

UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA

Gabriel Assis Erbetta

**MEU HORÁRIO 2: UMA APLICAÇÃO WEB PARA SIMULAÇÃO
DE MATRÍCULA**

*Trabalho apresentado ao DEPARTAMENTO DE CIÊNCIA
DA COMPUTAÇÃO do INSTITUTO DE MATEMÁTICA
da UNIVERSIDADE FEDERAL DA BAHIA como requisito
parcial para obtenção do grau de Bacharel em CIÊNCIA DA
COMPUTAÇÃO.*

Orientador: Rodrigo Rocha Gomes e Souza

Salvador
06 de abril de 2017

Ficha catalográfica.

Erbetta, Gabriel Assis

MeuHorário 2: uma aplicação web para simulação de matrícula/ Gabriel Assis Erbeta– Salvador, 06 de abril de 2017.

52p.: il.

Orientador: Rodrigo Rocha Gomes e Souza.
Monografia (graduação)– UNIVERSIDADE FEDERAL DA BAHIA, INSTITUTO DE MATEMÁTICA, 06 de abril de 2017.

TOPICOS PARA FICHA CATALOGRAFICA.
I. Souza, Rodrigo Rocha Gomes e. II. UNIVERSIDADE FEDERAL DA BAHIA. INSTITUTO DE MATEMÁTICA. III Título.

NUMERO CDD

TERMO DE APROVAÇÃO

GABRIEL ASSIS ERBETTA

MEU HORÁRIO 2: UMA APLICAÇÃO WEB PARA SIMULAÇÃO DE MATRÍCULA

Este Trabalho de Graduação foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovada em sua forma final pelo Departamento de Ciência da Computação da Universidade Federal da Bahia.

Salvador, 06 de abril de 2017

Prof. Dr. Rodrigo Rocha Gomes e Souza
Universidade Federal da Bahia

Prof. Dr. Tiago de Oliveira Januario
Universidade Federal da Bahia

Prof. Dr. Danilo Barbosa Coimbra
Universidade Federal da Bahia

À todos que me ajudaram durante essa longa jornada

AGRADECIMENTOS

Ao meu orientador Rodrigo Rocha pela criação do MeuHorário em 2004 e pela ajuda nesta reta final da graduação. À minha amiga Amanda Sotero pela ajuda no design da aplicação. À minha namorada Aléxia Victória Ladeia pelo incentivo e por suas revisões desta monografia.

Education isn't something you can finish.

— ISAAC ASIMOV (Bill Moyer's World of Ideas, 1989)

RESUMO

Na Universidade Federal da Bahia (UFBA), o processo de matrícula dos discentes é dificultado por causa da pouca transparência de informações das turmas e da baixa qualidade do sistema acadêmico utilizado para matrícula. Nesse contexto surge em 2004 uma aplicação web chamada MeuHorário, que tinha como objetivo auxiliar os estudantes da UFBA, disponibilizando informações sobre as disciplinas e permitindo a simulação da matrícula antes que ela pudesse ser feita no sistema acadêmico da universidade.

Com o passar dos anos, a utilização da aplicação se tornou comum para muitos estudantes da UFBA. No entanto, suas pequenas atualizações ao longo do tempo não foram suficientes para que ela acompanhasse as novas tendências de sistemas web. Por esse motivo, este trabalho traz uma reformulação completa da aplicação. A nova versão foi desenvolvida utilizando novas tecnologias e com preocupações modernas, como a sua adaptabilidade a dispositivos móveis. Este trabalho traz também uma avaliação da versão atualizada e uma comparação de seus acessos com os da versão original.

Palavras-chave: Universidade Federal da Bahia, matrícula, extração de dados da web, aplicação web.

ABSTRACT

The process of class enrollment at Universidade Federal da Bahia (UFBA) is made difficult by the little availability of information about classes and low quality of the academic system used for online enrollment. In this context the MeuHorário web application is created in 2004 with the goal of helping the undergraduate students of UFBA, making information of its courses and classes easily available and allowing the students to simulate their enrollment in classes before it becomes possible to do it in the university's academic system.

As years passed, many of the UFBA students started using MeuHorário regularly. However, its small updates over the course of those years were not enough to keep up with the new trends in web development. For this reason, this dissertation provides a remake of the application. The new version of MeuHorário was developed using recent technology and with modern concerns, such as its adaptability to mobile devices. This dissertation also provides an evaluation of the updated version and a comparison of its visits with the original application.

Keywords: Universidade Federal da Bahia, class enrollment, web scraping, web application.

SUMÁRIO

Capítulo 1—Introdução	1
Capítulo 2—Referencial Teórico	4
2.1 Domínio acadêmico	4
2.2 Extração de dados de páginas web	5
2.2.1 Web Crawler	6
2.2.2 Wrapper	7
2.3 Avaliação de aplicações web	8
2.3.1 Website evaluation methods (WSEM)	8
2.3.1.1 Métodos de inspeção	8
2.3.1.2 Testes empíricos	9
2.3.1.3 Métodos automáticos de avaliação	10
2.3.2 Web evaluation methods (WEM)	10
2.3.3 Métodos utilizados para avaliar o MeuHorário 2	10
2.4 Trabalhos relacionados	11
2.4.1 Universidade Federal de Santa Catarina (UFSC)	11
2.4.2 Universidade de São Paulo (USP)	13
2.4.3 Universidade Tecnológica Federal do Paraná (UTFPR)	13
2.4.4 Universidade Federal de Campina Grande (UFCG)	15
2.4.5 Universidade Federal da Bahia (UFBA)	15
Capítulo 3—A Aplicação	18
3.1 Passos iniciais	18
3.2 Tecnologias utilizadas	19
3.2.1 Ruby on Rails	19
3.2.2 Outras gems (Nokogiri, Mechanize, wicked_pdf)	19
3.2.3 Foundation	20
3.2.4 NGINX	20
3.2.5 DigitalOcean	20
3.3 Funcionamento	21
3.3.1 Selecionar área	21
3.3.2 Selecionar curso	21
3.3.3 Selecionar turmas	23
3.3.4 Resolver conflitos	25
3.3.5 Exportar como PDF	25

3.4	Web Scraping	26
3.5	Arquitetura lógica da aplicação	29
3.5.1	Banco de Dados	29
3.5.2	Models	30
3.5.3	Controllers	32
3.5.4	Views	32
Capítulo 4—Avaliação e resultados		35
4.1	Avaliação Heurística	37
4.2	Mixpanel	39
4.2.1	Coleta dos dados	39
4.2.2	Estatísticas de uso	40
4.3	Sugestões	41
4.4	Comparação de acessos	42
Capítulo 5—Conclusão		47

LISTA DE FIGURAS

2.1	Página inicial do MatrUFSC	12
2.2	Página inicial do MatrUFSC 2	12
2.3	Página inicial do MatrUSP	13
2.4	Página inicial do MatrUSP 2	14
2.5	Página de montagem da grade da segunda versão do Grade na Hora!	14
2.6	Área ‘Minha Grade’ do Cursos UFCG	15
2.7	Página inicial do MeuHorário original	16
2.8	Seleção de turmas no MeuHorário original	16
2.9	Grade de horários gerada pelo MeuHorário original	17
3.1	Página inicial	22
3.2	Listagem de cursos da área 1	22
3.3	Aba de disciplinas obrigatórias mostrando os pré-requisitos (em vermelho) e as disciplinas liberadas (em verde) por uma disciplina (em azul)	24
3.4	Grade de horários com cinco turmas selecionadas	24
3.5	Modal de informações da disciplina	25
3.6	Exibição de conflito de horários	25
3.7	PDF com uma grade de horários exportado pelo MeuHorário 2	27
3.8	Fragmento da página de listagem de disciplinas do curso ciência da computação	27
3.9	Página de listagem de disciplinas obrigatórias (com pré-requisitos) do curso ciência da computação	28
3.10	Listagem de guias de matrícula por curso da área 1 no site da SUPAC	28
3.11	Guia de matrícula do curso Ciência da Computação	28
3.12	Modelo entidade relacionamento do MeuHorário 2	30
3.13	Diagrama de <i>models</i> da aplicação gerado pela <i>gem</i> RailRoady	31
3.14	Diagrama de <i>controllers</i> da aplicação gerado pela <i>gem</i> RailRoady	33
3.15	Disciplinas obrigatórias do curso de ciência da computação (à esquerda) e grade de horários (à direita) do MeuHorário 2 como visto em um smartphone	34
4.1	Número de tracks registrados no MeuHorário 2	40
4.2	Usuários distintos por dia do MeuHorário 2	41
4.3	Dispositivo (à esquerda) e sistema operacional (à direita) de origem dos <i>tracks</i> do MeuHorário 2	42
4.4	De verde o número de acessos feitos por usuários distintos à página do curso e de amarelo o número de acessos totais à página do curso do MeuHorário 2	43
4.5	Usuários distintos por dia do MeuHorário original	45

4.6	Usuários distintos por dia do MeuHorário original (à esquerda) e do MeuHorário 2 (à direita) durante os períodos de matrícula web	45
4.7	Dispositivo (à esquerda) e sistema operacional (à direita) de origem dos <i>tracks</i> do MeuHorário 2	46

LISTA DE TABELAS

3.1	Comparação de recursos dos droplets	21
4.1	As dez heurísticas de Nielsen	36
4.2	Classificação de severidade	37
4.3	Problemas identificados no MeuHorário 2 pela avaliação heurística	38

Capítulo

1

Contextualiza esta monografia, trazendo o histórico, a motivação para o trabalho e a organização dos capítulos.

INTRODUÇÃO

Muitas das universidades brasileiras utilizam aplicações web para permitir aos discentes o acesso a informações pertinentes a sua vida acadêmica, como o seu histórico escolar e a grade curricular do seu e de outros cursos, e a realização da matrícula web, onde os estudantes devem escolher as disciplinas e turmas que irão cursar no semestre letivo seguinte.

Na Universidade Federal da Bahia (UFBA), o sistema responsável por isso é o Sistema Acadêmico, ou SIAC. Porém, assim como o de várias outras universidades, esse sistema é arcaico e inadequado, sendo raramente atualizado e não provendo visualizações práticas e agradáveis das informações.

Uma das principais necessidades dos estudantes da UFBA não supridas pelo SIAC é uma forma de visualizar as turmas que serão oferecidas para matrícula antes do período de matrícula web. Um outro website acadêmico, o da Superintendência de Administração Acadêmica (SUPAC), supre essa necessidade mas o faz de forma inadequada, provendo apenas uma tabela estática em HyperText Markup Language (HTML).

Em 2004 surge então o MeuHorário, aplicação web criada por Rodrigo Rocha que permite a geração de uma grade de horários a partir das informações sobre as turmas dos guias de matrícula disponibilizados pela SUPAC. No entanto, as tecnologias utilizadas em aplicações mudam constantemente (BARUA; THOMAS; HASSAN, 2014) e o MeuHorário, mesmo tendo passado por algumas atualizações, também se tornou ultrapassado.

Doze anos depois de sua criação, em 2016, surge a proposta de fazer uma reformulação completa do MeuHorário. Essa reformulação, chamada nesta monografia de MeuHorário 2, deveria seguir as tendências atuais de desenvolvimento web.

Uma dessas tendências é a criação de aplicações com alto nível de interatividade, utilizando HTML5, as novas versões de JavaScript, *frameworks* como o Foundation e bibliotecas como o jQuery. Anttonen (2011) afirma que a web tem evoluído de páginas estáticas para aplicações completas cada vez mais similares às aplicações convencionais que rodam sobre o sistema operacional.

A outra principal tendência é a construção de aplicações web responsivas, utilizando métodos para fazer com que as páginas se adaptem a telas de diferentes tamanhos de diferentes dispositivos. Essa tendência surge do grande aumento de usuários vindos de dispositivos móveis: smartphones já são mais utilizados para acessar a internet do que computadores no Brasil (IBGE, 2016) e substitui a prática de se desenvolver uma versão diferente do website apenas para dispositivos móveis, por gerar versões limitadas e dificultar a manutenção da aplicação web (COMEAUX, 2016).

O desenvolvimento do MeuHorário 2 ocorreu durante o semestre letivo de 2016.1 da UFBA (de 4 de julho a 31 de outubro de 2016). Para que a aplicação pudesse ser utilizada para auxiliar os estudantes na matrícula do semestre letivo de 2016.2, o prazo final de desenvolvimento foi o último dia de aulas do semestre 2016.1, mas o desejado era que a aplicação ficasse pronta antes da disponibilização dos guias de matrícula, o que costuma ocorrer alguns dias antes do último dia de aulas. A aplicação foi lançada no dia 1 de novembro, no dia seguinte à disponibilização dos guias de matrícula, após pequenos ajustes nos algoritmos de extração de dados para adaptá-los aos novos guias.

O primeiro passo do desenvolvimento foi uma busca por trabalhos relacionados, onde várias aplicações similares foram descobertas em outras universidades do Brasil. A partir do estudo dessas outras aplicações e do conhecimento empírico foram definidos os requisitos da aplicação. O processo de programação dos algoritmos de extração de dados foi então iniciado, uma vez que os requisitos já haviam sido definidos. Essa foi a fase mais longa do desenvolvimento, pois envolvia a extração e a integração de dados provenientes de várias fontes. Com o fim da coleta dos dados iniciou-se o desenvolvimento das páginas web, criadas para serem visualmente agradáveis, responsivas e concentrar parte do processamento localmente, no dispositivo do usuário, para evitar o congestionamento do servidor.

Para validar a qualidade da nova aplicação foi necessário utilizar alguns métodos de avaliação de aplicações web. Após um levantamento dos métodos, envolvendo suas vantagens, desvantagens e popularidade, foram escolhidos três deles, cada um compreendendo diferentes aspectos: uma avaliação heurística, feita por um ou mais especialistas para avaliar a usabilidade da aplicação; um formulário de sugestões, para coletar comentários dos usuários da aplicação; e a análise de estatísticas de uso através do Mixpanel.

A avaliação heurística é um método de avaliação bastante utilizado para aplicações web (ARCHER, 2010) que consiste em julgar a adequação da interface da aplicação à uma lista de heurísticas. A avaliação foi feita pelo próprio desenvolvedor da aplicação e o conjunto de heurísticas utilizado foi o originalmente proposto por Nielsen (1994). Os comentários dos usuários sobre a aplicação são coletados por dentro da mesma, através de um formulário de sugestões. Nele, os usuários escrevem uma mensagem e, se desejarem, adicionam seus nomes e emails para contato. Essas informações são então enviadas para um email administrativo.

As estatísticas de uso foram coletadas durante o período de matrícula através do Mixpanel, uma ferramenta online de mobile e web analytics. A coleta é feita em segundo plano enquanto os usuários utilizam a aplicação e consiste no envio de suas interações para o servidor do Mixpanel. Para que fosse possível fazer uma comparação com o uso do MeuHorário original foi necessário fazer uma coleta similar de seus dados, também

usando o Mixpanel, durante a matrícula anterior a do lançamento da versão atualizada.

Os capítulos desta monografia estão organizados de tal maneira: o Capítulo 2 apresenta informações sobre o domínio acadêmico, a extração de dados da web e a avaliação de aplicações web, além de uma lista de trabalhos similares. No Capítulo 3 são apresentados os principais aspectos da aplicação desenvolvida, incluindo a sua arquitetura e o seu funcionamento. O Capítulo 4 traz uma avaliação da usabilidade da aplicação, as sugestões enviadas por usuários e uma análise de estatísticas de uso, comparando-as com a da aplicação original. Por fim, o Capítulo 5 apresenta uma recapitulação desta monografia, trabalhos futuros e as considerações finais.

Este capítulo fundamenta as bases necessárias sobre o domínio acadêmico utilizado, mineração de dados da web e avaliação de aplicações web para a compreensão desta monografia, além de trazer informações sobre outros trabalhos similares.

REFERENCIAL TEÓRICO

2.1 DOMÍNIO ACADÊMICO

Para o completo entendimento das funcionalidades da aplicação apresentada neste trabalho, se faz necessária a compreensão de diversos aspectos pertencentes ao domínio acadêmico da Universidade Federal da Bahia, tais como órgãos e processos institucionais. Esta seção traz as informações necessárias desses aspectos fundamentados a partir de conhecimento empírico e de informações disponíveis em documentos e sites oficiais da UFBA, em especial o Regulamento de Ensino de Graduação e Pós-Graduação da Universidade Federal da Bahia (2015).

Os cursos de graduação (daqui em diante referidos apenas como cursos, visto que a aplicação abrange apenas esses) são conjuntos de componentes curriculares (disciplina, atividade, estágio, etc.) que conferem grau acadêmico por meio de diploma. Cada curso possui um ou mais currículos, que identificam os componentes curriculares a serem cursados pelos alunos, de acordo com as modalidades disponíveis do curso (bacharelado, licenciatura, etc.). No caso de atualização dos currículos, não há mudança nos currículos seguidos pelos alunos ingressos antes da data de início da vigência do novo currículo.

A partir deste ponto, outros componentes não serão citados. Assume-se que os termos ‘componentes curriculares’ e ‘disciplinas’ são intercambiáveis uma vez que este trabalho não compreende outros tipos de componentes curriculares.

Dentro dos currículos, as disciplinas são classificadas pela sua natureza como obrigatória, optativa ou livre e a ordem na qual devem ser cursadas é definida por uma grade (ou matriz) curricular. Dá-se o nome de pré-requisito a um componente curricular que deve ser cursado anteriormente a outro.

