UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

BRUNA DE BARROS XAVIER SANTOS

OTIMIZAÇÃO AUTOMÁTICA DO *DESIGN* DE PÁGINAS *WEB* COM BASE NA ANÁLISE DAS INTERAÇÕES DE USUÁRIOS

Rio de Janeiro
2015

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI DE APLICAÇÕES E PESQUISAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

BRUNA DE BARROS XAVIER SANTOS

OTIMIZAÇÃO AUTOMÁTICA DO *DESIGN* DE PÁGINAS *WEB* COM BASE NA ANÁLISE DAS INTERAÇÕES DE USUÁRIOS

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Instituo Tércio Pacitti da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Orientador: Rodrigo Penteado Ribeiro de Toledo

Co-orientador: Adriano Joaquim de Oliveira Cruz

Rio de Janeiro
2015

Santos, Bruna de Barros Xavier

S237o Otimização automática do *design* de páginas *web* com base na análise das interações de usuários / Bruna de Barros Xavier Santos. – Rio de Janeiro, 2015.

77 f.

Orientador: Rodrigo Penteado Ribeiro de Toledo.

Co-orientador: Adriano Joaquim de Oliveira Cruz.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Programa de Pós-Graduação em Informática, 2015.

1. Otimização de website. 2. Algoritmos genéticos. 3. Web Analytics. 4. Experimentos Controlados. – Teses. I. Toledo, Rodrigo Penteado Ribeiro de orient. II. Cruz, Adriano Joaquim de Oliveira coorient. III. Título

BRUNA DE BARROS XAVIER SANTOS

Otimização automática do *design* de páginas *web* com base na análise das interações de usuários

Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática do Instituto de Matemática e do Instituo Tércio Pacitti da Universidade Federal do Rio de Janeiro - UFRJ, como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Aprovado em: Rio de Janeiro, _____ de _____ de _____.

_____

Prof. Docteur Rodrigo Penteado Ribeiro de Toledo (Orientador)

_____

Prof. Ph.D. Adriano Joaquim de Oliveira Cruz (Co-orientador)

_____

Prof. Doutor Claudio Miceli de Farias

_____

Prof. Ph.D. Felipe Maia Galvão França

_____

Prof. Docteur Josefino Cabral Melo Lima

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer a minha família e também ao meu companheiro, Carlos Felippe Cardoso, por todo amor, apoio, incentivo e paciência ao longo destes anos.

Também agradeço aos meus orientadores, Adriano Cruz e Rodrigo de Toledo, pela oportunidade e incentivos dados para a elaboração deste trabalho; à equipe do SiGA (Sistema Integrado de Gestão Acadêmica), em especial ao Ricardo Storino e ao Raphael Paiva, que não só me deram apoio como me permitiram aplicar as ideias desenvolvidas neste trabalho; aos que me obrigaram a reescrever este trabalho tantas vezes para que eu aprendesse como ele ficava melhor a cada vez.

A todos que direta ou indiretamente estiveram ao meu lado e que de alguma forma doaram um pouco de si para que a conclusão deste trabalho se tornasse possível, o meu muito obrigado.

RESUMO

SANTOS, Bruna de Barros Xavier. **Otimização automática do *design* de páginas *web* com base na análise das interações de usuários**. 2015. 77 f. Dissertação (Mestrado em Informática) - PPGI, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

A evolução das tecnologias web permite criar uma infinidade de interfaces para páginas *web*. Diversas são as possibilidades de combinações dos elementos e suas propriedades de forma que a escolha do *design* é uma tarefa difícil e que muitas vezes, com a influência de opiniões pessoais, pode não ser o indicado para atender aos objetivos do negócio.

Dada a complexidade de criação e decisão de *design*, este trabalho pretende validar se é possível realizar a otimização, de forma não assistida, de interfaces *web* em tempo real. Para isso, este trabalho: i) desenvolve uma ferramenta que gera diferentes versões de uma interface; coleta e analisa dados do comportamento dos usuários em tempo real e evolui os *designs* para encontrar o melhor, ou seja, o que mais de aproxima dos objetivos do negócio; ii) aplica a ferramenta a sistemas reais; iii) analisa os resultados obtidos .

