

DESENVOLVIMENTO DE APLICAÇÃO WEB PARA UNIFICAR HORÁRIOS E SALAS DA FATEC RIBEIRÃO PRETO

Ítalo Garcia Covas¹, Arthur de Oliveira Galanti¹, Lucas Baggio Figueira¹, Anna P. Z. China¹

¹Faculdade de Tecnologia de Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

italo.covas@fatec.sp.gov.br, arthur.galanti@fatec.sp.gov.br,
lucas.figueira@fatec.sp.gov.br, anna.china@fatec.sp.gov.br

Resumo. *Este artigo descreve o desenvolvimento de uma aplicação destinada a unificar as informações de horários e salas de aula da FATEC Ribeirão Preto em um único lugar. O objetivo é facilitar o acesso dos alunos às informações sobre aulas, proporcionando uma visão personalizada da localização das salas e das grades de horários, além de incluir algumas informações gerais do aluno.*

Abstract. *This article describes the development of an application designed to unify information about schedules and classrooms at FATEC Ribeirão Preto in a single place. The objective is to facilitate students' access to information about their classes, providing a personalized view of classroom locations and schedules, as well as including some general student information.*

1. Introdução

A gestão eficiente das informações referentes à grade horária das disciplinas e à localização das salas de aula é essencial para o bom andamento das atividades educacionais em instituições de ensino superior. No entanto, muitas dessas instituições enfrentam limitações em seus sistemas de informação, o que dificulta o acesso rápido e fácil dos alunos às informações necessárias sobre as salas de aula. Essa deficiência pode resultar em atrasos, perda de tempo e até mesmo impactar negativamente o desempenho acadêmico dos estudantes.

A falta de integração adequada entre os dados de horários das disciplinas e as informações sobre as salas de aula é uma problemática recorrente. Métodos tradicionais, como listas impressas ou avisos em murais, mostraram-se ineficazes na era digital, onde a necessidade de acesso rápido e preciso a essas informações é crescente. Embora a tecnologia tenha o potencial de simplificar esses processos, ainda existe uma lacuna significativa entre a disponibilidade dos dados e o acesso eficaz pelos alunos.

Este artigo tem como objetivo abordar essa problemática propondo a criação de um sistema que unifique as informações de horários e salas de aula de forma personalizada para cada aluno. O sistema proposto incluirá um mapa interativo do campus, permitindo que os alunos visualizem a localização exata de cada sala de aula e a rota mais eficiente para chegar até ela. Além disso, a plataforma oferecerá acesso a grades horárias personalizadas, com atualizações e mudanças sendo realizadas pela coordenação dos professores.

Espera-se que a implementação deste sistema reduza significativamente o tempo

que os alunos gastam procurando suas salas de aula e minimize o impacto de mudanças na grade horária. A adoção de tecnologias avançadas para resolver problemas cotidianos no ambiente educacional reforça o compromisso das instituições de ensino com a inovação e a melhoria contínua dos processos acadêmicos, contribuindo para um ambiente educacional mais organizado e produtivo.

A estrutura deste estudo inicia com a exploração da gestão eficiente de informações acadêmicas na Seção 2, seguida por uma análise detalhada da importância do Web Scraping na coleta de dados na Seção 2.1. Questões relacionadas à segurança no login e a integração de sistemas são abordadas nas Seções 2.2 e 2.3, respectivamente, enquanto aspectos de usabilidade e desempenho são discutidos na Seção 2.4. Os métodos e materiais utilizados são detalhados na Seção 3, e os resultados são apresentados de forma concisa na Seção 4. Por fim, os resultados alcançados são discutidos, e reflexões sobre os desafios enfrentados e os próximos passos são expostas na Seção 5.

2. Referencial Teórico

Para a elaboração do seguinte projeto, foi necessário embasamento teórico multidisciplinar. Os conhecimentos usados são derivados do *Design* e da Gestão de Informações. Entretanto, este trabalho também possui um enfoque em *Web Scraping* e Desenvolvimento Web, apoiando-se no que há de mais atualizado nessas vertentes.