A cada período letivo, a universidade disponibiliza vagas em turmas de diversos componentes curriculares para cada um dos cursos. Os alunos da UFBA devem então fazer suas inscrições semestrais, indicando as disciplinas que irão cursar, observando os pré-requisitos e os limites mínimos e máximos de carga horária. Essa inscrição pode ser feita

em duas fases diferentes, na matrícula web ou na matrícula presencial. As datas de início e fim dessas fases são disponibilizadas no calendário acadêmico, além das datas disponíveis para ajustes de matrícula, sendo que estes só podem ser feitos por alunos que fizeram sua inscrição nos períodos de matrícula web ou presencial.

As informações sobre as turmas que serão ofertadas para cada curso são disponibilizadas para o público no site da Superintendência de Administração Acadêmica¹ (SUPAC) na forma de uma tabela em um documento HTML. Os documentos que contêm essas informações são chamados de guias de matrícula e também fornecem informações quanto ao número de vagas, horários das aulas e os professores que irão ministrar essas turmas. Existem guias de matrícula por cursos e por unidades de ensino.

Por alguns dias ou semanas, até o início da matrícula web, as informações sobre as turmas são encontradas apenas nos guias de matrícula e, a partir de então, são disponibilizadas no Sistema Acadêmico² (SIAC) durante o processo de inscrição semestral. Ao final do período da matrícula web, essas informações voltam a ser encontradas apenas no site da SUPAC.

2.2 EXTRAÇÃO DE DADOS DE PÁGINAS WEB

Com pelo menos 40 bilhões de páginas indexadas (BOSCH; BOGERS; KUNDER, 2016) e com um valor pelo menos 400 a 550 vezes maior de páginas não indexadas na *deep web* (BERGMAN, 2001), a web está cheia de dados. No entanto, estes dados disponíveis na web são quase sempre encontrados de forma não-estruturada ou semi-estruturada graças ao principal formato utilizado, o HTML. Sua grande tolerância a erros no uso de seus elementos e de arquivos mal formatados dificultam bastante a extração de dados estruturados de documentos escritos nesta linguagem.

Isso é facilmente observável nas chamadas páginas dinâmicas, que possuem conteúdo variável normalmente trazendo dados de um banco de dados. Durante o processo de transformação desses dados em elementos HTML, sua estrutura original é substituída por uma que prioriza a visualização e não a extração automatizada de dados.

Existem alguns movimentos que buscam estruturar os dados disponíveis na web, sendo os principais a web semântica, que propõe a utilização de *tags* semânticas do HTML5 e de atributos personalizados a outras *tags* HTML de acordo com *schemas*, a utilização de XHTML, uma combinação do uso das *tags* HTML com as regras mais restritas do XML (W3C, 2000), e o uso de Resource Description Framework (RDF), um modelo padrão para a representação de dados na web³.

Apesar de haver expectativas quanto a utilização de web semântica e APIs para extração de dados da web (WENINGER et al., 2016), sua pouca utilização (apenas 16,5% dos domínios acessados pelo Common Crawl possuem dados estruturados⁴) e o declínio no uso do XHTML (de 65,6% em 2012 para 26% em 2017 segundo o W3Techs⁵) mostram

¹<https://supac.ufba.br>

²<https://siac.ufba.br>

³<https://www.w3.org/RDF/>

⁴<http://webdatacommons.org/structureddata/2016-10/stats/stats.html>

⁵https://w3techs.com/technologies/history_overview/markup_language

que ainda é necessário fazer um processo chamado de *web scraping* para obter dados de forma estruturada da web.

Um *web scraper*, também conhecido como *web data extraction system*, é uma sequência de procedimentos utilizados para a extração de dados de páginas web (LAENDER et al., 2002). Laender (2002) define dois aspectos principais na extração desses dados:

- Interação com as páginas web — Encontrar e acessar os sites que possuam os dados procurados, muitas vezes utilizando *crawlers* e técnicas similares às usadas por indexadores da web;
- Criação e execução de um *wrapper* — Utilização de técnicas para a extração e manipulação de dados de páginas web, transformando-os em dados estruturados.

Uma outra definição de sistemas de extração de dados da web é dada por Baumgartner, Gatterbauer e Gottlob (2009) (como citado por Ferrara et al., 2014), no livro *Encyclopedia of Database Systems* eles os definem como “um software que extrai, automaticamente e repetidamente, dados de páginas web com conteúdos dinâmicos, e que entrega os dados extraídos para um banco de dados ou alguma outra aplicação”. Esta definição introduz três novos aspectos:

- Automação e agendamento — Automação envolve, entre outros, a simulação da interação do usuário e o suporte a Javascript para acesso de conteúdos dinâmicos e o agendamento envolve a repetição da tarefa de extração de forma a manter os dados atualizados no caso de atualizações na página web;
- Transformação dos dados — Padronização da saída de um ou mais wrappers quanto à estrutura de dados obtidos de diferentes fontes;
- Utilização dos dados extraídos — Entrega dos dados extraídos e reestruturados a uma aplicação ou banco de dados ao final do processo de *web scraping*.

2.2.1 Web Crawler

A recuperação das páginas web que contêm os dados procurados é geralmente feita através da utilização de um *web crawler*, uma ferramenta de navegação automatizada de sites. O funcionamento de um *crawler* inicia-se com a inserção de URLs iniciais chamadas *seeds* (geralmente definidas de forma manual). Ao início da sua execução, o *crawler* irá identificar e extrair todas as URLs existentes em *hyperlinks* nessas páginas e inseri-las em uma fila, ele então irá coletar URLs dessa fila e repetir o processo.

Web crawlers geralmente são classificados de acordo com as estruturas de dados usadas em *breadth-first*, que utiliza fila do tipo *first-in first-out* (FIFO), e *preferential*, que utiliza fila de prioridade (LIU, 2007). Enquanto um *breadth-first web crawler* irá percorrer as URLs por ordem de chegada, um *preferential web crawler* irá atribuir prioridades às URLs utilizando diversas métricas, como a relevância do seu conteúdo, sendo bem mais eficiente em manter atualizados índices ou coleções de páginas modificadas em diferentes ritmos e

em percorrer apenas páginas referentes a um tipo de conteúdo (CHAKRABARTI; BERG; DOM, 1999).

Quando utilizado para fins de extração de dados, um *web crawler* deve identificar e guardar as páginas que contêm os dados procurados para futura extração. Para isso ele deve marcar cada página percorrida como uma página alvo, página com dados a serem extraídos, ou como uma página de navegação, página cujo acesso pode ser necessário para alcançar as páginas alvo. As regras para marcação das páginas podem ser definidas manualmente ou construídas automaticamente em tempo de execução utilizando diversas abordagens (VIAKMAA, 2016).

Crawlers também podem ser feitos de forma a acessarem a chamada *deep web*, dessa forma eles irão encontrar também páginas escondidas atrás de formulários e códigos Javascript. Baumgartner e Ledermüller (2005) definem que os principais métodos de navegação da *deep web* para extração de dados são o uso de modelos navegacionais (conjunto de ações simuladas de usuários, como clicar e preencher formulários) e geração visual (que envolve gravar interações de usuários e repeti-las).

2.2.2 Wrapper

Uma vez que as páginas com conteúdos a serem extraídos são identificadas pelo *web crawler*, elas são transferidas para o *wrapper*, que tem a responsabilidade de encontrar os dados dentro da estrutura do documento, extraí-los, reestruturá-los e exportá-los para futuro uso (normalmente para um banco de dados relacional).

De acordo com Liu (2007), um *wrapper* pode utilizar três diferentes abordagens para extração de dados de acordo com seu nível de automação:

- Abordagem manual;
- Wrapper induction;
- Extração automatizada.

O mais simples método de identificação dos dados na estrutura do documento é chamado de abordagem manual. Nessa abordagem, um humano deve identificar manualmente a localização dos dados, olhando as páginas e seus códigos-fonte, e escrever as regras de extração de dados a serem utilizadas pelo *wrapper*. Apesar de ser simples e possuir os melhores resultados (VIAKMAA, 2016), a abordagem manual não é escalável, pois requer a escrita de conjuntos de regras distintos caso haja grande diferença na estrutura das páginas, além da manutenção manual dessas regras caso haja mudanças na estrutura de qualquer uma das páginas.

Wrapper induction é uma abordagem semi-automatizada de extração de dados que utiliza aprendizado supervisionado. Nela, os *wrappers* devem inferir as regras de extração a partir de um conjunto de páginas cujos dados foram previamente identificados de forma manual. As regras resultantes são então utilizadas na extração de dados de outras páginas com formatação semelhante (LIU, 2007).

As técnicas automatizadas que utilizam aprendizado não-supervisionado pertencem à abordagem de extração automatizada. Essas técnicas envolvem a identificação de padrões

e gramáticas em uma ou mais páginas e a posterior geração automatizada do conjunto de regras. Por eliminar completamente o trabalho manual de marcação de dados, esta é a abordagem mais escalável (LIU, 2007).

2.3 AVALIAÇÃO DE APLICAÇÕES WEB

Com o aumento do número de páginas web existentes, especialmente o de páginas comerciais, e a disponibilidade de maiores recursos na criação de aplicações web, também aumenta a necessidade de se construir *websites* de maior qualidade, de forma a prover uma boa experiência aos usuários e se destacar na vastidão de web.

Para garantir a qualidade da experiência dos usuários, fez-se necessário o desenvolvimento de diversos métodos, técnicas e ferramentas para avaliação de aplicações web. Esses costumam ser utilizados na identificação de problemas existentes, para solucioná-los ou minimizar os seus efeitos (WINCKLER; PIMENTA, 2002), mas eles também são utilizados para avaliar o impacto geral de um site na web. Zahran et al. (2014) chamam essas duas abordagens de, respectivamente, *website evaluation methods* (WSEM) e *web evaluation methods* (WEM).

2.3.1 Website evaluation methods (WSEM)

A usabilidade é definida na norma ISO 9241-11 como “medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso” e é o principal foco de avaliação dos WSEMs. Esta categoria abrange a maior parte dos métodos de avaliação e é caracterizada pela utilização de critérios pré-definidos para construir uma lista de problemas de usabilidade e/ou de recomendações de melhorias (ZAHARAN et al., 2014). Winckler e Pimenta (2002) classificam esses métodos como métodos de inspeção e testes empíricos e Zahran et al. (2014) acrescentam a esta taxonomia os métodos automáticos.

2.3.1.1 Métodos de inspeção São chamados de métodos de inspeção aqueles onde os vários aspectos da usabilidade da aplicação são analisados de acordo com a sua conformidade a um conjunto de diretrizes (FERNANDEZ; INSFRAN; ABRAHÃO, 2011). Essa análise é normalmente feita por um ou um pequeno grupo de avaliadores especialistas em usabilidade e/ou no domínio da aplicação. Os principais métodos de inspeção são:

- Avaliação heurística — A avaliação heurística é um método tradicional e um dos mais utilizados para avaliação de usabilidade websites (ARCHER, 2010), principalmente por sua simplicidade e o seu baixo custo de tempo e recursos. Nela, os avaliadores devem interagir com a interface e julgar a sua adequação à uma lista de princípios de usabilidades reconhecidos, chamados de heurísticas. O conjunto de 10 heurísticas propostas originalmente por Nielsen (1994) continua sendo um dos mais utilizados, apesar de diversos novos conjuntos terem sido propostos posteriormente (FERNANDEZ; INSFRAN; ABRAHÃO, 2011);
- Inspeção de recomendações ergonômicas — Neste método, os avaliadores devem

analisar a aplicação buscando violações de um extenso conjunto de recomendações ergonômicas. Apesar de ter um custo relativamente baixo, a utilização dessas listas de recomendações ergonômicas, também chamadas de *guidelines*, restringe a avaliação aos itens presentes nas listas (WINCKLER; PIMENTA, 2002);

- Percurso cognitivo — Esta avaliação consiste na simulação, por avaliadores, de tarefas comumente feitas pelos usuários, com base na suposição de que usuários navegam pela aplicação com a finalidade de executar alguma tarefa (BLACKMON et al., 2002). Os avaliadores devem simular o passo-a-passo da interação do usuário com a aplicação e avaliar a sua adequabilidade para a execução dessas tarefas (BLACKMON et al., 2002).

2.3.1.2 Testes empíricos Testes empíricos, também chamados de métodos de avaliação baseados em usuários, são os métodos que possuem os usuários como figuras centrais na avaliação. Neles são selecionados pequenos grupos de usuários, que devem representar o público da aplicação, para que sejam coletados dados sobre as dificuldades encontradas por eles e suas impressões sobre a aplicação (ZHRAN et al., 2014). Listados abaixo estão os principais testes:

- Ensaio de interação — Similar ao percurso cognitivo, neste método os usuários são instruídos a utilizar a aplicação e realizar um conjunto pré-definido de tarefas. Os passos para a realização das tarefas não devem ser dados aos usuários, pois o objetivo do teste é verificar se esses conseguem realizar as tarefas sozinhos (ZHRAN et al., 2014). As interações dos usuários são então observadas pelos avaliadores que, em alguns casos, podem também fazer perguntas para entender melhor os motivos por trás de algumas de suas ações;
- Think-aloud method — Esse método para avaliação de usabilidade envolve a vocalização dos pensamentos dos usuários enquanto esses utilizam a aplicação. A utilização deste método permite que os avaliadores entendam os motivos por trás das interações desses usuários com o sistema avaliado utilizando um processo bastante simples;
- Questionário — Neste tipo de avaliação, os usuários da aplicação devem preencher um questionário criado pelos avaliadores. Os questionários podem ser utilizados para coletar dados sobre a qualidade da aplicação e os problemas encontrados durante sua utilização, além de dados sobre o perfil dos usuários (WINCKLER; PIMENTA, 2002). Possuem vantagem quando são utilizados para avaliar aplicações web pois a internet é um meio ideal para alcançar o grande número de usuários necessários para coletar dados significativos utilizando formulários online (WINCKLER; PIMENTA, 2002);
- Entrevista — Consiste em entrevistas de usuários da aplicação, realizadas pelos avaliadores, onde são feitas perguntas sobre as suas experiências de uso. O uso de entrevistas permite a coleta de informações mais detalhadas sobre a interação

dos usuários com o sistema avaliado do que os outros métodos (LÁRUSDÓTTIR, 2009).

2.3.1.3 Métodos automáticos de avaliação São os métodos onde outros softwares são utilizados para fazer uma avaliação automática da aplicação web. Podem ser utilizados para avaliar a sua usabilidade, acessibilidade, performance e segurança (BRAJNIK, 2004). Esses softwares geralmente entregam listas dos problemas encontrados após fazerem análises de *logs* e verificações do código ou de recomendações de acessibilidade e ergonomia.

Apesar de oferecerem avaliações de forma rápida e com custos baixos, além de não necessitar de avaliadores experientes, as ferramentas automáticas de avaliação não devem ser exclusivamente utilizadas para análise de usabilidade, pois seus resultados são bastante limitados (ZHRAN et al., 2014). Winckler e Pimenta (2002) afirmam que isto ocorre em parte por causa da necessidade de se adaptar as recomendações das diretrizes para que possam ser verificadas de forma automática.

2.3.2 Web evaluation methods (WEM)

Segundo a taxonomia proposta por Zahran et al. (2014), WEMs são os métodos que buscam medir o impacto do site na web, ao invés da sua usabilidade e adequação. Essa medição é feita utilizando dados de tráfego, interação e ranqueamento, como acessos, padrões de navegação, *page views* e sessões. Esses métodos podem ser divididos em *web analytics* e *link analysis*.

- Web analytics — A *Web Analytics Association* define *web analytics* como “a medição, coleta, análise e documentação de dados da internet com o objetivo de entender e otimizar a utilização da web” (2008). Para isso são utilizadas ferramentas que calculam estatísticas detalhadas de uso do website, usando desde o número de acessos até interações dos usuários com os elementos da página. Sua principal vantagem é a habilidade de coletar informações sobre todos os usuários a um baixo custo (ATKINSON, 2007). Duas das principais ferramentas de *web analytics* são o Google Analytics e o Mixpanel;
- Análise de links — Método que analisa e ranqueia websites de acordo com a quantidade e a qualidade de links apontando para ele. A mais conhecida utilização desse método é o PageRank, algoritmo do Google criado por Brin e Page em 1998 (PAGE et al., 1998) que é utilizado para decidir a ordem na qual os websites devem aparecer nos resultados do buscador, atribuindo um valor para cada site, que é definido em parte pelos links que os referenciam (SCOWEN; REGENBRECHT, 2009).

2.3.3 Métodos utilizados para avaliar o MeuHorário 2

Após consideração dos métodos de avaliação apresentados anteriormente, foi decidido o uso de um método de cada abordagem e um método adicional para avaliação do MeuHorário 2. O WSEM escolhido foi a avaliação heurística, o WEM escolhido foi o uso da

ferramenta Mixpanel para *web analytics* e o método adicional escolhido foi a coleta de sugestões dos usuários para suprir a falta de testes empíricos.

A inspeção heurística foi escolhida principalmente pela sua alta eficiência, ou seja, a sua capacidade de identificar um grande número de problemas de usabilidade com um baixo custo de recursos e tempo. Além disso, ela também pode ser realizada com algum sucesso por não-especialistas em usabilidade (BAKER; GREENBERG; GUTWIN, 2002) apesar de ser pensada para ser feita por especialistas. O conjunto de heurísticas utilizado foi o proposto originalmente por Nielsen (1994), por ser um conjunto tradicional, muito utilizado até hoje e por não haver consenso em qual conjunto de heurísticas deve ser utilizado para avaliação de aplicações web, apesar de vários outros terem sido propostos (HERMAWATI; LAWSON, 2016).