Palavras-chave: Otimização de website, Algoritmos genéticos, Web Analytics, Experimentos Controlados.

ABSTRACT

SANTOS, Bruna de Barros Xavier. **Otimização automática do *design* de páginas *web* com base na análise das interações de usuários**. 2015. 77 f. Dissertação (Mestrado em Informática) - PPGI, Instituto de Matemática, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015.

The evolution of web technologies allows us to create unlimited designs for web graphical user interfaces. Finding a design is not an easy task, since combinations of web elements and changes in their style generate multiple possibilities. We intend to validate the possibility of optimizing (without human intervention) designs of web pages in real-time through: i) the development of a tool that generates different versions of web pages, collects and analyzes data about user behavior in real-time and improves those versions so that the design that is closest to the business goal can be reached; ii) the usage of this tool in real web systems; iii) analysis of results.

Keywords: design, website optimization, genetic algorithm, web analytics, controlled experiments.

LISTA DE FIGURAS

Figura 1.1: Exemplo de algumas das muitas visualizações possíveis de um botão 13
Figura 1.2: Versão original para o *website* da campanha presidencial de Barack Obama em 2007. Disponível em:[34] 14
Figura 1.3: Versão final para o *website* da campanha presidencial de Barack Obama em 2007. Disponível em:[34] 14

Figura 2.1: Criação de uma nova geração. Disponível em: [16] 19
Figura 2.2: One-point crossover 21
Figura 2.3: Two-point crossover 22
Figura 2.4: Mutação 22
Figura 2.5: Relatórios da Piwik. Disponível em: [30] 26
Figura 2.6: Relatórios do Google Analytics. Disponível em: [9] 26
Figura 2.7: Exemplo de teste A/B 28
Figura 2.8: Exemplo de teste multivariável 28

Figura 3.1: Resultado final encontrado pelo algoritmo genético no trabalho de Peñalver e Merelo. Disponível em: [29] 31
Figura 3.2: Apresentação dos *layouts* de uma geração para seleção. Disponível em: [28] 32
Figura 3.3: Estrutura do funcionamento do sistema de *design* de *website*. Disponível em: [44] 33

Figura 4.1: Visão geral do funcionamento de UIDesign 36
Figura 4.2: Alicerces de UIDesign 37
Figura 4.3: Etapas de UIDesign 40
Figura 4.4: Passos da etapa de configuração 41
Figura 4.5: Primeiro passo da etapa de configuração 42
Figura 4.6: Segundo passo da etapa de configuração 43
Figura 4.7: Terceiro passo da etapa de configuração 45
Figura 4.8: Etapas da fase de experimentação 46
Figura 4.9: Indivíduos do algoritmo genético 47
Figura 4.10: Indivíduo 1010 49
Figura 4.11: Resposta da requisição em uma aplicação *web* 49
Figura 4.12: Resposta da requisição em uma aplicação *web* com UIDesign . . . 50
Figura 4.13: Diagrama de sequência 51
Figura 4.14: Resposta das requisições em uma aplicação *web* com UIDesign . . 52
Figura 4.15: Coleta informações sobre o comportamento dos usuário no *website* 53

Figura 5.1: *Design* original da emissão de documentos 56
Figura 5.2: *Design* representado pelo indivíduo 100011 60
Figura 5.3: *Design* representado pelo indivíduo 001100 60
Figura 5.4: Indivíduos com o menor tempo médio encontrados em cada uma das gerações 61
Figura 5.5: Indivíduo 000000 e a variação da métrica 62
Figura 5.6: Indivíduos com o menor tempo médio encontrados em cada uma das gerações, com destaque para os das últimas gerações 63
Figura 5.7: *Design* representado pelo indivíduo 000100 64
Figura 5.8: Comparativo entre os melhores indivíduos e o *design* original . . . 65

LISTA DE TABELAS

Tabela 2.1: Tipos de métricas. Disponível em: [3] 24
Tabela 2.2: Indicadores-chave de *performance* de acordo com o tipo do *website*. Disponível em: [3] 25

