI no meu trabalho. Normalmente, ela não é usada para fullstack, mas sim para fazer APIs REST, que retornam JSON na resposta de uma requisição. Um frontend feito em alguma framework de JavaScript faria a requisição e atualizaria a página.

Contudo, é possível retornar qualquer tipo de dado no FastAPI, inclusive páginas HTML. Como prefiro fazer a renderização de páginas server-side, continuei usando a funcionalidade de templates da biblioteca Jinja2.

A substituição da framework deu-se pelos seguintes motivos:

- Projeto mais maduro, continuamente mantido.
- Documentação bem mais detalhada do que a do Flask, com vários tutoriais sobre assuntos comumente usados, como autenticação e sessão de usuário.
- Servidor embutido (Uvicorn) extremamente fácil de configurar e suficiente para produção [??].
- Assíncrono, permitindo o uso junto com bibliotecas que utilizam asyncio.

No segundo semestre de 2023 comecei a fazer um projeto para uso próprio. O código está disponível em <https://github.com/antenor-z/aero>. Atualmente o projeto funciona, mas a arquitetura foi feita sem muito planejamento, as informações do aeroporto são hardcoded.

O usuário tem acesso a informações de frequência da torre, solo, tráfego, rampa e operações, bem como das frequências e dados para VOR (um sistema de radionavegação por antenas no solo), ILS (sistema de pouso por instrumentos) e informações de pista. Neste trabalho quero, armazenar estas em um

banco de dados relacional com uma arquitetura bem planejada. Farei testes de desempenho simulando uma alta taxa de acesso e, dependendo dos resultados, fazer uso de um banco em memória como intermediário.

Os aeródromos podem, ao longo do tempo, mudarem alguma frequência e outras informações, como o número da pista, que muda a depender da variação do norte magnético, uma ampliação da pista, etc. Atualmente, o código precisa ser alterado para atualizar estas informações. Desejo implementar um sistema diretamente no site, com uma autenticação por senha e TOTP, para que seja possível mudar qualquer informação no banco.

Através de uma API do serviço americano National Weather Service, são coletadas as informações atuais de meteorologia. Estas informações (que vêm em um formato chamado METAR) são processadas pelo backend e mostradas ao usuário de uma forma fácil de entender. Esta parte em específico possui um código de difícil manutenção. Desejo refatorar esta parte e adicionar suporte para a maior parte de códigos da especificação do METAR.