Já o Mixpanel foi selecionado por sua abordagem híbrida de *web analytics*, visto que ele tanto coleta informações sobre os usuários quanto o histórico de interações deles com o website. Seu funcionamento se dá por meio dos chamados *tracks*, ou rastros. Os desenvolvedores da aplicação definem ações dos usuários que devem ser observadas e os *tracks* ligados a elas. Quando os usuários realizam essas ações, dados sobre os seus *tracks* (contendo o nome do *track*, parâmetros adicionais personalizados e informações do usuário) são enviados para o servidor do Mixpanel, onde eles ficam disponíveis para futuro acesso.

Porém, como a recomendação de muitos autores é a utilização de ensaios de interação em paralelo à avaliação heurística (NIELSEN, 1994; ARCHER, 2010; ZAHRAN et al., 2014), também foi utilizado um formulário de sugestões na aplicação como um método substituto de avaliação baseado em usuários. Seu uso tem como objetivo coletar não apenas sugestões de melhorias, mas também comentários gerais sobre a aplicação. Um resumo das sugestões coletadas pode ser encontrado na Seção 4.3.

2.4 TRABALHOS RELACIONADOS

Os curtos períodos de matrícula comuns à várias das universidades brasileiras, em conjunto com as dificuldades de preparação dos alunos para este período, como a disponibilização ineficiente da listagem de turmas e a falta de integração dos sistemas de matrícula com as grades curriculares dos cursos, causaram o desenvolvimento de vários trabalhos similares em lugares e períodos diversos e de forma independente, tanto por estudantes quanto por grupos de pesquisa.

2.4.1 Universidade Federal de Santa Catarina (UFSC)

A Universidade Federal de Santa Catarina possui um grande histórico de aplicações criadas para auxiliar os alunos em suas matrículas. O primeiro deles, o Grade de Matrícula (GRAMA), foi desenvolvido no início dos anos 2000 por Glauco Peron, aluno de engenharia de produção e possuía o apoio da própria universidade. Após alguns anos, a aplicação perdeu o apoio da UFSC, foi retirada do ar e encontra-se indisponível.

Em 2012 surge o MatrUFSC⁶ (Figura 2.1), também conhecido como Combinador

⁶<http://ramiro.arrozcru.org/matrufsc>

Automático de Possibilidades Interativo de Matrícula (CAPIM), sucessor do GRAMA, desenvolvido pelo estudante Ramiro Polla durante a sua graduação em engenharia elétrica. O sistema se propõe a gerar uma grade de horários para o usuário com as disciplinas selecionadas a partir de uma busca. Apesar da semelhança de objetivos, esta aplicação difere-se do MeuHorário 2 principalmente por não levar em consideração os cursos dos alunos ao não exibir informações como a grade curricular e por buscar resolver os conflitos de horários de forma automática, atribuindo prioridades às disciplinas selecionadas.

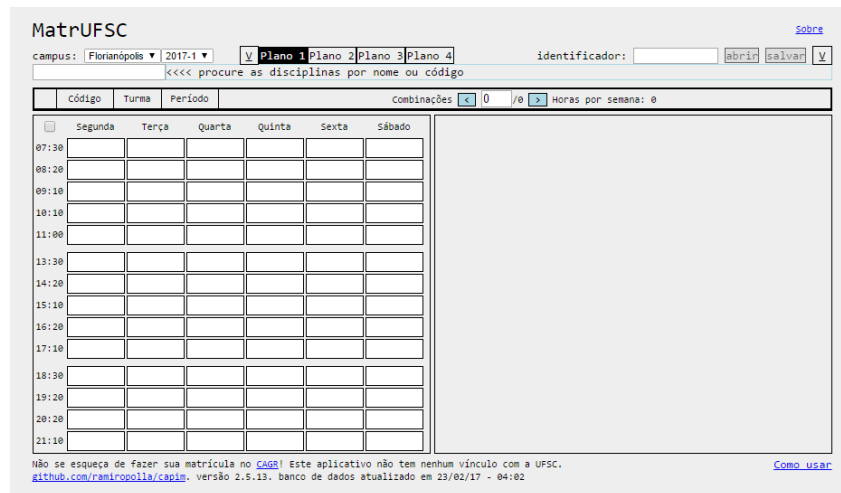


Figura 2.1 Página inicial do MatrUFSC

Dois anos depois do lançamento do MatrUFSC, um sucessor foi lançado com o nome de MatrUFSC 2⁷ (Figura 2.2). Desenvolvido por Fernando Mota, a aplicação é uma remodelagem do seu antecessor e traz também uma funcionalidade para seleção de horários disponíveis do estudante.

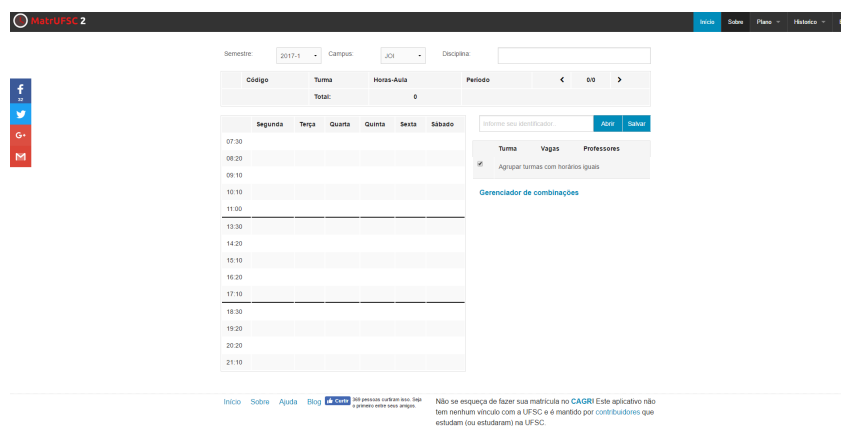


Figura 2.2 Página inicial do MatrUFSC 2

⁷<http://matrufsc2.appspot.com>

2.4.2 Universidade de São Paulo (USP)

Após o seu lançamento, o MatrUFSC chamou a atenção do projeto Apoio ao BCC da USP, formado por alunos da universidade. Estes resolveram adaptar o sistema de matrícula criado por Ramiro para a sua universidade. Em 2013 o MatrUSP⁸ (Figura 2.3) foi lançado possuindo as mesmas funcionalidades da aplicação original mas com as disciplinas da Universidade de São Paulo.

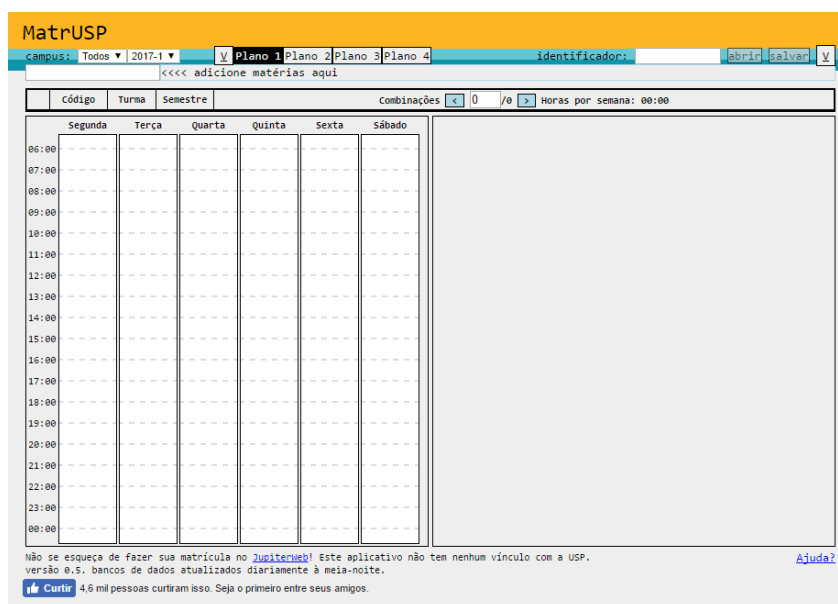


Figura 2.3 Página inicial do MatrUSP

Em 2016, o MatrUSP passou por uma remodelagem completa nas mãos dos estudantes Bruno Endo, José Rodrigues e Rafael Verona. A nova versão da aplicação conta com um funcionamento bastante similar ao seu predecessor mas com uma nova e melhorada interface gráfica⁹ (Figura 2.4).

2.4.3 Universidade Tecnológica Federal do Paraná (UTFPR)

No ano de 2012, alunos de engenharia da computação da UTFPR desenvolveram uma aplicação para simulação de matrícula de sua universidade chamada Grade na Hora!¹⁰. Em sua versão inicial, o *crawling* das turmas era feito ao colar na aplicação o código-fonte da página de turmas disponíveis para o curso do usuário, retirado do sistema acadêmico da UTFPR.

Ainda no mesmo ano do lançamento, a versão 2.0 do Grade na Hora! foi disponibilizada (Figura 2.5). Esta nova versão trazia os cursos do campus de Curitiba já cadastrados no sistema, eliminando a necessidade de acesso ao sistema acadêmico pelos

⁸http://bcc.ime.usp.br/matrusp_v1

⁹<http://bcc.ime.usp.br/matrusp>

¹⁰<http://gradenahora.com.br>

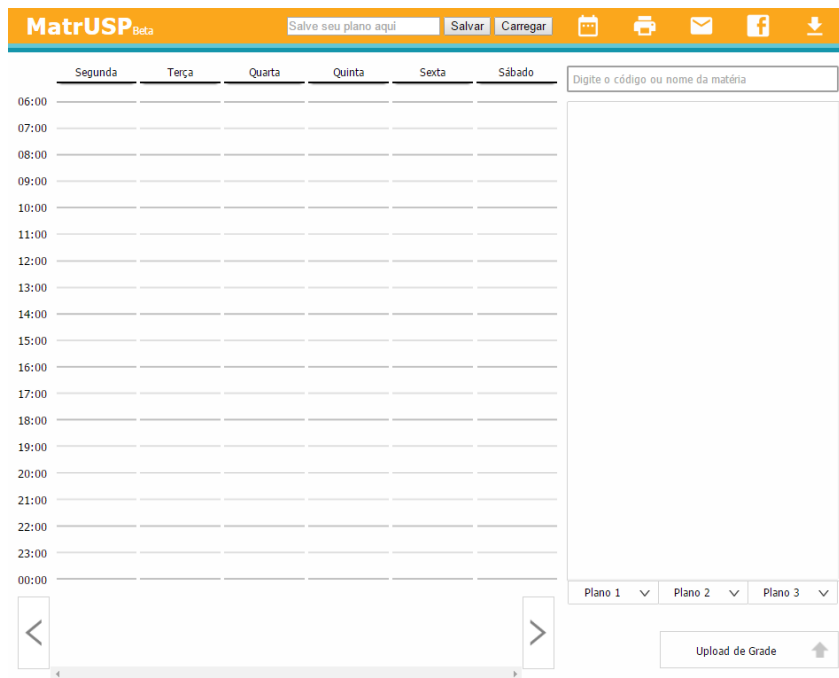


Figura 2.4 Página inicial do MatrUSP 2

usuários. Desde então, a aplicação vem sendo atualizada com a inserção de cursos de outros campi, abrangendo mais estudantes.



Figura 2.5 Página de montagem da grade da segunda versão do Grade na Hora!

O Grade na Hora! assemelha-se ao MeuHorário 2 em sua proposta de diferenciar os usuários por curso e deixar com que eles resolvam os conflitos de horário manualmente. No entanto, ele não possui informações adicionais dos cursos, como a grade curricular, e a grade final pode ser exportada apenas em formato de texto, aceito pela matrícula web do sistema acadêmico da UTFPR.

2.4.4 Universidade Federal de Campina Grande (UFCG)

No último mês de 2015, o Laboratório Analytics, um laboratório de pesquisa da Universidade Federal de Campina Grande, lançou o Cursos UFCG¹¹ (Figura 2.6), uma plataforma online que permite aos usuários visualizarem de forma intuitiva as grades curriculares dos cursos de graduação da UFCG, além de trazer dados estatísticos de aprovação das disciplinas, taxa de correlação entre matérias (baseada no desempenho de alunos) e a possibilidade de selecionar as disciplinas que já cursou e que pretende cursar. Em 2016, uma atualização trouxe também o cálculo do risco de reprovação após a montagem da grade do estudante.

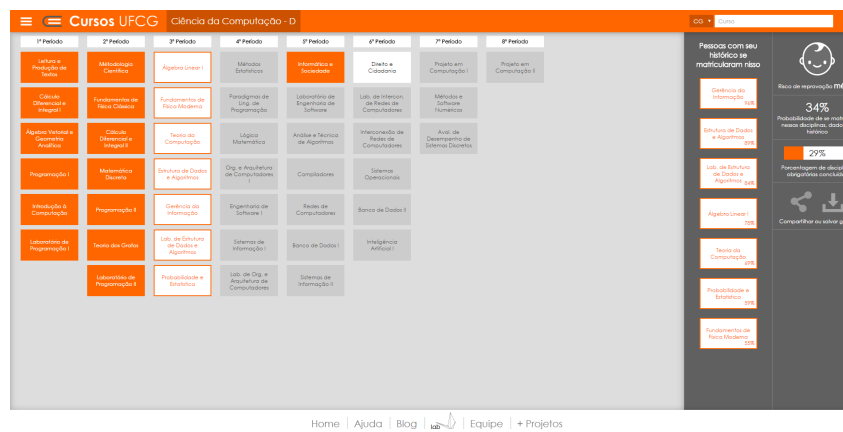


Figura 2.6 Área ‘Minha Grade’ do Cursos UFCG

Diferente das ferramentas citadas anteriormente, o Cursos UFCG não faz a simulação de matrícula dos alunos e não traz nenhuma informação sobre turmas e horários. Seu foco é em disponibilizar informações curriculares dos cursos e análise de dados estatísticos de matrícula e aprovação nas disciplinas.

2.4.5 Universidade Federal da Bahia (UFBA)

O MeuHorário¹² (Figura 2.7) era até então a única aplicação de simulação de matrícula da Universidade Federal da Bahia. Surgiu em 2004, criada pelo então bacharelado em ciência da computação, Rodrigo Rocha. Seu principal atrativo é a seleção automática das turmas a serem cursadas pelo estudante a partir das disciplinas selecionadas (Figura 2.8), trazendo todas as possíveis combinações que não produzem choques de horário (Figura 2.9).

A aplicação trouxe bastante inovação em seu lançamento e com o passar dos anos se concretizou como uma ferramenta comumente utilizada pelos alunos da UFBA em períodos de matrícula. O MeuHorário passou por algumas atualizações desde 2004 mas não houve nenhuma grande alteração em suas funcionalidades, permanecendo com a

¹¹<http://analytics.lsd.ufcg.edu.br/cursosufcg>

¹²<http://meuhorario.dcc.ufba.br/index2.php>



Figura 2.7 Página inicial do MeuHorário original

obrigatoriedade de selecionar todas as disciplinas desejadas antes de descobrir se haverá ou não conflito de horários, dificultando a troca de disciplinas.



Figura 2.8 Seleção de turmas no MeuHorário original



Figura 2.9 Grade de horários gerada pelo MeuHorário original

Este capítulo traz informações sobre o planejamento e o desenvolvimento do MeuHorário 2. Também são apresentadas a sua estrutura interna e o seu funcionamento.

A APLICAÇÃO

3.1 PASSOS INICIAIS

O desenvolvimento do MeuHorário 2 foi iniciado em julho de 2016, com a pesquisa e estudo de aplicações similares. Após a análise dessas aplicações, em especial do MeuHorário original, e de conversas informais com outros estudantes da UFBA, foram definidos os requisitos para o MeuHorário 2. Esses requisitos foram a base da reformulação da aplicação original, resultando em um sistema mais agradável de se utilizar, juntando aspectos positivos das outras aplicações e também trazendo novidades.

Os requisitos definidos foram:

- Informações sobre os cursos e suas grades de matrícula — a aplicação deveria trazer as grades de matrículas dos cursos da UFBA, mostrando tanto as disciplinas obrigatórias quanto as optativas;
- Informações das turmas ofertadas para o próximo semestre — incluindo todas as várias informações presentes na tabela das turmas disponível no SUPAC, como horários, professores e o número de vagas para os cursos;
- Informações sobre os pré-requisitos das disciplinas para os cursos — deveria mostrar visualmente os pré-requisitos das disciplinas dentro das grades de matrícula dos cursos;
- Visualização em tempo real de uma grade de horários com as turmas escolhidas — ao invés da geração automática da grade do MeuHorário original, o MeuHorário 2 deveria permitir que o usuário montasse a grade manualmente selecionando uma turma por vez;
- Habilidade de selecionar e desfazer a seleção de turmas em tempo real — a montagem da grade em tempo real traz também a necessidade de tornar a seleção e o cancelamento da seleção de turmas simples e rápidos para o usuário;

- Identificação de conflitos de horários entre turmas — deveria mostrar visualmente conflitos de horários para que pudessem ser resolvidos;
- Alguma forma de salvar o resultado final das turmas escolhidas — como a aplicação deveria ser utilizada muitas vezes antes do período de matrícula, deveria haver uma forma de salvar a seleção das turmas para este período;
- Responsividade da interface gráfica — todas as funcionalidades da aplicação deveriam ser corretamente traduzidas para telas de todos os tamanhos;

3.2 TECNOLOGIAS UTILIZADAS

Após a definição dos oito requisitos apresentados acima, iniciou-se o processo de pesquisa e escolha das tecnologias a serem utilizadas. As principais delas foram as seguintes: Ruby, Rails, Nokogiri, Mechanize, wicked_pdf, Foundation, NGINX e DigitalOcean.