Tabela 4.1: Indivíduos e classes de estilo do botão 48
Tabela 4.2: Indivíduos e classes de estilo do cabeçalho 48

Tabela 5.1: Indivíduos da primeira geração e o tempo médio dos usuários para selecionar um documento para impressão 59
Tabela 5.2: Indivíduos 001101 e 111000 e seus respectivos valores para o tempo médio (em segundos) nas primeiras cinco gerações 63

LISTA DE ABREVIATURAS E SIGLAS

AGI	Algoritmos Genéticos Interativos
DAA	Digits Analytics Association
GPL	General Public License
HTML	HyperText Markup Language
KPI	Key Performance Indicator
MVT	Multivariable Testing
OEC	Overall Evaluation Criterion
UFRJ	Universidade Federal do Rio de Janeiro

SUMÁRIO

1 INTRODUÇÃO 12
1.1 HIPÓTESE 15
1.1.1 Estratégia 15
1.2 CONTRIBUIÇÕES 16
1.3 ORGANIZAÇÃO DO TRABALHO 17

2 CONCEITOS BÁSICOS 18
2.1 ALGORITMO GENÉTICO 18
2.1.1 Codificação da solução em indivíduo 19
2.1.2 Avaliação da adaptabilidade do indivíduo 20
2.1.3 Método de seleção 20
2.1.4 Operador de cruzamento (*crossover*) 20
2.1.5 Operador de mutação 21
2.2 *WEB ANALYTICS* 22
2.2.1 Métricas 23
2.2.2 Ferramentas de *web analytics* 23
2.3 EXPERIMENTOS CONTROLADOS 27
2.3.1 Teste A/B 27
2.3.2 Teste Multivariável 28

3 TRABALHOS RELACIONADOS 30

4 FERRAMENTA DE EXPERIMENTAÇÃO 35
4.1 O QUE É UIDESIGN? 35
4.2 TÉCNICAS 37
4.3 ARQUITETURA 39
4.3.1 Etapa de configuração 40
4.3.2 Etapa de experimentação 46

5 EXPERIMENTO 54
5.1 PORTAL ALUNO 54
5.1.1 Metas do negócio e métricas de interesse 55
5.2 UIDESIGN E O PORTAL ALUNO 56
5.2.1 Etapa de configuração 57
5.2.2 Etapa de experimentação 58
5.3 RESULTADOS 61

6 CONCLUSÃO E TRABALHOS FUTUROS 66
6.1 CONCLUSÃO 66
6.2 TRABALHOS FUTUROS 68

REFERÊNCIAS 70

APÊNDICE A CLASSES DE ESTILO 76

1 INTRODUÇÃO

Avanços nas tecnologias *web* permitem a criação de interfaces gráficas *web* bastante sofisticadas para atender a usuários cada vez mais exigentes. O HTML (*HyperText Markup Language*) [38] descreve essas interfaces através de elementos (como textos, imagens e botões) [39], cujas propriedades (como cor, tamanho, posição, fonte etc) podem ser modificadas [40, 41]. Alterações no estilo do elemento podem modificar completamente sua aparência como visto na Figura 1.1.

Combinar elementos e modificar suas propriedades leva a uma gama de possibilidades para a construção da interface de uma página *web*. Como escolher a melhor combinação?

Existem algumas maneiras de se escolher um *design*: através da intuição sobre o que é um bom *design* ou por meio da análise de dados. A sensibilidade humana provê informações qualitativas que podem não refletir corretamente as necessidades dos usuários, o que pode levar a interpretações errôneas sobre sua relação com o sistema [23]. Por outro lado, a coleta e a análise de dados das interações entre usuários e sistema podem ajudar a melhorar a experiência do usuário e levar a decisões de sucesso para o negócio [23, 21, 34]. Ferramentas como Google Analytics [10] e Piwik [31] permitem extrair esse tipo informação.