2.1. Gestão de Informações Acadêmicas

A gestão de informações acadêmicas desempenha um papel fundamental no ambiente educacional (BATES & BATES, 2019). Sistemas de informação acadêmica são essenciais para gerenciar e disponibilizar dados relacionados a horários de aulas, disciplinas, professores, alunos e salas de aula. No entanto, a fragmentação e a falta de integração dessas informações podem resultar em dificuldades significativas para os alunos, especialmente na localização das salas de aula (GARRISON & VAUGHAN, 2018). Esse problema recorrente nas instituições de ensino pode levar a atrasos, confusão e estresse para os estudantes.

Embora muitos sistemas forneçam informações sobre horários e disciplinas, eles frequentemente falham em indicar a localização exata das salas de aula. Isso representa um desafio, pois a ausência de informações claras pode impactar negativamente a experiência acadêmica dos alunos.

A personalização da experiência do aluno é uma tendência crescente na educação (AMBROSE, 2010). Sistemas de informação acadêmica têm a capacidade de fornecer dados personalizados, adaptados às necessidades individuais de cada aluno. Ao integrar informações de horários de aula com a localização das salas de aula de forma personalizada, os alunos podem economizar tempo e minimizar contratempos em sua rotina acadêmica.

A integração de sistemas é uma abordagem estratégica que visa unificar dados de diversas fontes para criar uma visão abrangente e coesa. Neste projeto, a integração de sistemas é essencial para combinar informações de horários de aula com as localizações das salas de aula, assegurando que os alunos tenham acesso fácil e direto a todas as informações necessárias.

Além disso, a implementação de soluções tecnológicas que permitam a integração e a personalização das informações pode melhorar significativamente a eficiência

administrativa e a satisfação dos alunos. Ferramentas como aplicativos móveis e plataformas online podem ser utilizadas para fornecer informações em tempo real sobre horários e localizações, facilitando a navegação no ambiente acadêmico.

Em resumo, a gestão eficaz de informações acadêmicas, através da integração de sistemas e da personalização de dados, é crucial para melhorar a experiência dos alunos e a eficiência das instituições de ensino. A adoção de tecnologias avançadas e estratégias de integração pode transformar a forma como as informações são gerenciadas e disponibilizadas, beneficiando tanto os alunos quanto os administradores educacionais.

2.1. Web Scraping

Web Scraping é uma técnica automatizada de coleta de dados de sites da web, utilizando scripts para acessar páginas, baixar conteúdo e extrair informações específicas, como texto e imagens. Essa técnica é amplamente utilizada para obter dados de diversas fontes online, como sites de notícias, redes sociais e lojas virtuais, e tem aplicações em análise de informações, monitoramento de preços, pesquisa de mercado e rastreamento de tendências (MITCHELL, 2020).

No contexto acadêmico, o *Web Scraping* é essencial para a coleta de dados para pesquisas, permitindo a análise de padrões e tendências com maior precisão. A implementação envolve identificar fontes relevantes, desenvolver scripts de extração e processar os dados coletados. É fundamental que o *Web Scraping* seja realizado de acordo com leis e regulamentações, respeitando a privacidade e a ética na coleta de dados.

Em resumo, o *Web Scraping* é uma ferramenta poderosa para a coleta e análise de dados na internet, com inúmeras aplicações acadêmicas e de mercado, transformando a forma como os dados são obtidos e analisados.

2.2. Segurança no Login

A segurança no login é crucial para proteger informações pessoais e sistemas online, especialmente em aplicações que utilizam login e senha de outras plataformas para Web Scraping. Para garantir a segurança nesse contexto, várias medidas são essenciais.