3.2.1 Ruby on Rails

Ruby¹ é uma linguagem de programação multiparadigma *open-source* desenvolvida por Yukihiro Matsumoto e lançada em 1995. Ruby é muito conhecida pela sua sintaxe similar a linguagem natural e por suas bibliotecas, que levam o nome de *gems*. A sua *gem* mais famosa é a Rails², um *framework* que acelera o desenvolvimento de aplicações web utilizando a arquitetura *Model-View-Controller* (MVC). A partir do lançamento de Rails, em 2005, a linguagem Ruby passou a ganhar grande popularidade e hoje é a décima segunda linguagem de programação mais popular do mundo segundo o índice TIOBE de março de 2017³ e a quinta mais utilizada para programação *web server-side*⁴.

Ruby on Rails, um outro nome para a utilização de Ruby em conjunto com a sua *gem* Rails, foi escolhido para o desenvolvimento do MeuHorário 2 principalmente por causa da experiência prévia do desenvolvedor com a linguagem, mas também pelo seu uso para o *crawling* do MeuHorário original.

3.2.2 Outras gems (Nokogiri, Mechanize, wicked_pdf)

Além do Rails, outras *gems* foram utilizadas no desenvolvimento desta aplicação, destacam-se entre elas: Nokogiri⁵, Mechanize⁶ e wicked_pdf⁷.

Tanto o Nokogiri quanto o Mechanize são utilizados na extração de dados de páginas web. O primeiro é uma *gem* que permite o *parsing* de documentos HTML para que se possa fazer o *scraping* utilizando as *tags* HTML e seus atributos, já o segundo é uma biblioteca utilizada para automatizar a interação com websites, permitindo o envio de

¹<https://www.ruby-lang.org>

²<http://rubyonrails.org>

³<https://www.tiobe.com/tiobe-index>

⁴https://w3techs.com/technologies/overview/programming_language/all

⁵<http://www.nokogiri.org>

⁶<https://github.com/sparklemotion/mechanize>

⁷https://github.com/mileszs/wicked_pdf

formulários e a utilização de *cookies* como em um navegador web, algo que se mostrou necessário para a aplicação.

Já o `wicked_pdf` é utilizado para gerar documentos PDF a partir de documentos HTML, facilitando a implementação da funcionalidade de exportação para PDF da grade selecionada pelos usuários.

3.2.3 Foundation

Para o desenvolvimento da interface da aplicação, foi utilizado o Foundation⁸, um *framework front-end* gratuito que facilita o desenvolvimento de sites responsivos, utilizando a filosofia *mobile-first*, onde os sites são pensados inicialmente para telas de smartphones e posteriormente adaptados para as telas maiores de notebooks e desktops.

Também foi utilizado o jQuery⁹, uma biblioteca JavaScript, pois é um requisito para o funcionamento do Foundation e também acelera a programação de interações do usuário com o website através do JavaScript.

3.2.4 NGINX

O servidor HTTP utilizado para o MeuHorário 2 é o NGINX (engine X), um servidor web *open source* que tem ganhado muitos adeptos ao longo dos últimos anos¹⁰¹¹. Essa escolha foi feita por seu melhor desempenho e menor consumo de recursos do servidor em relação ao seu principal concorrente, o Apache HTTP Server (PRAKASH; BIJU; KAMATH, 2015)(CISNEIROS; RAMOS, 2015), além de sua fácil configuração.

3.2.5 DigitalOcean

Aplicações como o MeuHorário possuem números irregulares de acesso durante o ano, com grandes picos durante o período que abrange desde o final de um semestre letivo até o final do período de matrícula para o semestre letivo seguinte e longos períodos de escassez de acesso durante os períodos de aula. A partir dessas informações, o DigitalOcean¹² foi escolhido para a hospedagem da aplicação.

O DigitalOcean é uma empresa que oferece servidores virtuais privados na nuvem, chamados de *droplets*. Esses *droplets* são fáceis de configurar, pois podem ser construídos a partir de imagens prontas que contêm várias aplicações já instaladas e configuradas, e também são altamente customizáveis, com *droplets* podendo ter seus recursos (como memória e processamento) rapidamente expandidos e reduzidos sem nenhuma dificuldade. Dessa forma, é possível aumentar o tamanho do *droplet* da aplicação durante os períodos de pico e reduzi-lo posteriormente.

A aplicação roda em um *droplet* do DigitalOcean que é redimensionado de acordo com o período de matrícula e a quantidade de acessos. Dos nove tamanhos para *droplets*

⁸<http://foundation.zurb.com>

⁹<https://jquery.com>

¹⁰https://w3techs.com/technologies/history_overview/web_server

¹¹<https://news.netcraft.com/archives/2017/02/27/february-2017-web-server-survey.html>

¹²<https://www.digitalocean.com>

Tabela 3.1 Comparação de recursos dos droplets

	Droplet 1	Droplet 3
Memória RAM	512 MB	2 GB
Número de CPUs	1 CPU	2 CPUs
Disco	20 GB SSD Disk	20 GB SSD Disk
Limite de Transferência	1000 GB	3 TB
Custo Mensal	US\$5	US\$20

que o DigitalOcean disponibiliza, dois são utilizados. O primeiro é utilizado por padrão, devido ao seu baixo custo, e o segundo é utilizado durante os períodos de férias, quando há picos no acesso, para evitar um sobrecarregamento no servidor. Suas especificações estão descritas na Tabela 3.1.

3.3 FUNCIONAMENTO

A seguir será mostrado o funcionamento do MeuHorário 2. Cada uma das subseções mostrará um dos passos principais realizados pelos usuários para montar uma grade de horários na aplicação e exportá-la no formato PDF. Esses passos são:

1. Selecionar área;
2. Selecionar curso;
3. Selecionar turmas;
4. Resolver conflitos;
5. Exportar como PDF.

3.3.1 Selecionar área

Ao acessar a página inicial da aplicação, os usuários se deparam com uma listagem das áreas em que os cursos são separados, mostrada na Figura 3.1. As áreas são exibidas em blocos retangulares que levam para as páginas de seleção de curso quando clicadas.

3.3.2 Selecionar curso

Após selecionar uma área, os usuários são direcionados para uma página com a listagem dos cursos da área selecionada (Figura 3.2). A lista dos cursos traz o código e o nome do curso e é ordenada alfabeticamente.

Para agilizar a seleção do curso desejado, a página conta com um *input* (campo para entrada de dados pelo usuário) textual para busca do curso. Ao inserir caracteres no *input*, é feita uma filtragem na lista de cursos, mostrando apenas aqueles que contêm a sequência de caracteres inserida. Os cursos podem ser filtrados tanto pelo código quanto pelo nome. Essa filtragem é feita utilizando JavaScript e é executada localmente no dispositivo do usuário, aumentando a sua performance por não ter a sobrecarga de enviar ou receber requisições do servidor.



Figura 3.1 Página inicial

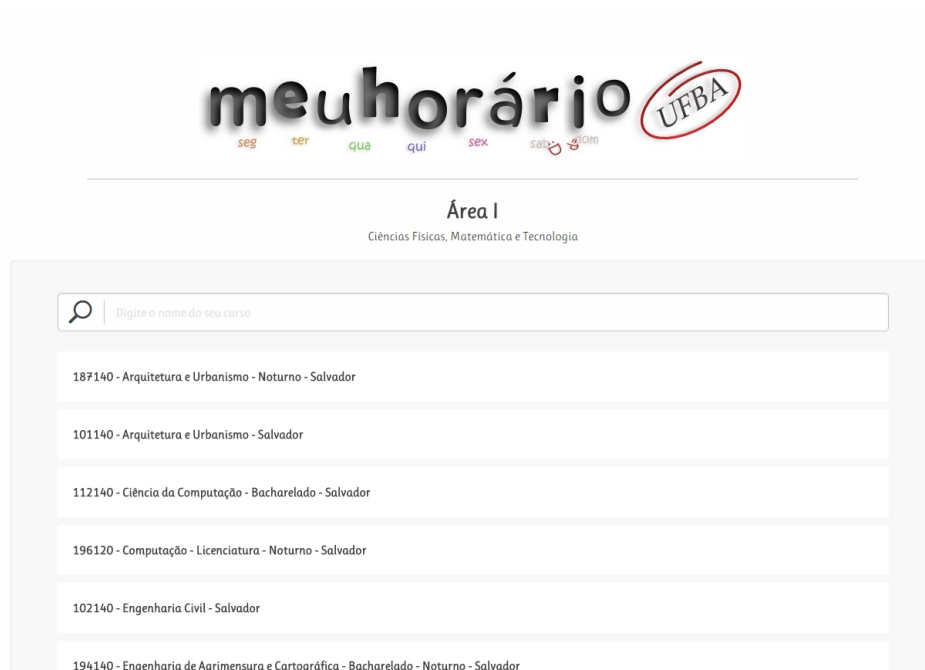


Figura 3.2 Listagem de cursos da área 1

3.3.3 Selecionar turmas

A seleção de um dos cursos na etapa anterior leva os usuários para uma nova página contendo os elementos necessários para a seleção de turmas e montagem da grade de horários. O conteúdo dessa página pode ser dividido em três blocos:

O primeiro deles é um bloco que contém as disciplinas existentes na UFBA. Esse bloco possui quatro abas e cada uma delas mostra um conjunto diferente de disciplinas. São elas:

- Obrigatórias — Mostra as disciplinas de natureza obrigatória do curso selecionado. Conta com uma exibição gráfica do currículo desse curso, separando as disciplinas por semestres e realçando os pré-requisitos e as disciplinas liberadas referentes a uma disciplina quando o ponteiro do mouse pairar sobre ela (Figura 3.3);
- Optativas — Traz as disciplinas de natureza optativa do curso. Por possuírem essa natureza, muitas das disciplinas aqui listadas podem não possuir turmas com vagas disponíveis para alunos do curso selecionado. Possui um *input* textual muito similar ao de filtragem de cursos mostrado na subseção anterior, porém filtrando disciplinas por seu código e nome;
- Ofertadas — Possui uma listagem com todas as disciplinas ofertadas para aquele curso, ou seja, todas que estão no guia de matrícula do curso. Nessa listagem estão presentes disciplinas de natureza obrigatória, optativa e não definida. Possui também um *input* para filtragem das disciplinas;
- Todas — Aba que possui um formulário de busca que permite encontrar qualquer disciplina da aplicação (independente do curso) por seu código ou nome.

O segundo bloco é uma grade de horários onde é possível visualizar os horários das turmas selecionadas e a existência de possíveis conflitos de horário (Figura 3.4). As turmas são representadas por retângulos coloridos contendo o código da sua disciplina.

O terceiro bloco só é visível quando um usuário clica nas disciplinas dos blocos anteriores e consiste em um modal para a visualização de informações da disciplina escolhida (Figura 3.5). Sua principal área é um *accordion*, menu horizontal que se expande para mostrar informações adicionais de cada um de seus itens, onde é possível visualizar as informações da disciplina.

O *accordion* possui 4 itens: pré-requisitos, matérias liberadas, turmas ofertadas para o curso e outras turmas. Nos dois primeiros é possível clicar nas disciplinas listadas para passar a ver informações delas no modal e nos dois últimos é possível ver informações sobre as turmas (como dias e horários das aulas e professores que a lecionam). Ao clicar em uma dessas turmas ela é selecionada e fica visível na grade de horários. Um clique em uma turma já selecionada irá desfazer a sua seleção e um clique em uma outra turma de uma disciplina que já possui uma turma selecionada irá remover a turma já selecionada da grade de horários e adicionar a clicada.

O modal também possui uma réplica idêntica do segundo bloco para que os usuários possam ver imediatamente os efeitos da seleção de uma turma na sua grade de horários.

Obrigatórias	Optativas	Ofertadas	Todas			
Semestre						
Disciplinas						
1	MATA42 Matemática Discreta I	MATA39 Seminários de Introdução ao Curso	MATA38 Projeto de Circuitos Lógicos	MATA37 Introdução à Lógica de Programação	MATA02 Cálculo A	MATA01 Geometria Analítica
2	MATA97 Matemática Discreta II	MATA95 Complementos de Cálculo	MATA57 Laboratório de Programação I	MATA48 Arquitetura de Computadores	MATA40 Estruturas de Dados e Algoritmos I	MATA07 Álgebra Linear A
3	MATA55 Programação Orientada a Objetos	MATA50 Linguagens Formais e Autômatos	MATA49 Programação de Software Básico	MATA47 Lógica para Computação	MAT236 Métodos Estatísticos	FIS495 Elementos da Eletromagnetismo e de Circuitos Elétricos
4	MATA68 Computador, Ética e Sociedade	MATA61 Compiladores	MATA58 Sistemas Operacionais	MATA52 Análise e Projeto de Algoritmos	MATA51 Teoria da Computação	FCH45 Metodologia e Expressão Técnico-Científica
5	MATE12 Paradigmas de Linguagens de Programação A	MATA62 Engenharia de Software I	MATA59 Redes de Computadores I	MATA54 Estruturas de Dados e Algoritmos II	MATA53 Teoria dos Grafos	
6	MATE11 Engenharia de Software II-A	MATA88 Fundamentos de Sistemas Distribuídos	MATA65 Computação Gráfica	MATA64 Inteligência Artificial	MATA60 Banco de Dados	
7	MATA66 Projeto Final de Curso I					
8	MATA67 Projeto Final de Curso II					

Figura 3.3 Aba de disciplinas obrigatórias mostrando os pré-requisitos (em vermelho) e as disciplinas liberadas (em verde) por uma disciplina (em azul)

	SEG	TER	QUA	QUI	SEX	SAB
07:00						
07:55						
08:50						
09:45						
10:40						
11:35						
13:00			MATA62			
13:55			MATA62			
14:50	MATA54		MATA54		MATA62	
15:45	MATA54		MATA54		MATA62	
16:40		MATA52		MATA52		
17:35		MATA52		MATA52		
18:30	MATA61	MATA57	MATA61			
19:25	MATA61	MATA57	MATA61			
20:20		MATA57				
21:15						

▲ Topo Exportar como PDF

Figura 3.4 Grade de horários com cinco turmas selecionadas

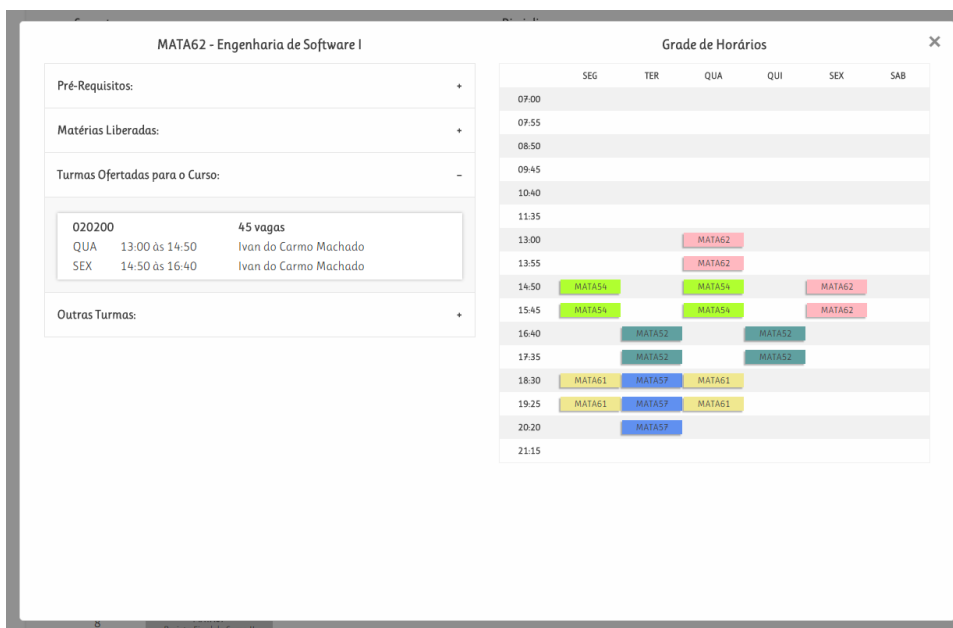


Figura 3.5 Modal de informações da disciplina

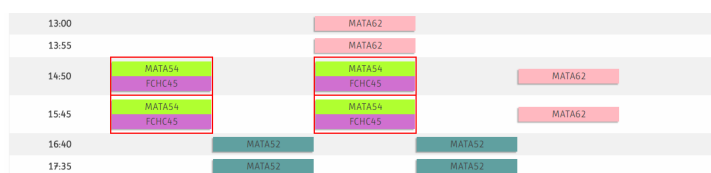


Figura 3.6 Exibição de conflito de horários

3.3.4 Resolver conflitos

Conflitos de horário ocorrem quando duas ou mais turmas selecionadas possuem aulas em um mesmo dia e horário, e são identificadas por uma borda vermelha na célula da grade de horários que representa o horário onde há o conflito (Figura 3.6). Os conflitos são visualizados e resolvidos na mesma página de seleção de turmas.

Os conflitos de horário precisam ser resolvidos manualmente. Para isso basta que os usuários desfaçam a seleção das turmas conflitantes até que sobre apenas uma turma por horário. O processo de desfazer a seleção é feito apenas pelo modal de informações da disciplina a qual a turma se refere. Para evitar que os alunos precisem procurar essas disciplinas em uma das abas do bloco de disciplinas da UFBA, os retângulos das turmas também são clicáveis e abrem o modal com as informações de suas disciplinas.

3.3.5 Exportar como PDF

No final da mesma página de seleção de turmas existe um botão chamado "Exportar como PDF". Ao ser clicado, um documento PDF é gerado e aberto em uma nova guia do navegador. Esse documento possui uma réplica da grade de horários com o escalonamento

das turmas selecionadas e uma legenda com informações detalhadas de cada uma dessas turmas. A Figura 3.7 mostra um exemplo de PDF gerado pela aplicação.