Artigos e livros mostram exemplos de sistemas reais que obtiveram sucesso através da análise de dados de seus usuários [23, 21, 34]. Um deles, é o *website* criado para a campanha presidencial de Barack Obama em 2007 com o objetivo de angariar doadores [34]. Na interface original (Figura 1.2), usuários se registravam por meio de um botão "Sign Up" e a partir de então poderiam contribuir com doações. No

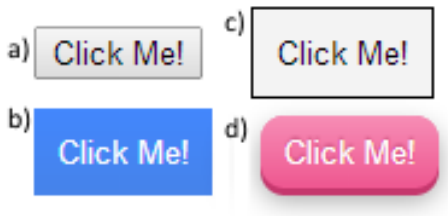

Figura 1.1: Exemplo de algumas das muitas visualizações possíveis de um botão. Da esquerda para a direita, de cima para baixo: a) botão original; botão com borda e cor de fundo modificadas; b) botão sem borda, com cores de fundo e de fonte modificadas; c) botão com borda, cores de fundo e de fonte modificadas

entanto, não era possível garantir que o *design* escolhido seria capaz de atrair o maior número possível de cliques no botão.

A equipe decidiu experimentar novos *designs*, alterando a imagem central e o texto do botão. Dentre as diversas combinações, a interface da Figura 1.3 foi considerada a melhor (dentre as experimentadas), uma vez que foi a responsável por um aumento comprovado de 40.6% do número de inscritos, convertidos em um adicional de 57 milhões de dólares em doações para a campanha.

Apesar de as possibilidades para construção de uma página *web* serem muitas, alguns autores desenvolveram técnicas para gerar interfaces de forma automática [28, 44, 35] utilizando algoritmos de inteligência artificial. No entanto, a seleção das soluções candidatas tem base apenas na intuição. Como alguns autores defendem, essa forma de escolha pode não ser a mais indicada para o negócio [23, 21]. Além disso, como a avaliação é manual, o processo é limitado tanto pelo número de interfaces para avaliação quanto pelo tempo de resposta.

Figura 1.2: Versão original para o *website* da campanha presidencial de Barack Obama em 2007. Disponível em:[34]

Figura 1.3: Versão final para o *website* da campanha presidencial de Barack Obama em 2007. Disponível em:[34]

1.1 Hipótese

A hipótese que esta pesquisa quer validar é a de que, dado um conjunto pré-fixado de estilos, é possível encontrar de forma automática uma combinação que apresente melhores resultados para o negócio. Um exemplo é o *design* obtido para o *website* da campanha presidencial de Obama, mostrado na Figura 1.3. Ele apresentou melhores resultados para o negócio pois, dentre as variações possíveis, foi o que proporcionou um aumento no número de cliques no botão.

1.1.1 Estratégia

Para tentar validar sua hipótese, este trabalho combina técnicas para o desenvolvimento de uma aplicação, denominada UIDesign, que cria, gerencia e avalia diferentes *designs* de uma página *web* de forma a encontrar aquele que traz melhores resultados para o negócio. Além disso ele aplica tal ferramenta num sistema *web* real e analisa os resultados obtidos. As técnicas que servem de alicerces para a elaboração de UIDesign são três: algoritmos genéticos [16, 27], *web analytics* [23, 21] e experimentos controlados [21].

A adoção dos algoritmos genéticos como técnica para otimização das interfaces é devido à: a) ampla variedade de *designs*, dadas as diversas combinações possíveis entre os elementos HTML e suas propriedades; b) facilidade de modelagem do problema; c) sua característica de experimentação simultânea das versões (soluções candidatas) de forma iterativa.

Uma vez criados pelo algoritmo genético, como avaliar cada um dos *designs*? Como decidir pelo melhor sem intervenção humana no processo? Para isso, o *web analytics* permite entender sobre o comportamento dos usuários no sistema e dão o

feedback necessário para decidir se o sistema e seus objetivos estão alinhados [23].

Experimentos controlados fornecem a metodologia de experimentação em tempo real de interfaces. Eles permitem a validação de modificações com base em métricas de interesse [21, 34]. De forma geral, eles expõem diferentes versões de uma interface a grupos de usuários, coletam e analisam essas métricas. As métricas de interesse são informações obtidas a partir das interações dos usuários com a interface, como o total de cliques em um botão, o total de acessos e o total de itens vendidos.