O uso de senhas fortes e únicas é fundamental, assim como monitorar atividades suspeitas e criptografar informações de login. Manter o software atualizado ajuda a proteger contra vulnerabilidades conhecidas. A educação dos usuários sobre ataques de phishing é essencial para evitar comprometimentos acidentais das credenciais. A autenticação biométrica, quando disponível, oferece uma segurança adicional.

O monitoramento constante e a realização de auditorias regulares ajudam a garantir o cumprimento das políticas de segurança. Políticas rigorosas de senhas, como mudanças periódicas e a proibição de senhas reutilizadas, também são importantes.

É importante destacar que, para aumentar a segurança e a privacidade dos usuários, as senhas utilizadas para login em outras plataformas não serão armazenadas pela aplicação de web scraping. Em vez disso, a aplicação utilizará métodos seguros de acesso direto sem a necessidade de armazenamento de credenciais.

Em suma, a segurança no login para aplicações que utilizam credenciais de outras plataformas para web scraping deve ser priorizada. Essas práticas ajudam a proteger a aplicação contra uma ampla gama de ameaças cibernéticas, garantindo a integridade e a confidencialidade dos dados (CLOUDFLARE, 2023).

2.3. Testes de usabilidade e desempenho

Testes de usabilidade são essenciais para avaliar a facilidade de uso de um produto, identificando problemas de interface e fluxo de usuário. Eles envolvem a coleta de feedback de usuários reais para aprimorar a experiência. No contexto de uma aplicação que utiliza login e senha de outra plataforma para web scraping, esses testes garantem que a interação do usuário seja intuitiva e eficiente.

Testes de desempenho, por outro lado, medem a capacidade de resposta, escalabilidade e estabilidade do sistema sob carga. Eles asseguram que o produto funcione bem mesmo em situações de alto tráfego. Para uma aplicação de *Web Scraping*, é vital garantir que o sistema possa lidar com grandes volumes de dados e múltiplas solicitações sem perder desempenho.

Uma das dificuldades em realizar testes complexos em uma aplicação que depende de uma plataforma de terceiros é a falta de controle sobre os dados. As informações podem ser modificadas ou seguir padrões desconhecidos, dificultando a criação de testes precisos. Além disso, mudanças na plataforma externa podem introduzir novos problemas ou alterar o comportamento esperado, exigindo monitoramento contínuo e ajustes frequentes nos testes.

Em suma, tanto os testes de usabilidade quanto os de desempenho são cruciais para garantir que a aplicação atenda às expectativas dos usuários e funcione de maneira confiável, apesar dos desafios adicionais apresentados pela dependência de uma plataforma de terceiros (LAMBDA TEST, 2024).

3. Materiais e Métodos

Com base no referencial teórico levantado, podemos esmiuçar a parte técnica do trabalho. A partir daqui, serão descritos materiais e métodos utilizados para alcançar o objetivo final.

O sistema que foi desenvolvido coletou informações fornecidas pela instituição de ensino de duas fontes principais: o sistema de informações de horários de aula da faculdade (SIGA) e o sistema de informações de localização das salas de aula. Os dados foram coletados em formato digital, incluindo um banco de dados *NoSQL*, *APIs* (*Application Programming Interfaces*) e *Web Scraping*.

Para o desenvolvimento do sistema, foi utilizada a linguagem de programação *C#* (Hejlsberg, 2000) com a biblioteca *AngleSharp* para interpretar os dados *HTML* da plataforma externa. *AngleSharp* é uma biblioteca poderosa que permite a extração de dados estruturados a partir de documentos *HTML*, facilitando a obtenção de informações precisas e relevantes necessárias para o sistema. A interface do usuário foi criada utilizando o framework web *React* (FACEBOOK, 2013). Para o armazenamento de dados, foi utilizado o banco de dados *Cloud Firestore*.

Uma camada de integração de dados foi implementada, que permite a combinação das informações de horários de aula e localização de salas de aula, garantindo que os dados sejam atualizados regularmente.