3.4 WEB SCRAPING

O primeiro aspecto da aplicação a ser desenvolvido foi o de extração de dados da web, pois era necessário primeiramente identificar as informações que podiam ser utilizadas na aplicação, que estavam disponíveis para livre acesso e que sejam atualizadas regularmente pela Universidade Federal da Bahia.

Devido ao escopo reduzido do MeuHorário 2, tanto o *crawler* quanto o *wrapper* utilizados na aplicação foram desenvolvidos usando abordagens manuais. As páginas com as informações foram encontradas após pesquisa manual nos vários sites da UFBA e as regras para a extração das informações dessas páginas foram definidas diretamente no código após a análise da estrutura das suas *tags* HTML.

O processo de *web scraping* foi dividido em cinco etapas, cada uma referindo-se ao tipo de informação extraída: cursos, disciplinas, pré-requisitos, turmas e áreas. Para cada uma dessas etapas, foi desenvolvida uma *rake task*, tarefas do Ruby que podem ser executadas mesmo que a aplicação não esteja em execução.

Na primeira tarefa, a de cursos, uma página de listagem dos cursos da UFBA foi utilizada como *seed* para o *crawler* e das páginas individuais dos cursos acessíveis através dela foram extraídos os seguintes dados: nome e código do curso, além do semestre de início da vigência do seu currículo atual (chamado de período do currículo). A forma que esses dados estão disponíveis é visível na Figura 3.8.

As informações dos códigos dos cursos e dos períodos de seus currículos foram utilizadas para as duas *tasks* seguintes, *scraping* de disciplinas e de pré-requisitos. As páginas *seeds* foram páginas públicas das grades curriculares dos cursos, cujo acesso é feito utilizando o código do curso e o período do currículo como parâmetros de uma requisição HTTP GET. Essas páginas *seeds* contêm dois links, um que leva a uma página com informação das disciplinas obrigatórias (Figura 3.9) e a outra das optativas do currículo acessado. No entanto, o acesso a essas duas páginas só é possível com a utilização de *cookies*, pois o curso e currículos que estão sendo acessados são guardados em um *cookie*. Uma vez acessadas, foi possível encontrar nessas duas páginas o nome e código das disciplinas, além da natureza, o semestre e os pré-requisitos delas de acordo com cada currículo.

As últimas duas *tasks*, turmas e áreas, também utilizam um mesmo site para extrair suas informações, o site da SUPAC. Nele existe uma área que possui links para os guias de matrícula de cada curso. Esses cursos estão separados em categorias, chamadas de áreas na aplicação (apesar dessa divisão não ser feita apenas pelas áreas de ensino dos cursos) e dessa categorização é extraída a área de cada curso pela *task* de áreas. Essa categorização é utilizada também na aplicação para que seja igual a do site da SUPAC e também a do MeuHorário original. A Figura 3.10 mostra a listagem de cursos da Área 1 no site da SUPAC e a Figura 3.11 mostra um fragmento de um guia de matrícula.

Dos guias de matrícula são tiradas as informações sobre as turmas de um semestre letivo, são elas: os números das turmas, os dias, horários e os professores que lecionam

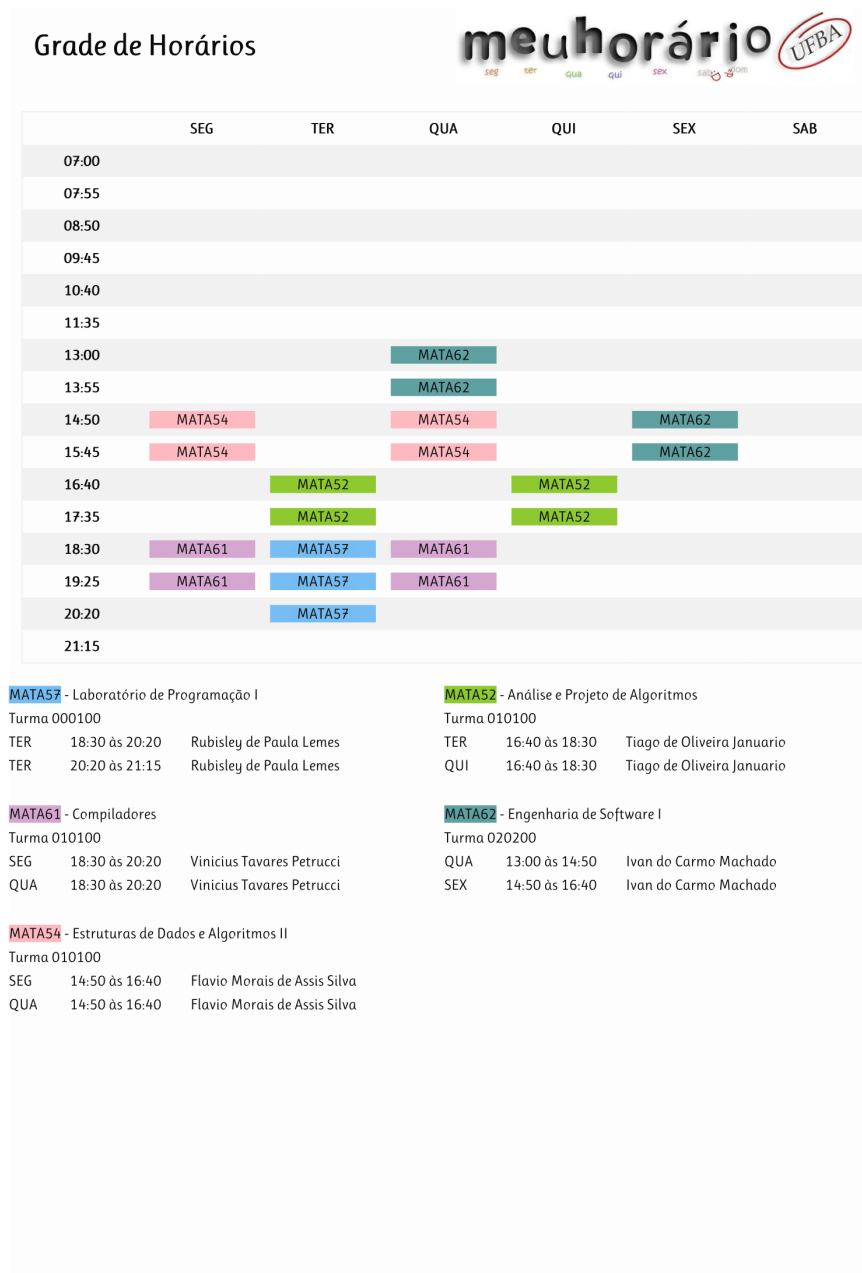


Figura 3.7 PDF com uma grade de horários exportado pelo MeuHorário 2

Curso	Turno	Duração Mínima	Duração Máxima	Período do Currículo
112140 - CIÊNCIA DA COMPUTAÇÃO	Diurno	4.0	7.0	2013.2

Figura 3.8 Fragmento da página de listagem de disciplinas do curso ciência da computação

LISTA DE DISCIPLINAS OBRIGATORIAS - CIENCIA DA COMPUTAÇÃO

	Natureza	Código	Disciplina	Pré-requisitos
1ª Sem.	OB	MATA01	GEOMETRIA ANALITICA	--
	OB	MATA02	CALCULO A	--
	OB	MATA37	INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO	--
	OB	MATA38	PROJETO DE CIRCUITOS LÓGICOS	--
2ª Sem.	OB	MATA39	SEMINÁRIOS DE INTRODUÇÃO AO CURSO	--
	OB	MATA42	MATEMÁTICA DISCRETA I	--
	OB	MATA07	ÁLGEBRA LINEAR	MATA01
	OB	MATA40	ESTRUTURAS DE DADOS E ALGORITMOS I	MATA37 / MATA42
	OB	MATA48	ARQUITETURA DE COMPUTADORES	MATA38
3ª Sem.	OB	MATA57	LABORATÓRIO DE PROGRAMAÇÃO I	MATA37
	OB	MATA95	COMPLEMENTOS DE CÁLCULO	MATA01 / MATA02
	OB	MATA97	MATEMÁTICA DISCRETA II	MATA42
	OB	FISA75	ELEMENTOS DO ELETROMAGNETISMO E DE CIRCUITOS ELÉTRICOS	MATA95
	OB	MAT236	MÉTODOS ESTATÍSTICOS	MATA42 / MATA95
4ª Sem.	OB	MATA47	LÓGICA PARA COMPUTAÇÃO	MATA97
	OB	MATA49	PROGRAMAÇÃO DE SOFTWARE BÁSICO	MATA40 / MATA48 / MATA57
	OB	MATA50	LINGUAGENS FORMAIS E AUTÔMATOS	MATA42
	OB	FCHC45	METODOLOGIA E EXPRESSÃO TÉCNICO-CIENTÍFICA	MATA40

Figura 3.9 Página de listagem de disciplinas obrigatórias (com pré-requisitos) do curso ciência da computação

The screenshot shows the website of the Universidade Federal da Bahia, specifically the 'Superintendência de Administração Acadêmica'. The main navigation bar includes 'SUPAC', 'CARE - Aluno', 'CPRA - Acadêmico', 'CIEPE - Espaço Físico', and 'PROCESSO SELETIVO'. The 'MENU DO ALUNO' is expanded to show 'Matrícula'. The main content area is titled 'Guia de Matrícula por Curso 2016.2' and lists various courses under 'Área I - Ciências Físicas, Matemática e Tecnologias'. The list includes courses like '101 Arquitetura', '102 Engenharia Civil', '103 Engenharia de Minas', '104 Engenharia Elétrica', '105 Engenharia Mecânica', '106 Engenharia Química', '107 Engenharia Sanitária e Ambiental', '108 Física', '109 Geografia', '110 Geologia', '111 Matemática', '112 Ciência da Computação', '113 Química', '116 Estatística', '118 Geofísica', and '119 Oceanografia'.

Figura 3.10 Listagem de guias de matrícula por curso da área 1 no site da SUPAC

UNIVERSIDADE FEDERAL DA BAHIA - MATRÍCULA 2016-2
CURSO: Ciência da Computação

Disciplina	Turma	Vagas Of.	Dia	Horário	Docente
ADM001 - INTRODUÇÃO A ADMINISTRAÇÃO	030000	11	SEG	20:20 às 22:10	Francisco Bonelli
			QUA	20:20 às 22:10	Francisco Bonelli
BIOC88 - ACCS - CIENCIA, COMUNICAÇÃO E CIDADANIA: Engajamento da soci	010100	2	SEX	10:40 às 12:30	Blandina Felipe Vianna
			CMB	00:00 às 00:00	Blandina Felipe Vianna
FCC001 - CONTABILIDADE GERAL I	010000	10	TER	10:40 às 12:30	Eduardo Tadeu Santana
			QUI	10:40 às 12:30	Eduardo Tadeu Santana
			TER	20:20 às 22:10	Inacilma Rita Silva Andrade
			QUI	20:20 às 22:10	Inacilma Rita Silva Andrade
FCHC45 - METODOLOGIA E EXPRESSÃO TÉCNICO-CIENTÍFICA	010000	40	SEG	14:50 às 16:40	Masilowa Islanowva Cavalcanti Freitas
			QUA	14:50 às 16:40	Masilowa Islanowva Cavalcanti Freitas
FISA75 - ELEMENTOS DO ELETROMAGNETISMO E DE CIRCUITOS ELÉTRICOS	010100	20	SEG	14:50 às 16:40	Docente não informado pelo Departamento
			QUA	14:50 às 16:40	Docente não informado pelo Departamento
			SEX	14:50 às 16:40	Docente não informado pelo Departamento
LETB33 - LINGUA ALEMÁ EM NIVEL BASICO	000100	5	SEG	13:00 às 14:50	Ingrid Maria Santos da Silva
			QUA	13:00 às 14:50	Ingrid Maria Santos da Silva
			SEX	13:00 às 14:50	Ingrid Maria Santos da Silva
LETB34 - LINGUA ALEMÁ EM NIVEL INTERMEDIARIO	000100	5	SEG	13:00 às 14:50	Leila Magalhães Santos Schultz
			QUA	13:00 às 14:50	Leila Magalhães Santos Schultz
			SEX	13:00 às 14:50	Leila Magalhães Santos Schultz
LETB35 - LINGUA ALEMÁ EM NIVEL AVANÇADO	000100	3	SEG	13:00 às 14:50	Zaira Nascimento da Frauca

Figura 3.11 Guia de matrícula do curso Ciência da Computação

cada uma de suas aulas e, por fim, o número de vagas daquela turma oferecidas para o curso ao qual se refere cada guia de matrícula.

Vale notar também que não há uma relação de um para um entre os cursos obtidos na primeira tarefa e os guias de matrícula de curso presentes no site da SUPAC. Isso ocorre principalmente porque enquanto se faz necessário ter diferentes currículos para diferentes modalidades de um mesmo curso (licenciatura e bacharelado) não é necessário criar diferentes guias de matrícula para eles, pois ambos dividem as mesmas turmas e vagas.

Uma *task* adicional também foi criada mas esta não é utilizada para extrair mais informações e sim para padronizar as já extraídas pelas outras tarefas. Chamada de *titleize*, ela padroniza a escrita dos nomes de cursos e disciplinas, convertendo-as para caixa baixa e em seguida capitalizando as palavras utilizando um mesmo conjunto de regras que envolvem, por exemplo, a escrita com letra maiúscula das palavras devidas e a escrita em caixa alta de numerais romanos.

3.5 ARQUITETURA LÓGICA DA APLICAÇÃO

Assim como todas as aplicações feitas utilizando Ruby on Rails, a arquitetura do MeuHorário 2 foi desenvolvida com base no padrão MVC. Esse padrão separa a apresentação e a interação dos dados do sistema, estruturando-o em três componentes lógicos que interagem entre si: *models*, *views* e *controllers*. Enquanto essa separação pode desnecessariamente aumentar a complexidade de aplicações simples, ele garante a independência entre os três componentes, permitindo que eles sejam modificados sem afetar uns aos outros (SOMMERVILLE, 2011).

- Model (Modelo) — Representa os dados da aplicação, contendo as regras de escrita, leitura e validação desses dados;
- View (Visão) — Camada responsável pela interação direta com o usuário. Gera uma representação dos dados presentes nos modelos;
- Controller (Controlador) — Componente que intermedia os *models* e as *views*. As requisições dos usuários chegam nas *controllers*, que deve fazer a comunicação entre os dados dos *models* e a representação das *views*.

As seguintes subseções trazem detalhes sobre a arquitetura do MeuHorário 2. A primeira subseção mostra os detalhes da implementação do banco de dados e as subseções seguintes mostram os *models*, *controllers* e *views* da aplicação.

3.5.1 Banco de Dados

No Ruby on Rails, a comunicação com o banco de dados é feita através de *gems*, que são chamadas de *database adapters*. A mais popular delas é a *pg*¹³, utilizada para o PostgreSQL, um sistema gerenciador de banco de dados relacional (SGBDR) *open-source*.

¹³<https://infinum.co/the-capsized-eight/analyzing-rubygems-stats-v2016>

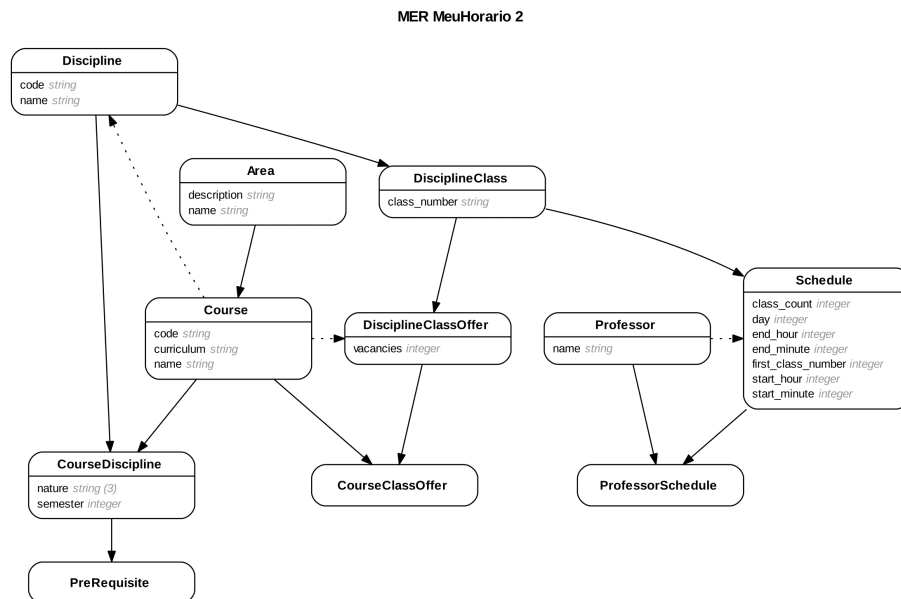


Figura 3.12 Modelo entidade relacionamento do MeuHorário 2

Essa popularidade e os benefícios que ela traz (como uma maior comunidade trocando informações) foram os motivos que levaram à sua escolha como o SGBDR do MeuHorário 2.

O banco de dados da aplicação foi desenvolvido em paralelo ao processo de *web scraping*, ou seja, as entidades e seus relacionamentos foram adicionados de acordo com a criação de novos *wrappers*. Esse desenvolvimento iterativo é possível graças ao uso de *migrations*, blocos de código Ruby que permitem a uma aplicação Ruby on Rails fazer alterações na estrutura do seu banco de dados. O uso dessas *migrations* não só permite que o banco de dados seja facilmente desenvolvido de forma iterativa como também o versionamento do mesmo, garantindo que as *migrations* sejam sempre executadas na ordem correta e fazendo possível replicar a estrutura do banco de dados apenas com a execução delas.

O modelo entidade relacionamento do banco de dados do MeuHorário 2 pode ser visto na Figura 3.12.