Este trabalho se diferencia dos demais porque: a) escolhe um *design* melhor sem intervenção humana (de forma não-assistida); b) avalia *designs* através de dados quantitativos e alcança as metas do negócio; c) permite que o processo de experimentação ocorra indefinidamente.

1.2 Contribuições

A principal contribuição deste trabalho está na abertura de um novo viés para otimização de páginas *web*. Ao invés de validar modificações com base no instinto, em opiniões pessoais, dados de usuários reais são coletados e analisados e servem como guia para a validação das alterações [32]. Esses dados permitem ainda que todo o processo ocorra sem a necessidade de um supervisionamento.

Além disso, este trabalho contribui para a construção de uma ferramenta (UIDesign) que comprova que é possível otimizar *designs* de páginas *web* automaticamente utilizando métricas de interesse do negócio [32]. Os *designs* obtidos alinham sistema e objetivos do negócio e podem trazer o diferencial competitivo para empresas. UIDesign já está em funcionamento e pode ser utilizada por qualquer sistema

web.

Outra contribuição está nas pesquisas realizadas sobre *web analytics* e experimentos controlados. Além dos conceitos, este trabalho traz diversas referências para exemplos de aplicações de ambas as técnicas, que podem servir de inspiração para outros trabalhos.

1.3 Organização do trabalho

Os próximos capítulos detalham toda a pesquisa realizada para a elaboração deste trabalho. O Capítulo 2 apresenta os conceitos e as técnicas utilizados, como por exemplo os algoritmos genéticos, *web analytics* e experimentos controlados. O Capítulo 3 discute alguns dos trabalhos publicados que motivaram o desenvolvimento deste. O Capítulo 4 descreve a ferramenta UIDesign, sua arquitetura e como as técnicas até então apresentadas se combinam para otimizar o *design* de uma página *web*. O Capítulo 5 utiliza a aplicação descrita no Capítulo 4 num sistema *web* real e analisa os resultados obtidos. O Capítulo 6 encerra o trabalho, com a conclusão e os trabalhos futuros.

2 CONCEITOS BÁSICOS

Este capítulo apresenta as técnicas utilizadas neste trabalho. A Seção 2.1 aborda os algoritmos genéticos e cada uma de suas etapas. A Seção 2.2 apresenta a ideia de *web analytics*, sua importância e aplicações, bem como alguns exemplos de ferramentas usadas hoje no ramo comercial. Por último, a Seção 2.3 apresenta os experimentos controlados, como o teste A/B e o teste multivariado.

2.1 Algoritmo Genético

Algoritmos genéticos são métodos de busca inspirados na teoria da evolução de Charles Darwin. Propostos por John Holland na Universidade do Michigan no início dos anos 70, são aplicados a diversos problemas de otimização [27, 16].

A principal característica desses algoritmos é a evolução de um conjunto de soluções (uma população de indivíduos) ao longo de iterações. O Algoritmo 1 descreve o processo de forma simplificada.

Algoritmo 1: Passos do algoritmo genético

1. Inicialize a população com indivíduos gerados aleatoriamente.
2. Avalie o nível de adaptação de cada indivíduo.
3. Selecione dois membros da população com probabilidades proporcionais a seus níveis de adaptação.
4. Aplique o cruzamento com probabilidade igual à taxa de cruzamento.
5. Aplique a mutação com probabilidade igual à taxa de mutação.
6. Repita os passos de 3 a 5 até que membros suficientes sejam gerados para a próxima geração.
7. Retorne ao passo 2 até o critério de parada ser atingido.