O sistema solicita que os alunos façam login com suas credenciais da faculdade para acessar as informações personalizadas para cada aluno, garantindo a segurança e a privacidade dos dados. Importante destacar que as senhas dos usuários não são armazenadas pelo sistema, assegurando maior segurança e privacidade.

3.1 TypeScript

TypeScript é uma linguagem de programação criada pela *Microsoft* em 2012, liderada por Anders Hejlsberg. Ela é um superset do *JavaScript* que adiciona tipagem estática opcional ao código. Inicialmente projetada para corrigir algumas das limitações do *JavaScript*, *TypeScript* permite a criação de aplicações mais robustas e de fácil manutenção, especialmente em projetos grandes. Com a chegada do *Node.js*, que permite a execução de *JavaScript* no servidor, *TypeScript* se beneficiou dessa flexibilidade, proporcionando um ambiente de desenvolvimento mais seguro e escalável (HEJLSBERG, 2012).

TypeScript é uma linguagem de alto nível, tipada e compilada. Sua sintaxe é similar à do *JavaScript*, mas com adições que permitem o uso de tipos estáticos, interfaces e outras características avançadas. Isso melhora a detecção de erros durante o desenvolvimento e facilita o trabalho em equipe. A sintaxe geral deriva da linguagem *Java*, com influências de *Scheme* e *Self*, e inclui recursos avançados para a orientação a objetos e programação funcional (HEJLSBERG, 2012).

3.2 Biblioteca React

React é uma biblioteca *open source* desenvolvida por engenheiros da *Meta* (antigo *Facebook*) em 2013, com Jordan Walke sendo um dos principais criadores. O principal objetivo do *React* é facilitar a criação de interfaces de usuário altamente interativas e responsivas. A biblioteca permite a mistura de *HTML* com *JavaScript* (ou *TypeScript*) para simplificar o desenvolvimento de aplicações. Entre os benefícios do *React*, destacam-se sua flexibilidade na criação de componentes reutilizáveis, alta performance graças ao *Virtual DOM*, e a capacidade de ser utilizado tanto para aplicações web quanto móveis, permitindo um grande reaproveitamento de código (FACEBOOK, 2013).

React utiliza uma abordagem declarativa para a construção de interfaces de usuário, o que facilita a atualização eficiente de componentes. A biblioteca suporta atualização *Hot Swap*, permitindo que as alterações no código sejam refletidas instantaneamente na interface do usuário sem a necessidade de recarregar a página, aumentando a produtividade do desenvolvedor.

O processo de desenvolvimento com *React* envolve várias etapas. Primeiro, ocorre a transpilação, onde o *Babel* é usado para converter a última versão do *JavaScript* ou *TypeScript* em uma versão mais antiga compatível com a maioria dos navegadores. Em seguida, o código transpilado é empacotado em um arquivo que será servido pelo servidor de desenvolvimento. Finalmente, o código é executado no navegador, permitindo a interação do usuário final com a aplicação. Ferramentas como *Webpack* ou *React CLI* são frequentemente utilizadas para otimizar a compilação e o empacotamento do código (PINHO & ESCUDELARIO, 2023).

3.3 Engenharia de Requisitos

A engenharia de requisitos é uma disciplina fundamental no desenvolvimento de software, focada na definição, documentação e manutenção dos requisitos do sistema. É através dessa engenharia que se garante que o software desenvolvido atenderá às

necessidades dos usuários e das partes interessadas. Esse processo inclui várias etapas essenciais, conforme discutido por Wieggers (2022).

Primeiramente, o levantamento de requisitos envolve a realização de entrevistas com usuários, clientes e stakeholders para garantir que o software atenderá plenamente às suas necessidades. O *stakeholder* é a pessoa responsável pelo produto do lado do cliente e pode incluir gerentes de área, gerentes de TI, ou desenvolvedores especializados. Esse levantamento também pode envolver visitas ao local onde o software será utilizado, permitindo uma compreensão mais profunda das necessidades e desafios do ambiente de trabalho.