A primeira das *migrations* desenvolvidas adicionava ao banco de dados apenas a entidade Curso e dois de seus atributos: nome e código, informações que são encontradas pela primeira *task* de extração de dados, a *task* de cursos. Doze outras *migrations* foram adicionadas posteriormente e juntas geram um banco de dados com um total de onze entidades.

3.5.2 Models

No Rails, em geral, cada *model* criado representa uma entidade do banco de dados e portanto o MeuHorário 2 também possui onze *models*. Como esta aplicação é baseada

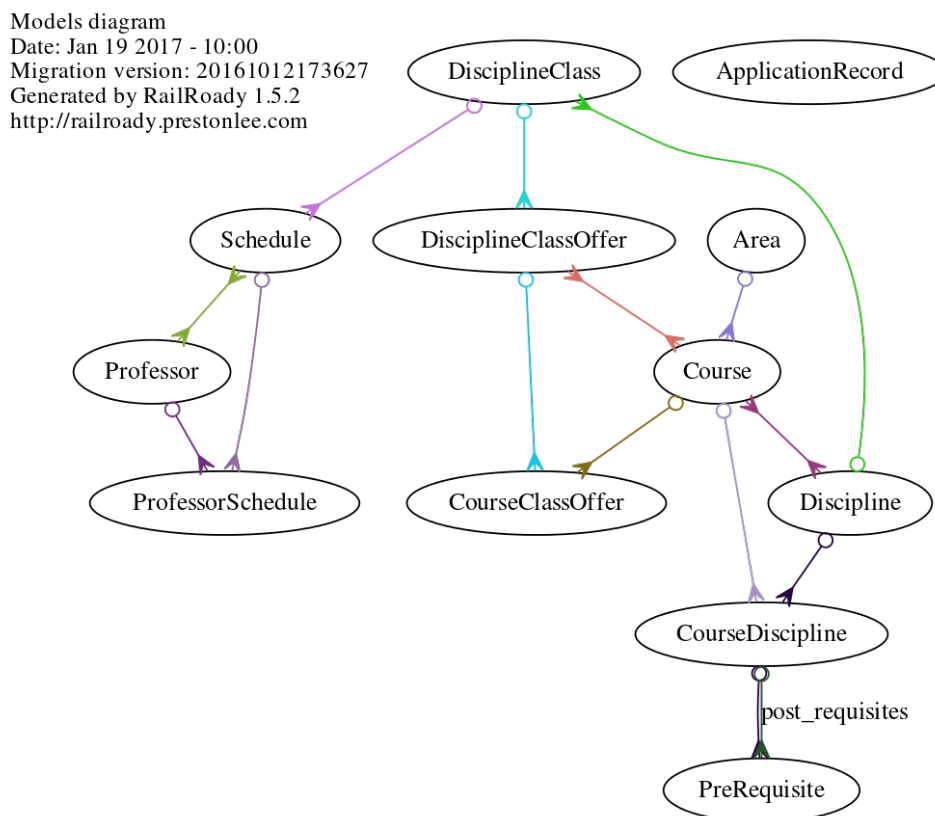


Figura 3.13 Diagrama de *models* da aplicação gerado pela *gem* RailRoady

em grande parte na exibição de uma grande quantidade de dados estruturados um tanto complexos, com entidades possuindo vários relacionamentos apesar da pouca quantidade de entidades no banco de dados, uma boa organização dos modelos se torna essencial para o funcionamento eficiente da aplicação. Os onze modelos e seus relacionamentos estão representados na Figura 3.13.

Para facilitar essa organização, o Rails conta com uma funcionalidade para definição dos relacionamentos dentro dos *models*. Dessa forma é possível acessar os outros modelos ligados por um relacionamento da mesma forma que atributos, abstraindo a camada do banco de dados. Um exemplo disso pode ser visto abaixo:

```

class CourseDiscipline < ApplicationRecord
  ...
  has_many :pre_requisites, foreign_key: :post_discipline_id,
    class_name: "PreRequisite", dependent: :destroy
  ...
end

```

Este código permite que os pré-requisitos de uma disciplina dentro de um currículo sejam acessados por um simples comando: `cd.pre_requisites` (onde `cd` é um objeto da classe

CourseDiscipline). Isso é possível porque ele define que os objetos da classe CourseDiscipline possuem vários *pre_requisites*, objetos pertencentes a classe PreRequisite, através da chave estrangeira *post_discipline_id* e que esses *pre_requisites* devem também ser excluídos do banco de dados caso o objeto da classe CourseDiscipline seja excluído.

3.5.3 Controllers

Nesta aplicação que possui a exibição de informações para o usuário como aspecto importante, as *controllers*, ou controladores, têm como principais responsabilidades a busca pelos modelos que possuem as informações que serão exibidas em cada página e a organização dessas informações.

No total, o MeuHorário 2 possui cinco *controllers*: *admin*, *application*, *areas*, *courses* e *disciplines*. Três delas possuem nomes de *models* da aplicação e suas *actions* (métodos que respondem diretamente às requisições dos usuários) são utilizadas para buscar informações dos seus modelos relacionados. Uma outra (AdminController) permite que um usuário administrador acesse as *tasks* de *web scraping* por meio de requisições HTTP para a aplicação e, por último, a ApplicationController possui *actions* da aplicação que não estão ligadas diretamente a um *model*, como a exportação para PDF.

A Figura 3.14 mostra os cinco *controllers* da aplicação e as suas *actions*.

3.5.4 Views

As *views* são documentos HTML que representam páginas ou partes de páginas web. Junto com códigos CSS (que incrementam o visual dessas páginas) e códigos JavaScript (que permitem a interação dos usuários com os elementos das páginas) eles formam o chamado *front-end* da aplicação.

O MeuHorário 2 possui um total de 15 *views*. A maioria delas representa páginas da aplicação enquanto o restante representa partes de páginas, sendo tanto *layouts* (*views* que servem de embrulho para outras *views*) quanto *partials* (blocos de código normalmente utilizado em várias páginas).

Todas as *views* foram desenvolvidas para serem responsivas, ou seja, adaptar o seu layout para garantir uma boa usabilidade em diferentes dispositivos com telas de diferentes tamanhos. Para isso foi utilizado o já explicado Foundation com suas classes responsivas e *media queries*, que definem diferentes regras CSS para diferentes tamanhos de tela. A Figura 3.15 traz a exibição de fragmentos da página de geração de matrícula como vistos em um smartphone. Versões para desktop desses fragmentos podem ser vistos nas Figuras 3.3 e 3.4.

A funcionalidade de geração de matrícula, incluindo a habilidade de selecionar e desfazer a seleção de turmas, é feita em sua maior parte utilizando código JavaScript e está, portanto, presente nas *views*. A parte principal desse código pode ser encontrada na *view* referente a *action* show de CoursesController.

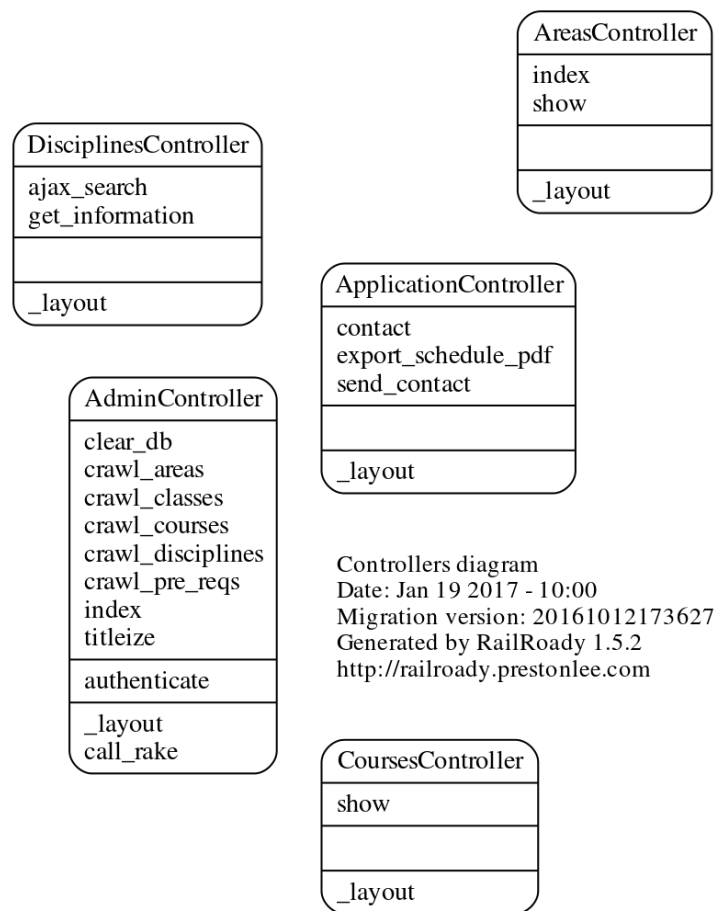


Figura 3.14 Diagrama de controllers da aplicação gerado pela gem RailRoady

meu horário UFBA

Monte a sua grade
Ciência da Computação - Bacharelado - Salvador

Obrigatórias Optativas Ofertadas Todas

1º Semestre

MATA42 Matemática Discreta I	MATA39 Seminários de Introdução ao Curso	MATA38 Projeto de Circuitos Lógicos
MATA37 Introdução à Lógica de Programação	MATA02 Cálculo A	MATA01 Geometria Analítica

2º Semestre

MATA97 Matemática Discreta II	MATA95 Complementos de Cálculo	MATA57 Laboratório de Programação I
MATA48 Arquitetura de	MATA40 Estruturas de Dados	MATA07

	SEG	TER	QUA	QUI	SEX	SAB
07:00						
07:55						
08:50						
09:45						
10:40						
11:35						
13:00		MATA02	MATA02	MATA02		
13:55		MATA02	MATA02	MATA02		
14:50	MATA01	MATA42	MATA01	MATA42		
15:45	MATA01	MATA42	MATA01	MATA42		
16:40		MATA37		MATA37		
17:35		MATA37		MATA37		
18:30						
19:25						
20:20						
21:15						

▲ Topo [Exportar como PDF](#)

Figura 3.15 Disciplinas obrigatórias do curso de ciência da computação (à esquerda) e grade de horários (à direita) do MeuHorário 2 como visto em um smartphone

Este capítulo possui uma análise do resultado final da aplicação a partir de uma avaliação heurística, de sugestões e de estatísticas de uso.

AValiação E RESULTADOS

Neste capítulo serão mostradas uma avaliação heurística, sugestões enviadas por usuários e estatísticas de uso da aplicação, comparando os seus acessos aos do MeuHorário original no semestre anterior. A linha do tempo abaixo servirá de guia para as seções seguintes:

- 31 de outubro de 2016 — Os guias de matrícula foram disponibilizados pela UFBA;
- 1 de novembro de 2016 — O MeuHorário 2 foi disponibilizado ao público;
- 2 de novembro de 2016 — O MeuHorário original passou a redirecionar para o MeuHorário 2;
- 4 de novembro de 2016 — Divulgação da aplicação em redes sociais;
- 8 a 10 de novembro de 2016 — Período de matrícula web;
- 14 de novembro de 2016 — Divulgação pela UFBA dos resultados da matrícula web;
- 16 a 19 de novembro de 2016 — Período de matrícula presencial;
- 21 a 23 de novembro de 2016 — Período de ajuste de matrícula;
- 24 a 28 de novembro de 2016 — Período de ajuste de matrícula exclusivo para alunos de bacharelados interdisciplinares.

Tabela 4.1 As dez heurísticas de Nielsen

Heurística	Descrição
H1	Visibilidade do status do sistema
H2	Correspondência entre o sistema e o mundo real
H3	Controle e liberdade para o usuário
H4	Consistência e padronização
H5	Prevenção de erros
H6	Reconhecimento ao invés de memorização
H7	Flexibilidade e eficiência no uso
H8	Design estético e minimalista
H9	Ajuda para reconhecer, diagnosticar e se recuperar de erros
H10	Ajuda e documentação

O sistema deve sempre manter os usuários informados sobre o que está acontecendo através de feedback rápido e apropriado.

O sistema deve falar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário. Ele deve seguir convenções do mundo real, fazendo informações aparecerem em uma ordem lógica e natural.

Muitas vezes os usuários podem realizar ações por engano. A aplicação deve permitir que os usuários desfaçam e refaçam suas ações.

O sistema deve seguir as convenções existentes, utilizando sempre as mesmas palavras e ícones para as mesmas ações.

Um sistema que evite que erros ocorram é ainda melhor que um sistema com boas mensagens de erro.

O usuário não deve ter que lembrar de informações de outras partes do sistema. Instruções de uso do sistema devem estar sempre visíveis ou serem facilmente encontradas.

A utilização do sistema deve ser agilizada pelo uso de atalhos.

O sistema não deve exibir informações desnecessárias para os usuários pois elas irão competir com as necessárias.

Mensagens de erro devem ser claras, indicar o problema e sugerir uma solução.

Apesar de que o ideal é que o sistema não precise de documentação para ser usado, pode ser necessária a inclusão de diálogos de ajuda e/ou de uma documentação sucinta.

Tabela 4.2 Classificação de severidade

Severidade	Descrição
0	Não concordo que esse seja um problema de usabilidade.
1	Problema cosmético apenas — só deve ser consertado se houver tempo de desenvolvimento sobrando.
2	Pequeno problema de usabilidade — o conserto desse problema deve ter prioridade baixa.
3	Grande problema de usabilidade — importante de ser consertado, deve ter prioridade alta.
4	Catástrofe de usabilidade — é indispensável que seja consertado antes do lançamento do sistema.

4.1 AVALIAÇÃO HEURÍSTICA

Após o seu desenvolvimento, o MeuHorário 2 foi analisado através de uma avaliação heurística, um método rápido e de baixo custo para avaliar a usabilidade de aplicações. Como explicado na Seção 2.3.1.1, este método consiste na inspeção da aplicação por um ou mais especialistas à procura de infrações ao conjunto de heurísticas (princípios reconhecidos de usabilidade) utilizado.

No caso desta avaliação, foi utilizado o conjunto de dez heurísticas proposto por Jakob Nielsen (1994). Essas heurísticas podem ser encontradas na Tabela 4.1. As infrações encontradas foram julgadas a partir da classificação de severidade entre 0 e 4 também proposta por Nielsen (1994). A classificação dos problemas de usabilidade é feita de acordo com a frequência em que eles ocorrem, o seu impacto na utilização da aplicação e a sua persistência, que indica se ele pode ou não ser evitado pelos usuários uma vez que seja conhecido. A Tabela 4.2 mostra os cinco valores da classificação e suas descrições.

A avaliação heurística deve ser tipicamente realizada por um número de especialistas entre três e cinco para maximizar o seu custo-benefício, porém este número não é exato e difere para cada aplicação (Nielsen, 1994). Além disso, a escolha da quantidade de especialistas deve ser feita de acordo com a necessidade da aplicação e o retorno esperado pela correção de problemas de usabilidade (Nielsen, 1994). Por causa do escopo limitado do MeuHorário 2, do curto tempo disponível e da dificuldade em se encontrar outros avaliadores, sua avaliação heurística foi feita apenas pelo seu desenvolvedor.

A avaliação foi realizada em duas etapas, assim como indicado por Nielsen (1994). Na primeira delas foi avaliado o fluxo da aplicação, julgando a usabilidade das funcionalidades gerais do sistema. Já na segunda etapa a avaliação foi repetida mas o foco foi nos elementos individuais da interface do sistema.

No total foram identificados dez problemas, sendo três de severidade 1, quatro de severidade 2 e três de severidade 3. Uma lista com todos os problemas encontrados pode ser vista na Tabela 4.3 junto com as heurísticas afetadas, a sua severidade e uma descrição detalhada do que é o problema e onde foi identificado.

Apesar do número aparentemente elevado de problemas de usabilidade encontrados para uma aplicação com escopo tão limitado, muitos deles são bastante simples de serem

Tabela 4.3 Problemas identificados no MeuHorário 2 pela avaliação heurística

Problema	Heurísticas	Severidade	Descrição
Texto na página inicial	H2 H6	1	O texto “Escolha a sua área” na página inicial não deixa claro que deve ser feita a seleção da área do curso procurado.
Legenda no currículo do curso	H6	1	Não há legenda para indicar o significado das cores quando se passa o mouse sobre uma disciplina na aba de disciplinas obrigatórias de um curso.
Exportação de PDF com conflitos de horário	H5	1	Nenhuma mensagem de aviso é exibida ao exportar PDF com conflitos de horário.
O que fazer na página de seleção de curso	H6	2	Não há indicação do que deve ser feito na página de seleção de curso.
Informação sobre o currículo	H2 H6	2	Não há indicação de qual o período do currículo sendo exibido na página de seleção de turmas.
Resolução de conflitos	H3	2	Para resolver os conflitos de horário é necessário abrir o modal de informações da disciplina e clicar novamente na turma, desfazendo a seleção.
Sugestão enviada?	H1	2	O envio do formulário de sugestão não resulta em nenhuma informação de sucesso ou falha para o usuário.
Clareza das abas	H2 H10	3	Não há texto de ajuda para indicar a diferença entre as abas da página de seleção de turmas. Particularmente “Ofertas” e “Todas” não têm significado óbvio.
Nome das disciplinas escondido	H6	3	Na grade de horários só é possível visualizar o código das disciplinas, o que dificulta o seu rápido reconhecimento.
Página de erro inadequada	H9	3	As páginas de erro da aplicação não trazem informações sobre o erro, como resolvê-lo e nem uma forma de retornar à aplicação.

resolvidos, necessitando apenas de pequenas mudanças visuais e a adição de textos de ajuda e dicas de uso. Somado ao fato de nenhum problema de severidade 4 ter sido encontrado, é possível afirmar que a aplicação está pronta para uso mas se beneficiaria bastante com uma atualização para a correção dos problemas identificados.