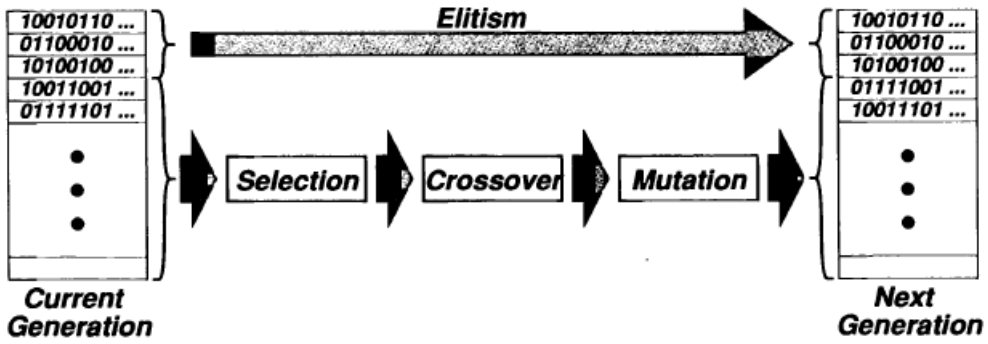

Figura 2.1: Criação de uma nova geração. Disponível em: [16]

A Figura 2.1 ilustra uma iteração: indivíduos selecionados de uma geração passam pelas operações genéticas (como recombinação e mutação) para darem origem a uma nova geração. Aqueles indivíduos considerados mais aptos tem uma chance maior de sobreviver e, portanto, de se reproduzir e propagar suas informações às novas gerações. É comum ainda utilizar o elitismo para garantir que os melhores indivíduos de uma geração integrem a geração seguinte e assim garantir que a qualidade da solução aumente com o passar do tempo [1].

2.1.1 Codificação da solução em indivíduo

A codificação da solução em indivíduo é uma etapa anterior à execução do algoritmo. A escolha da representação é uma das partes mais importantes pois tem impacto direto, no desempenho e também nas operações de cruzamento e mutação [16]. Em geral o indivíduo é representado por uma cadeia de bits (caracteres zeros e uns) de tamanho fixo, mas também existem outras representações frequentes, como as cadeias de caracteres e de números reais.

2.1.2 Avaliação da adaptabilidade do indivíduo

Após a criação da população, indivíduos são avaliados. A cada um é atribuído uma pontuação, chamada nível de adaptação (*fitness value*). Esse valor mensura a qualidade da solução e é atribuído ao indivíduo pela função de adaptação (*fitness function*).

2.1.3 Método de seleção

Após a avaliação, indivíduos são selecionados para criar a nova geração. Em geral, a probabilidade de escolha é proporcional ao nível de adaptação do indivíduo.

Esta etapa tem o propósito de destacar os mais aptos na esperança de que seus descendentes sejam ainda melhores. Ela é análoga à sobrevivência do mais apto no processo de seleção natural.

O método da roleta é um dos algoritmos utilizados como método de seleção de indivíduos. Nesse método, cada indivíduo recebe uma fatia da roleta, sendo a área proporcional ao seu nível de adaptação. A probabilidade de se escolher um indivíduo é, portanto, proporcional a sua área na roleta [26].

2.1.4 Operador de cruzamento (*crossover*)

Uma vez selecionados os indivíduos que participarão do cruzamento (recombinação), a operação é aplicada a um par com uma certa probabilidade, chamada taxa de cruzamento. O processo é análogo ao processo de cruzamento biológico, onde segmentos dos cromossomos dos pais são passados aos filhos para que estes

sejam ainda melhores.

No *one-point crossover*, um ponto no cromossomo é escolhido de forma aleatória e os pais trocam as partes, gerando dois novos indivíduos, como apresentado na Figura 2.2. Já no *two-point crossover*, dois pontos no cromossomo são escolhidos de forma aleatória e a cadeia entre eles é trocada entre os pais, gerando dois novos indivíduos (Figura 2.3).

2.1.5 Operador de mutação

A função da mutação é evitar que a população fique estagnada, inibindo uma convergência prematura. A forma mais comum de aplicar a mutação é trocar um bit aleatório do cromossomo (assumindo que o indivíduo é uma sequencia de bits). Essa operação também possui uma probabilidade de ocorrência, chamada taxa de mutação (Figura 2.4), que em geral é mantida baixa. Se a taxa de mutação fosse alta, o algoritmo seria resumido a uma busca aleatória.