Em seguida, a análise de requisitos filtra e valida as informações coletadas, descartando aquelas que são irrelevantes ou inviáveis. Os requisitos relevantes são então especificados detalhadamente em uma documentação clara e precisa.

A etapa final envolve a verificação e validação dos requisitos, onde são realizados testes para garantir que o software atende às especificações definidas. Esta etapa é crucial para assegurar a qualidade e a funcionalidade do produto final.

No desenvolvimento deste trabalho, a diagramação do software foi realizada utilizando a UML (*Unified Modeling Language*), uma linguagem de propósito geral que permite a modelagem de sistemas de qualquer nível de complexidade. Foi feito um diagrama de caso de uso para ilustrar as funcionalidades do sistema do ponto de vista do usuário (BOOCH, 2023). Esses diagramas foram criados utilizando a ferramenta *Astah UML*, que é gratuita para estudantes e facilita a elaboração de modelagens de software de forma eficiente. Os diagramas podem ser encontrados no repositório do projeto no *GitHub*, disponível em <https://github.com/ItaloCovas/horarios-fatec-rp>.

3.4 Prototipação

Durante o desenvolvimento do sistema, foi realizada uma prototipação inicial de algumas telas chave para guiar o design e a funcionalidade da interface do usuário. A ferramenta *Figma* foi utilizada para criar esses protótipos, permitindo uma visualização clara e interativa do layout e das interações esperadas. O protótipo desenvolvido pode ser acessado através do link <https://www.figma.com/design/kbw7et6ETBRoLzWqvCT4hp/Hor%C3%A1rios-FATEC-RP---Prot%C3%B3tipo-v1?node-id=0-1&t=RuEEE3yJvHozd0Cb-1>

Após a criação do protótipo inicial, o desenvolvimento do restante do sistema foi conduzido sem protótipos adicionais. Isso foi possível devido à clareza dos requisitos levantados e ao uso de ferramentas eficientes de desenvolvimento, permitindo ajustes rápidos e iterativos diretamente no código.

Ademais, a utilização do *Figma* como ferramenta de prototipação se mostrou extremamente vantajosa, uma vez que permitiu a colaboração em tempo real entre os membros da equipe, facilitando a comunicação e o alinhamento de expectativas. A visualização interativa das telas também ajudou na identificação precoce de possíveis problemas de usabilidade, permitindo que fossem corrigidos antes mesmo de serem

implementados no código. Dessa forma, a prototipação inicial foi fundamental para o sucesso do projeto, servindo como um guia claro e eficiente para o desenvolvimento do sistema.

4. Resultados

Nesta seção serão mostrados o diagrama de caso de uso, assim como as telas desenvolvidas a partir do protótipo inicial e uma explicação sobre o fluxo da aplicação “Horários FATEC RP”.

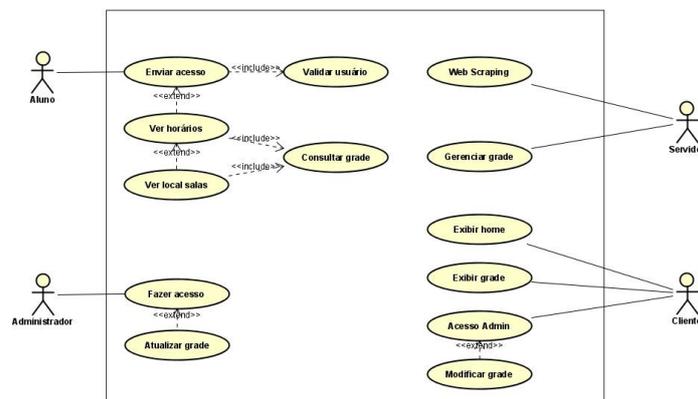


Figura 1: Diagrama de caso de uso da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

Após a análise de requisitos e o desenvolvimento do protótipo inicial como base, foi desenvolvido o restante das telas da aplicação.