4.2 MIXPANEL

A seguir serão mostrados o processo de coleta de dados utilizando o Mixpanel e os resultados encontrados pelo seu uso. Serão apresentados tanto o padrão de uso da aplicação quanto estatísticas sobre os usuários.

4.2.1 Coleta dos dados

Para fazer a coleta dos dados utilizando o Mixpanel foi necessário primeiramente criar uma conta e em seguida um projeto em seu site¹. Devido a baixa quantidade de usuários esperados para o MeuHorário 2, apenas alunos regulares da UFBA, foi possível utilizar o plano gratuito do Mixpanel. A principal limitação desse plano é disponibilização de dados de uma quantidade limitada de dias, portanto os dados a seguir foram coletados durante o período de 1 a 30 de novembro de 2016.

O Mixpanel coleta dados chamados de *tracks*. Esses *tracks* referem-se a eventos definidos pelos desenvolvedores da aplicação analisada e possuem diversas propriedades. Muitas propriedades são coletadas automaticamente, independente de qual evento está sendo rastreado, mas também é possível que os desenvolvedores definam a coleta de propriedades customizadas.

Para realizar o envio dos *tracks* foi necessário adicionar um código JavaScript disponível no site do Mixpanel à aplicação; esse código permite a utilização de funções JavaScript para o envio de *tracks*. Também foi necessário adicionar as chamadas para essas funções, definindo o evento a ser enviado e as suas propriedades customizadas. Sete eventos foram coletados no MeuHorário 2:

- Acessou a home — Rastreado quando um usuário acessa a página inicial da aplicação. Não possui propriedades customizadas;
- Acessou a página de área — Rastreado quando um usuário acessa uma página de área (com listagem dos cursos). Possui apenas uma propriedade customizada: `nome_area`;
- Acessou a página de curso — Rastreado quando um usuário acessa uma página de curso (com a listagem de disciplinas e a grade de horário). Possui apenas uma propriedade customizada: `nome_curso`;
- Buscou por disciplina — Rastreado quando o usuário faz uma busca por disciplina na página de curso. Possui duas propriedades customizadas: `nome_curso` e `query`.

¹<https://mixpanel.com>

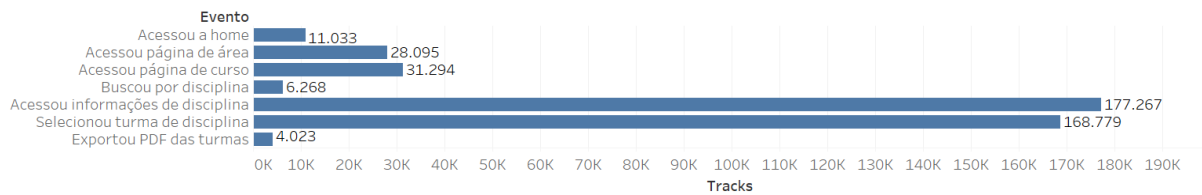


Figura 4.1 Número de tracks registrados no MeuHorário 2

- Acessou informações de disciplina — Rastreado quando um usuário abre o modal de informações de uma disciplina. Possui duas propriedades customizadas: `nome_curso` e `codigo_disciplina`.
- Selecionou turma de disciplina — Rastreado quando um usuário seleciona uma turma de uma disciplina, adicionando seus horários na grade de matrícula. Possui três propriedades customizadas: `nome_curso`, `codigo_disciplina` e `codigo_turma`.
- Exportou PDF das turmas — Rastreado quando um usuário exporta como PDF a grade gerada. Possui uma propriedade customizada estática (`nome_curso`) e propriedades customizadas dinâmicas para cada disciplina presente na grade de horários, com o código da disciplina como chave e a turma selecionada como valor.

4.2.2 Estatísticas de uso

Durante o mês de novembro, período em que os dados aqui apresentados foram coletados, foram enviados 426.759 *tracks* do MeuHorário 2 para o Mixpanel. A distribuição dos eventos desses *tracks* pode ser vista na Figura 4.1.

Segundo os *tracks*, 8.637 usuários distintos acessaram a aplicação durante esse período. Vale notar que esse número não é completamente preciso por causa da forma em que o Mixpanel identifica os usuários. A identificação é feita por meio de um *cookie*, que é criado no navegador do usuário em seu primeiro acesso e não é substituído, permitindo que ele seja identificado posteriormente. No entanto, como *cookies* podem ser facilmente apagados e não há uma relação de um para um entre usuários e navegadores, esse valor certamente não representa a realidade.

A quantidade de usuários distintos por dia se manteve inconstante durante os trinta dias, muitas vezes dobrando ou caindo pela metade de um dia para outro. Isso pode ser explicado pela natureza da aplicação, que deve possuir picos de acesso durante as fases de matrícula e vales durante os períodos de espera de resultados e após o período de matrícula.

Na Figura 4.2 é possível notar o comportamento dos usuários. Os acessos ganham força a partir do redirecionamento do MeuHorário original para a sua versão atualizada e têm seu maior pico no dia de divulgação do MeuHorário 2 nas redes sociais. Houve também o crescimento de acessos no início dos três períodos de matrícula gerais seguidos de períodos de declínio.

A propriedade padrão do Mixpanel de sistema operacional mostra que a maioria dos *tracks* foi feita por computadores de mesa ou laptops, aproximadamente 76% contra 24%

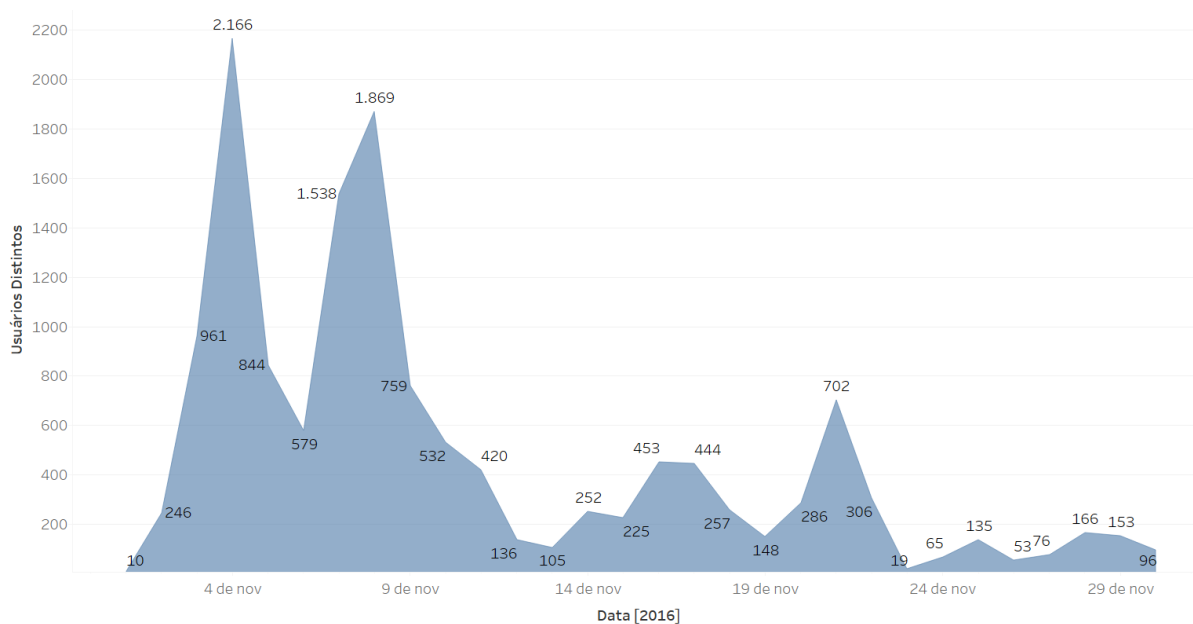


Figura 4.2 Usuários distintos por dia do MeuHorário 2

vindos de dispositivos móveis. O Windows é o sistema operacional campeão de acessos, com mais de quatro vezes o número de *tracks* do segundo colocado, o Android (Figura 4.3).

A Figura 4.4 mostra a quantidade de acessos dos trinta cursos mais populares do MeuHorário 2. As barras amarelas representam o total de acessos às páginas dos cursos enquanto que as barras azuis representam o número de acessos a essas páginas feitos por usuários distintos.

4.3 SUGESTÕES

A aplicação também possui um formulário para envio de sugestões. As sugestões contêm uma mensagem e, opcionalmente, o nome do sugeridor e um email para contato. Essas informações são então enviadas por email para meuhorarioufba@gmail.com.

No total foram enviadas 16 sugestões. Dessas, seis continham elogios ou parabenizações, quatro continham sugestões de usabilidade, três continham sugestões de adição de outras informações e três continham sugestões de funcionalidades. Para garantir a anonimidade dos usuários serão mostradas apenas as iniciais dos seus nomes.

“O serviço do site é incrível! Muito obrigado!” - G.

Uma das sugestões de usabilidade indicou alguma dificuldade em desfazer a seleção de turmas, afirmando que não tinha ficado claro o que deveria fazer. As outras sugeriam novas interações com a aplicação, como mostrar o horário de início e término da turma ao passar o mouse sobre uma de suas aulas na grade de horários.

“acho que deveria ficar mais visível como tirar a materia , para colocar ficou claro que so clicando e tal mais ai vai fazendo bateu ate uma duvida como

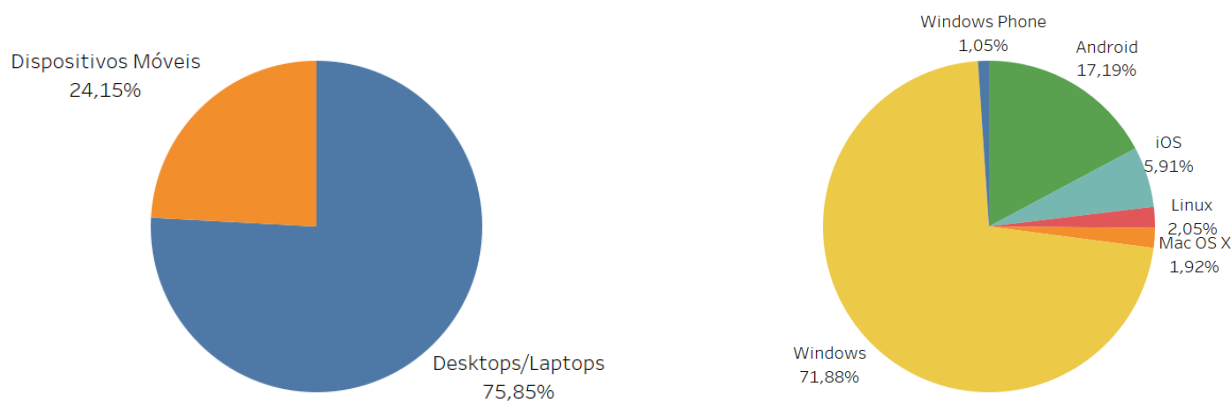


Figura 4.3 Dispositivo (à esquerda) e sistema operacional (à direita) de origem dos *tracks* do MeuHorário 2

remover a materia , dps me toquei que tinha que clicar dnv , não ficou bem claro isso para mim” - L.

Nas sugestões de adição de informações foram pedidas as adições da ementa e da carga horária das disciplinas, além do local das aulas de suas turmas.

“[...] Adicionar a quantidade de horas de cada matéria. [...]” - R.

Duas das três sugestões de funcionalidades pediam por uma forma de salvar a grade de horários construída para que fosse possível continuar com a sua montagem posteriormente.

“[...] Como sugestão dou a ideia da grade continuar com as materias q vc colocou no curso anterior. [...]” - M.

4.4 COMPARAÇÃO DE ACESSOS

Nesta seção serão feitas comparações entre os acessos ao MeuHorário original e ao MeuHorário 2. Para isso são utilizados os dados do Mixpanel apresentados na Seção 4.2 referentes a versão atualizada da aplicação e dados do Mixpanel referentes a sua versão anterior coletados durante a matrícula do semestre 2016.1. A seguir será mostrada uma linha do tempo do semestre 2016.1 para melhor entendimento das comparações:

- 19 de maio de 2016 — Os guias de matrícula foram disponibilizados pela UFBA;

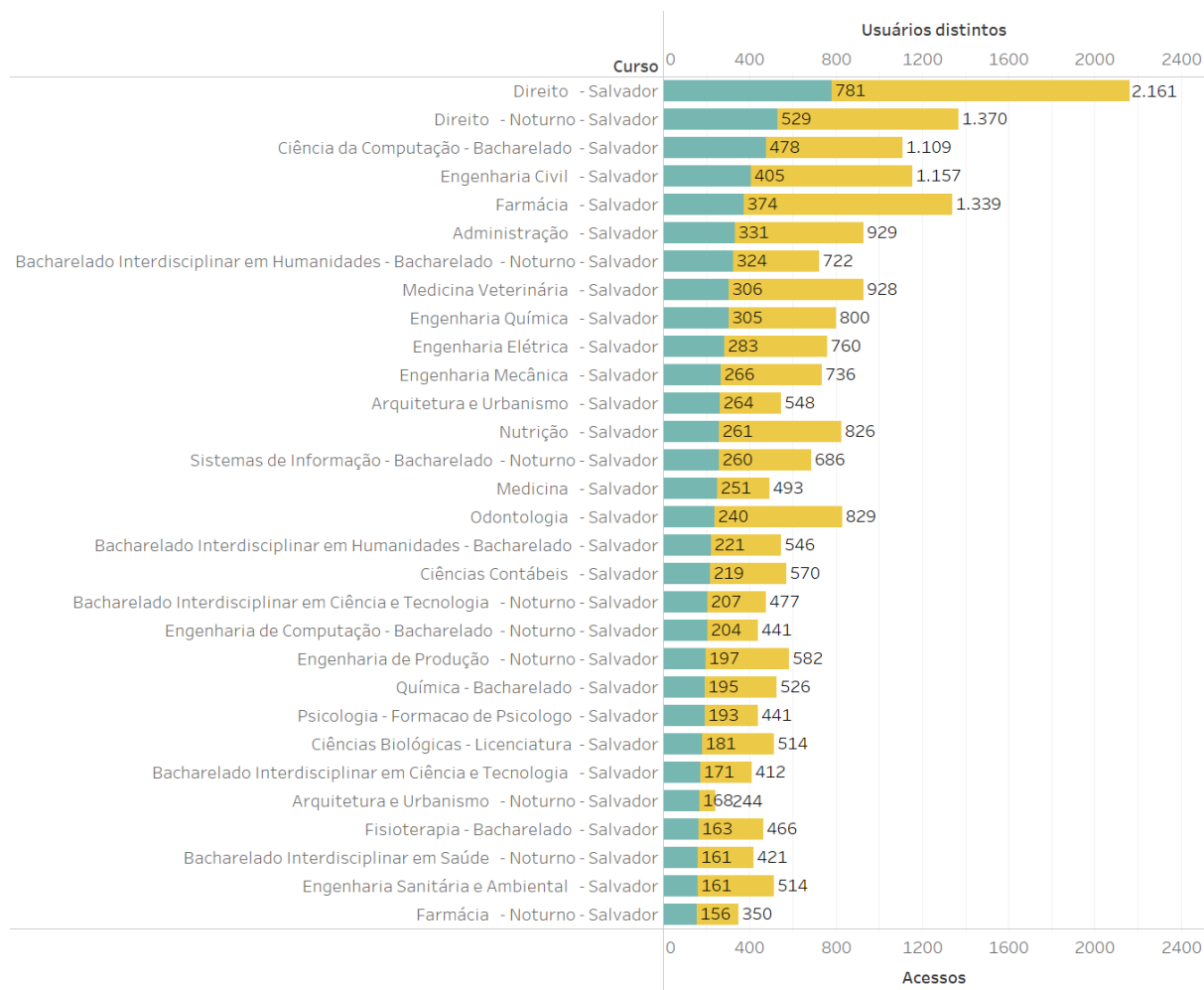


Figura 4.4 De verde o número de acessos feitos por usuários distintos à página do curso e de amarelo o número de acessos totais à página do curso do MeuHorário 2

- 26 de maio de 2016 — Atualização dos guias de matrícula do MeuHorário original para o semestre 2016.1;
- 3 de junho 2016 — Início da coleta de estatísticas de uso do MeuHorário original através do Mixpanel;
- 9 a 13 de junho de 2016 — Período de matrícula web;
- 15 de junho de 2016 — Divulgação pela UFBA dos resultados da matrícula web;
- 16 a 22 de junho de 2016 — Período de matrícula presencial;
- 27 a 29 de junho de 2016 — Período de ajuste de matrícula;
- 30 de junho a 9 de julho de 2016 — Período de ajuste de matrícula exclusivo para alunos de bacharelados interdisciplinares.

Infelizmente não será possível comparar os acessos das aplicações durante o período imediatamente posterior à disponibilização dos guias de matrícula, visto que a utilização do Mixpanel pelo MeuHorário original começou muito após essa disponibilização. Inicialmente será feita uma comparação geral dos acessos, em seguida uma comparação de acessos no período de matrícula web e por fim comparações de dispositivos e sistemas operacionais.

A Figura 4.5 mostra a quantidade de usuários distintos por dia no MeuHorário original. Em uma comparação com a Figura 4.2 é possível ver que os acessos aumentaram bastante. Com um total de 3.679 usuários distintos quando considerado um período de 30 dias (de 3 de junho a 2 de julho) igual ao do MeuHorário 2, nota-se um aumento de aproximadamente 135% no número de usuários. No entanto, é impossível afirmar que esse aumento se deu apenas pela reformulação do site, pois a coleta não foi feita desde o início da atualização dos guias de matrícula do MeuHorário original e também não houve divulgação em igual quantidade nas redes sociais.