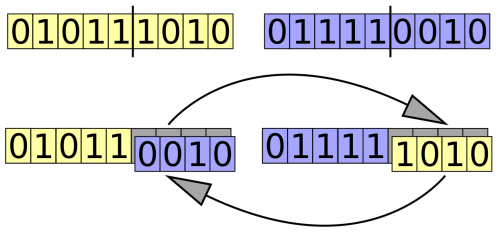

Figura 2.2: One-point crossover

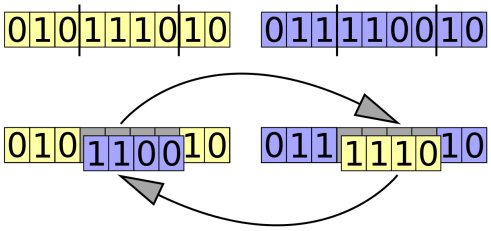

Figura 2.3: Two-point crossover

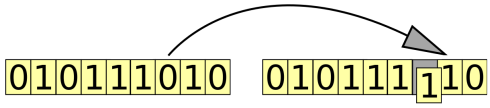

Figura 2.4: Mutação

2.2 *Web analytics*

De acordo com a *Digital Analytics Association* (DAA) [6], "*web analytics* é coletar, medir, reportar e analisar dados quantitativos da Internet para otimizar *websites* e iniciativas de *web marketing*" [18]. Seu principal objetivo é compreender o comportamento dos usuários para que otimizações sejam realizadas no *site* de forma a melhorar a experiência do usuário [33].

Desde seu surgimento em 1991 até os dias atuais, *web analytics* tem evoluído e, de acordo com Singal *et al.*, é uma industria em constante evolução [33]. De lá para cá, surgiram diversas ferramentas e serviços para acessar e coletar informações de usuários como por exemplo o Google Analytics [10], o Facebook Page Insights [7] e a Piwik [31].

Diversos trabalhos confirmam sua utilidade e importância na otimização de *websites*, nas mais diversas áreas. Sua aplicação traz mudanças nas relações entre

organizações e seus clientes pois podem ser extraídas informações valiosas que auxiliam as decisões do negócio [23]. Landim *et al.* utilizam ferramentas de *web analytics* para impulsionar as vendas de um *website* de serviços de tecnologia da informação [25]. Enquanto isso, Vivekananthamoorthy *et al.* utilizam-nas para conseguir os *feedbacks* necessários para focar nas melhorias de seus sistema de *e-learning* de forma a suprir os objetivos do aprendizado [37]. Um uso semelhante é feito por Grout *et al.* [14] e Barbosa [2].

2.2.1 Métricas

No contexto de *web analytics*, a métrica é a informação mais rudimentar extraída do comportamento de usuários, como por exemplo o número de visitantes e o tempo de cada visita [33]. Existem quatro categorias: uso do *site*, *refereers* (como visitantes chegaram ao *site*), análise de conteúdo do *site* e garantia de qualidade [3]. A Tabela 2.1 apresenta alguns tipos de métricas *web*.

A tradução dos objetivos do negócio em uma combinação eficaz de métricas permite a otimização do *website* de forma que usuários e negócio sejam atendidos [33]. Essas métricas devem ser combinadas de forma a criar um indicador que possua significado para o negócio, chamado indicador-chave de *performance* (do inglês, *Key Performance Indicator*, KPI). Cada tipo de *website* possui um KPI diferente, como apresentado na Tabela 2.2 [3].

2.2.2 Ferramentas de *web analytics*

Atualmente existem algumas ferramentas de *web analytics*, como Piwik [31] e Google Analytics [10]. Além de capturar as informações dos usuários, elas permitem

Tabela 2.1: Tipos de métricas. Disponível em: [3]

Uso do *site*
Número de visitantes e sessões
Quantas pessoas repetidamente visitam o *site*
Informação geográfica
Atividade do motor de busca
Refereers
Quais *websites* estão enviando visitantes para seu *site*
Quais termos estão sendo utilizados nas buscas para encontrar o seu *site*
Quantas pessoas colocam *bookmarks* no *site*
Análise de conteúdo do *site*
Top entry pages
Páginas mais populares
Principais páginas de saída
Principais caminhos pelo *site*
Eficácia do conteúdo-chave
Garantia de qualidade
Páginas quebradas ou erros de servidor
Resposta do visitante a erros

analisar os dados e exibi-los por meio de relatórios personalizados. A seguir, são apresentadas as duas ferramentas citadas.