A tela inicial (*landing page*) é representada pelas figuras 2 e 3, onde possui um cabeçalho com a logo e alguns links úteis na parte direita. Além disso, temos as seções de introdução, motivos para usar, e um incentivo ao *open source*, finalizando com um rodapé similar ao cabeçalho.



Figura 2: Landing Page da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).



Figura 3: Landing Page da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

Para entrar na aplicação como um aluno, deve-se clicar no botão “ENTRAR”. E um *dialog* será exibido para inserção de suas credenciais (RA e senha). Tal fluxo é representado na figura 4.

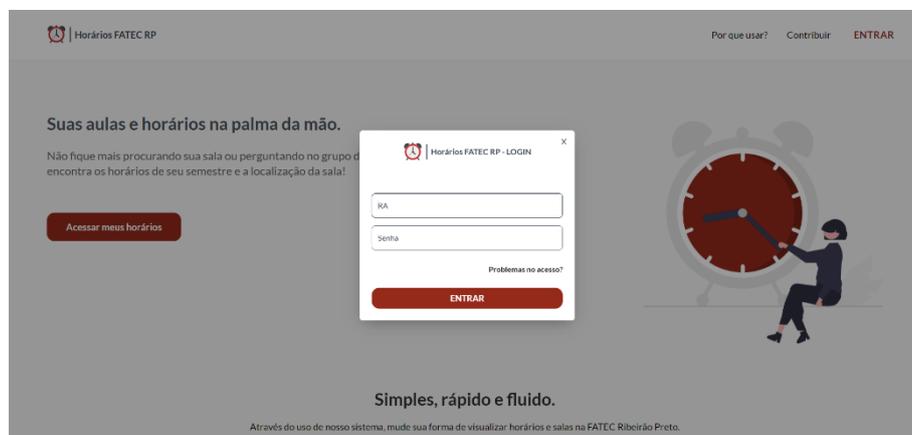


Figura 4: Modal de Login da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

Após efetuar com sucesso a entrada na aplicação, a tela exibida será a tela principal, que consiste em informar o dia atual e o nome do aluno, além de possibilitar selecionar o dia da semana para visualizar os horários, matérias, professores e localização das respectivas aulas (figura 5). Ao clicar no ícone de mapa, será exibido um *dialog* para visualização do mapa da FATEC, mostrando a sala de aula (figura 6).