Uma comparação mais correta é uma entre os acessos vindos de usuários distintos nas aplicações durante os períodos de matrícula web, períodos de matrícula que tiveram maior quantidade de usuários nas duas aplicações. A Figura 4.6 mostra que o MeuHorário 2 teve um índice maior de usuários durante todos os dias de matrícula web em comparação ao MeuHorário original e que mesmo a soma de usuários distintos dos cinco dias da matrícula web de 2016.1 é inferior a quantidade de usuários distintos do primeiro dia da matrícula web de 2016.2.

Outra comparação importante é a de acessos vindos de dispositivos móveis, visto que um requisito importante da criação do MeuHorário 2 era tornar a aplicação amigável a eles. Ao comparar o gráfico da Figura 4.7, que mostra de qual dispositivo foram enviados os *tracks* do MeuHorário original, com o da Figura 4.3, é possível ver um pequeno aumento no número de acessos vindos de dispositivos móveis e que esse aumento foi consistente para todos os sistemas operacionais desses deles (Windows Phone, Android e iOS).

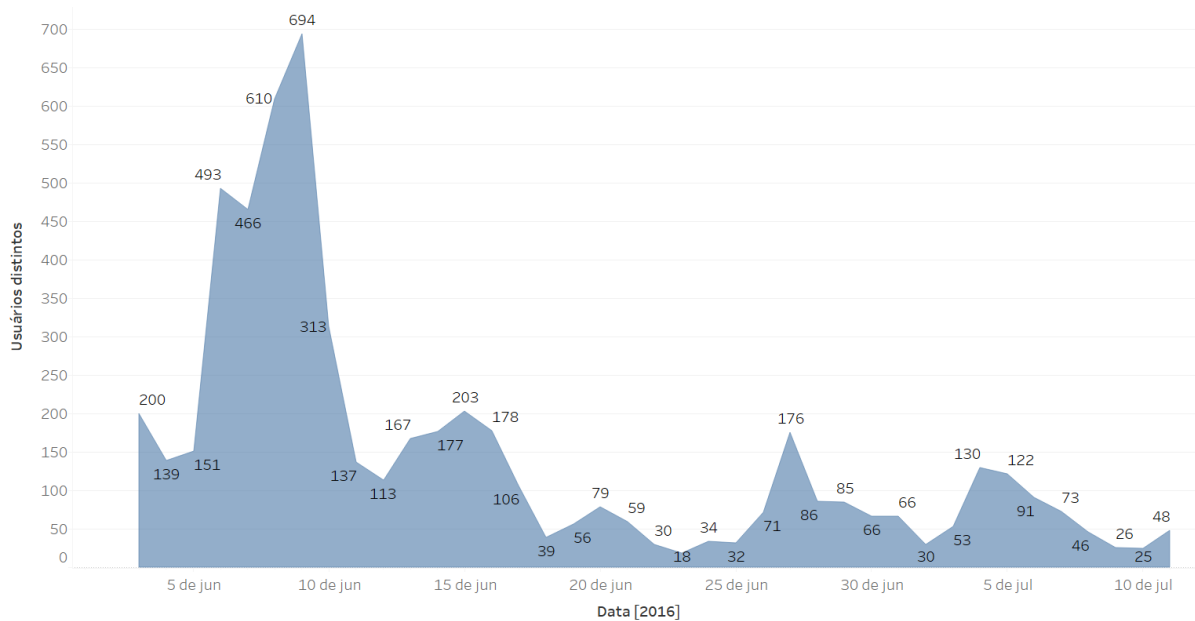


Figura 4.5 Usuários distintos por dia do MeuHorário original

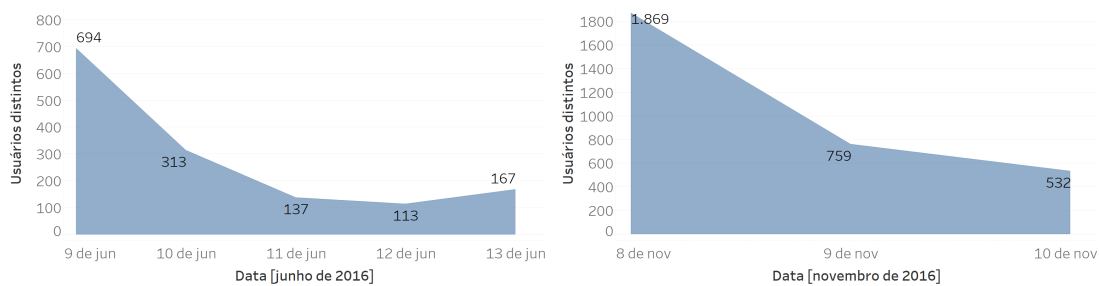


Figura 4.6 Usuários distintos por dia do MeuHorário original (à esquerda) e do MeuHorário 2 (à direita) durante os períodos de matrícula web

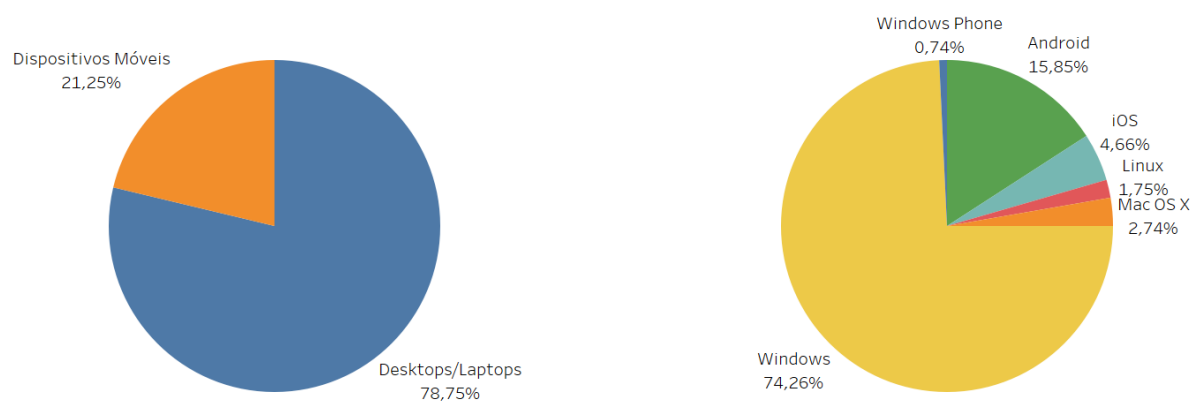


Figura 4.7 Dispositivo (à esquerda) e sistema operacional (à direita) de origem dos *tracks* do MeuHorário 2

Oferece uma recapitulação do trabalho, destacando suas principais contribuições e propõe trabalhos futuros.

CONCLUSÃO

Este trabalho apresentou primeiramente o processo de matrícula e os sites acadêmicos da Universidade Federal da Bahia, em seguida foram mostrados os motivos que levaram à perda de relevância do MeuHorário, uma aplicação criada em 2004 para remediar esses problemas da UFBA.

As limitações da universidade e da aplicação criada para auxiliar o processo de matrícula dos seus alunos mostraram a necessidade do desenvolvimento de uma nova aplicação com objetivos similares porém que utilizasse tecnologias contemporâneas e seguisse tendências atuais. A partir dessa necessidade foi desenvolvido o MeuHorário 2, uma aplicação web para simulação de matrículas à qual refere-se este trabalho.

Para garantir que a aplicação realizasse seus objetivos fez-se necessário um estudo aprofundado dos procedimentos da UFBA, complementando as informações já conhecidas empiricamente. Também foi feito um estudo das técnicas de extração de dados da web, parte essencial da aplicação, e de métodos de avaliação de aplicações web, permitindo comprovar a eficiência e adequabilidade do MeuHorário 2.

O processo de desenvolvimento englobou a definição de requisitos, a escolha de uma linguagem de programação e ferramentas, a modelagem de um banco de dados e a implementação de *crawlers*, *wrappers* e da própria aplicação. O resultado final desse processo foi uma aplicação de uso simples e fácil, com uma execução dividida em etapas bastante definidas.

Essa simplicidade e facilidade de uso foi avaliada por três diferentes métodos, cada um com abordagens diferentes. Uma avaliação heurística, embora um tanto limitada pela quantidade de avaliadores, mostrou que a aplicação não possui nenhum problema catastrófico de usabilidade apesar de ter identificado alguns problemas menos graves. Já as sugestões dos usuários enviadas pela aplicação mostraram que muitos usuários estão contentes com a nova versão e que alguns outros gostariam de melhorias e novas funcionalidades, enquanto que as estatísticas de uso coletados pelo Mixpanel mostraram

um grande aumento na quantidade de usuários da aplicação e na porcentagem de acessos vindos de dispositivos móveis em comparação ao MeuHorário original.

É possível destacar deste trabalho as quatro principais contribuições a seguir:

- Levantamento de aplicações web de simulação de matrícula em universidades brasileiras — Esta monografia mostrou que dificuldades na matrícula não são exclusividade da Universidade Federal da Bahia. A lista de trabalhos relacionados apresentados na Seção 2.4 mostrou que várias aplicações similares foram desenvolvidas, muitas vezes de forma independente, em outras universidades brasileiras para remediar problemas parecidos com os enfrentados pelos estudantes da UFBA;
- O MeuHorário 2 — A aplicação desenvolvida é provavelmente a maior contribuição deste trabalho. Ela já está disponível para uso, foi bastante acessada e poderá ajudar os estudantes da UFBA nos semestres por vir;
- Avaliação abrangente da aplicação — A utilização em conjunto de três métodos de avaliação de abordagens diferentes (avaliação heurística, formulário de sugestões e Mixpanel) resultam em uma avaliação robusta do MeuHorário 2, indicando várias possíveis melhorias a serem feitas;
- Análise das estatísticas de acesso — O Capítulo 4 mostrou que o MeuHorário original e, principalmente, o MeuHorário 2 tiveram grande quantidade de acessos, sendo muitos deles vindos de dispositivos móveis. Esses números podem mostrar para a UFBA a defasagem e necessidade de atualização de alguns de seus sites acadêmicos, que precisam ser melhorados e otimizados para acessos vindos de smartphones e tablets.

Embora a aplicação tenha se mostrado adequada e já esteja sendo utilizada, existem várias possibilidades de trabalhos a serem futuramente realizados:

- Corrigir problemas encontrados — A avaliação heurística em conjunto com as sugestões mostraram que a aplicação ainda possui muitos aspectos que podem ser futuramente melhorados;
- Incluir novas informações — Várias outras informações da UFBA podem ser acrescentadas ao MeuHorário, como ementas, diferentes currículos para um mesmo curso, cursos de pós-graduação, locais das aulas, etc.;
- Adição de perfis dos usuários — Uma vez que a aplicação permita o cadastro e login dos usuários, será possível acompanhar a matrícula dos estudantes em cada semestre, permitindo a coleta de dados sobre quais disciplinas foram cursadas em quais semestres, porcentagem de aprovação de cada turma e até mesmo as notas recebidas;
- Aplicativo para dispositivos móveis — A criação de um aplicativo para dispositivos móveis pode trazer ao MeuHorário funcionalidades não limitadas pelo período de

matrícula. Uma opção seria a disponibilização de grupos de discussão para cada turma aos alunos matriculados nela. Esses grupos permitiriam, por exemplo, a discussão e troca de materiais (como livros e artigos) entre os estudantes da turma ao longo do semestre letivo.

Esses são apenas alguns dos possíveis trabalhos futuros para a aplicação, que tem um alto potencial de expansão para suprir outras necessidades dos estudantes da UFBA. Independente de quais trabalhos sejam desenvolvidos futuramente, o histórico do MeuHorário original e dos trabalhos similares mostra a importância de atualizações para essas aplicações e, em um ambiente universitário com alta rotatividade de alunos, é imprescindível que novos estudantes apareçam para contribuir e manter o projeto MeuHorário relevante.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANTTONEN, M. et al. Transforming the web into a real application platform: new technologies, emerging trends and missing pieces. In: ACM. *Proceedings of the 2011 ACM Symposium on Applied Computing*. [S.l.], 2011. p. 800–807.
- ARCHER, J. Web page design: Heuristic evaluation vs. user testing. 2010.
- ATKINSON, E. *Web analytics and think aloud studies in web evaluation: understanding user experience*. Tese (Doutorado) — London University, 2007.
- BAKER, K.; GREENBERG, S.; GUTWIN, C. Empirical development of a heuristic evaluation methodology for shared workspace groupware. In: ACM. *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. [S.l.], 2002. p. 96–105.
- BARUA, A.; THOMAS, S. W.; HASSAN, A. E. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, Springer, v. 19, n. 3, p. 619–654, 2014.
- BAUMGARTNER, R.; GATTERBAUER, W.; GOTTLÖB, G. Web data extraction system. In: LIU, L.; ÖZSU, M. T. (Ed.). *Encyclopedia of Database Systems*. [S.l.]: Springer US, 2009. p. 3465–3471.
- BAUMGARTNER, R.; LEDERMIILLER, G. Deepweb navigation in web data extraction. In: IEEE. *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*. [S.l.], 2005. v. 2, p. 698–703.
- BERGMAN, M. K. White paper: the deep web: surfacing hidden value. *Journal of electronic publishing*, Michigan Publishing, University of Michigan Library, v. 7, n. 1, 2001.
- BLACKMON, M. H. et al. Cognitive walkthrough for the web. In: ACM. *Proceedings of the SIGCHI conference on human factors in computing systems*. [S.l.], 2002. p. 463–470.
- BOSCH, A.; BOGERS, T.; KUNDER, M. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics*, Springer, v. 107, n. 2, p. 839–856, 2016.
- BRAJNIK, G. Using automatic tools in accessibility and usability assurance processes. In: SPRINGER. *ERCIM Workshop on User Interfaces for All*. [S.l.], 2004. p. 219–234.

- CHAKRABARTI, S.; BERG, M. Van den; DOM, B. Focused crawling: a new approach to topic-specific web resource discovery. *Computer networks*, Elsevier, v. 31, n. 11, p. 1623–1640, 1999.
- CISNEIROS, V. C. de A.; RAMOS, H. Comparação entre servidores web apache http server e nginx. 2015.
- COMEAUX, D. J. Web design trends in academic libraries—a longitudinal study. *Journal of Web Librarianship*, Taylor & Francis, p. 1–15, 2016.
- FERNANDEZ, A.; INSFRAN, E.; ABRAHÃO, S. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, Elsevier, v. 53, n. 8, p. 789–817, 2011.
- FERRARA, E. et al. Web data extraction, applications and techniques: A survey. *Knowledge-Based Systems*, Elsevier, v. 70, p. 301–323, 2014.
- HERMAWATI, S.; LAWSON, G. Establishing usability heuristics for heuristics evaluation in a specific domain: Is there a consensus? *Applied ergonomics*, Elsevier, v. 56, p. 34–51, 2016.
- IBGE (Ed.). *Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal : 2014*. [S.l.: s.n.], 2016.
- LAENDER, A. H. et al. A brief survey of web data extraction tools. *ACM Sigmod Record*, ACM, v. 31, n. 2, p. 84–93, 2002.
- LÁRUSDÓTTIR, M. K. *Listen to your users: The effect of usability evaluation on software development practice*. Tese (Doutorado) — Uppsala universitet, 2009.
- LIU, B. *Web data mining: exploring hyperlinks, contents, and usage data*. [S.l.]: Springer Science & Business Media, 2007.
- MOYERS, B. D. *A world of ideas: Conversations with thoughtful men and women about American life today and the ideas shaping our future*. [S.l.]: Doubleday Books, 1989.
- NIELSEN, J. *Heuristic Evaluation*. Nielsen and Mack (eds.), *Usability Inspection Methods*. [S.l.]: John Wiley & Sons, New York, NY, 1994.
- PAGE, L. et al. *The PageRank citation ranking: Bringing order to the web*. [S.l.], 1998.
- PRAKASH, P.; BIJU, R.; KAMATH, M. Performance analysis of process driven and event driven web servers. In: IEEE. *Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on*. [S.l.], 2015. p. 1–7.
- SCOWEN, G.; REGENBRECHT, H. Increased popularity through compliance with usability guidelines in e-learning web sites. *International Journal of Information Technology and Web Engineering (IJITWE)*, IGI Global, v. 4, n. 3, p. 38–57, 2009.

SOMMERVILLE, I. *Software engineering*. [S.l.]: Pearson, 2011.

Universidade Federal da Bahia. *Regulamento de Ensino de Graduação e Pós-Graduação*. 2015. 41 p. Disponível em: <https://www.ufba.br/sites/devportal.ufba.br/files/REGPG_Completo_Revisado_em_16-04-2015.pdf>.

VIKMAA, A. *Web Data Extraction For Content Aggregation From E-Commerce Websites*. Dissertação (Mestrado) — University of Tartu, 2016.

W3C. *XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)*. <https://www.w3.org/TR/xhtml1/>, 2000.

WEB ANALYTICS ASSOCIATION. *Web Analytics Definitions*. [S.l.], 2008.

WENINGER, T. et al. Web content extraction: a metaanalysis of its past and thoughts on its future. *ACM SIGKDD Explorations Newsletter*, ACM, v. 17, n. 2, p. 17–23, 2016.

WINCKLER, M.; PIMENTA, M. S. Avaliação de usabilidade de sites web. *Escola de Informática da SBC SUL (ERI 2002) ed. Porto Alegre: Sociedade Brasileira de Computação (SBC)*, v. 1, p. 85–137, 2002.

ZAHARAN, D. I. et al. A comparative approach to web evaluation and website evaluation methods. *International Journal of Public Information Systems*, v. 10, n. 1, 2014.