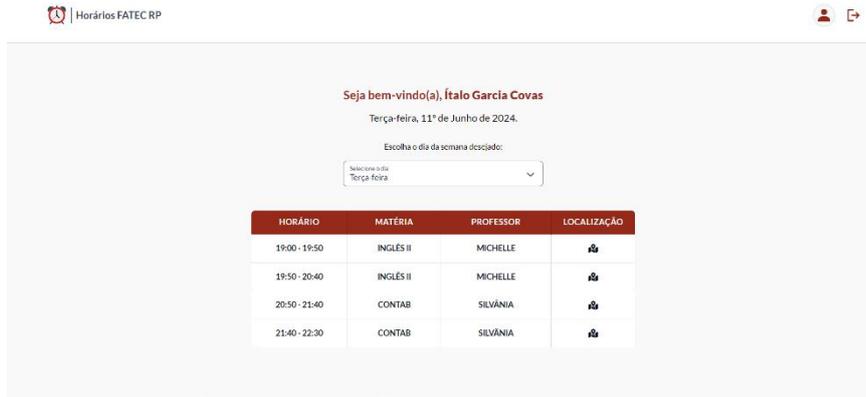


Figura 5: Tela principal da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

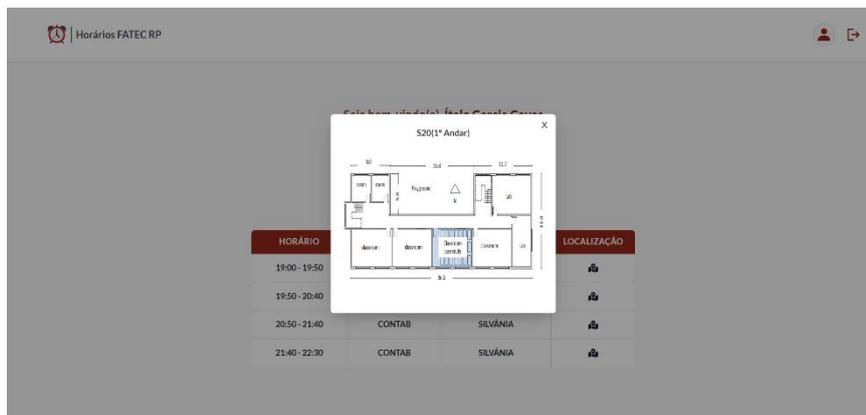


Figura 6: Exibição do mapa da FATEC da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

O cabeçalho da página é composto pela logo e por um ícone de usuário, onde se clicado será exibido as informações do aluno como nome, RA e o semestre atual em que ele está matriculado. Além disso temos o ícone de logout para que se clicado, faça com que volte a *landing page*. O fluxo é representado na figura 7.



Figura 7: Cabeçalho da tela principal da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

Com a função de popular o banco de dados e possibilitar com que a secretaria faça o *upload* da carga de horários do semestre, foi desenvolvida uma tela de administrador.

A figura 8 representa a tela de *login* do administrador, onde os responsáveis irão inserir suas credenciais para entrar no sistema. Já a figura 9 representa a tela de *upload* da carga de horários do semestre, onde após a secretaria subir um arquivo .xls ou .xlsx, que são as extensões do *Excel*, é possível clicar no botão “CARREGAR” para efetuar a carga.

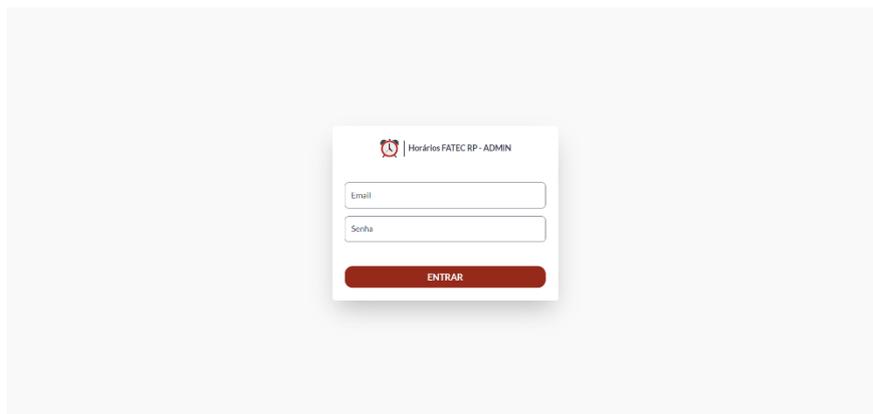


Figura 8: Tela de Login do administrador da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

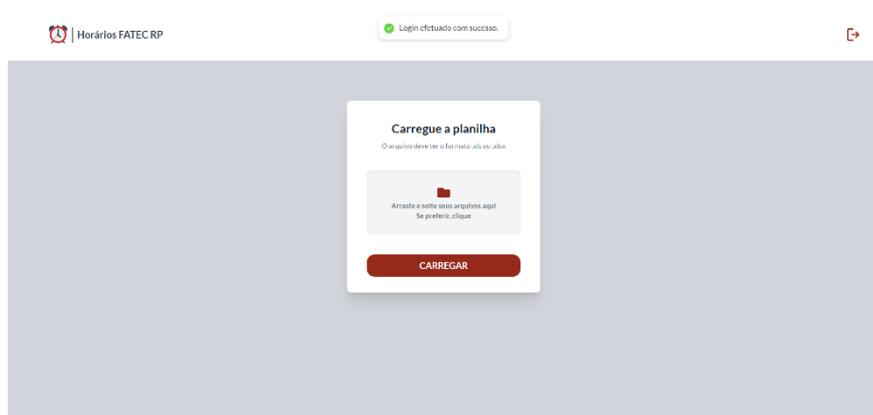


Figura 9: Tela de Upload do administrador da Aplicação Horários FATEC RP.

Fonte: (Autoria própria, 2024).

5. Considerações finais

Para alcançar os objetivos do projeto, necessitou-se de estudo aprofundado e da aplicação de técnicas avançadas de web scraping. O desenvolvimento da aplicação focou na unificação das informações de horários e salas de aula da FATEC Ribeirão Preto, facilitando o acesso dos alunos a essas informações. Espera-se que essa aplicação proporcione uma experiência mais intuitiva e visualmente atraente para os usuários, possibilitando uma visão personalizada das salas e grades horárias, além de incluir informações gerais do aluno.

Durante o desenvolvimento, foram enfrentadas algumas dificuldades significativas. A principal delas foi a implementação do web scraping, especialmente na extração e coleta precisa de dados do SIGA. Além disso, o tempo necessário para desenvolver a aplicação foi superior ao inicialmente planejado, devido à complexidade técnica e à necessidade de garantir a integração e segurança dos dados.

Para os encaminhamentos futuros, espera-se proporcionar uma experiência ainda mais intuitiva e visualmente atraente para os usuários. Planeja-se melhorar o design dos mapas apresentados, tornando-os mais detalhados e interativos, facilitando ainda mais a navegação e localização das salas de aula pelos alunos. Essas melhorias visam aumentar a usabilidade e a eficiência do sistema, beneficiando tanto alunos quanto administradores da instituição.

É importante valorizar que o objetivo foi atingido e todas as etapas percorridas com sucesso. Este desenvolvimento marca um passo significativo para a melhoria da experiência geral dos alunos na FATEC Ribeirão Preto.

Referências

Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How Learning Works: Seven Research-Based Principles for Smart Teaching*.

AngleSharp Documentation. (2024). *AngleSharp*. Recuperado de <https://anglesharp.github.io/>

Bates, A. W., & Bates, T. (2019). *Teaching in a Digital Age: Guidelines for Designing Teaching and Learning for a Digital Age*.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language: A Reference Manual* (3ª ed.). Addison-Wesley Professional.

Cloud Firestore. (2024). *Cloud Firestore Documentation*. Recuperado de <https://firebase.google.com/docs/firestore>

Cloudflare. (2023). *What is credential stuffing? Credential stuffing vs. brute force attacks*. Cloudflare Learning Center.

Facebook. (2013). *React: A JavaScript Library for Building User Interfaces*.

Recuperado de <https://reactjs.org/>

Garrison, D. R., & Vaughan, N. D. (2018). *Blended Learning in Higher Education: Framework, Principles, and Guidelines*.

Hejlsberg, A. (2000). *The C# Programming Language*.

LambdaTest. (2024). *Testing Methodologies: A Detailed Guide To Software Testing Methodologies*. LambdaTest Learning Hub.

Mitchell, R. (2018). *Web Scraping with Python: Collecting Data from the Modern Web* (2ª ed.). O'Reilly Media.

Pinho, R., & Escudelario, T. (2023). *Webpack and React CLI*. Recuperado de

<https://webpack.js.org/guides/getting-started/>

SIGA. (2024). *Sistema Integrado de Gestão Acadêmica (SIGA)*. Recuperado de

<https://siga.cps.sp.gov.br/aluno/login.aspx>

Wieggers, K. E., & Bach, J. L. (2022). *Engenharia de Requisitos: Uma Abordagem Prática*. Elsevier.