2.2.2.1 Piwik

A Piwik [31] é uma plataforma de *analytics* aberta (sob a licença GPL [8] de *software*), que dentre outras funções, coleta e analisa informações acerca dos usuários de uma aplicação *web* em tempo real [31] (Figura 2.5). Esta ferramenta possui as métricas mais comuns utilizadas em ferramentas de *analytics*: informações acerca do comportamento do usuário (de onde vem, o que estão fazendo e para onde vão), monitoramento de *downloads* e ainda a possibilidade de expansão de funcionalidades por meio de *plugins*. A lista completa pode ser vista em `http:`

Tabela 2.2: Indicadores-chave de *performance* de acordo com o tipo do *website*. Disponível em: [3]

Tipo do *website*	**Indicadores-chave de *performance***
Comércio	Taxa de conversão Valor médio do pedido Valor médio da visita Lealdade do consumidor Taxa de rejeição
Conteúdo/Mídia	Profundidade da visita Taxa de retorno de visitante Taxa de novos visitantes Profundidade de página
Suporte/Auto-atendimento	Profundidade de página Taxa de rejeição Satisfação do cliente Principais frases de busca

`//piwik.org/features/`.

2.2.2.2 Google Analytics

O Google Analytics [10] oferece um serviço gratuito de *web analytics*, que permite a captura e análise de diversas métricas [11]. Lançado em 2005, é atualmente o serviço mais utilizado de *web analytics* na Internet [43]. A Figura 2.6 é um exemplo de relatório que pode ser gerado pelo serviço.

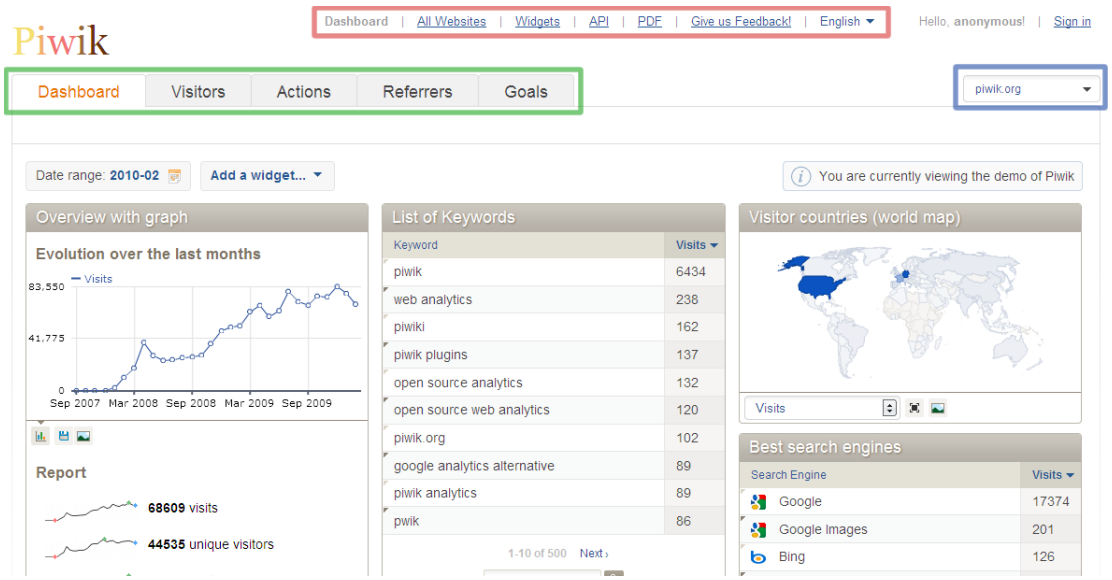

Figura 2.5: Relatórios da Piwik. Disponível em: [30]

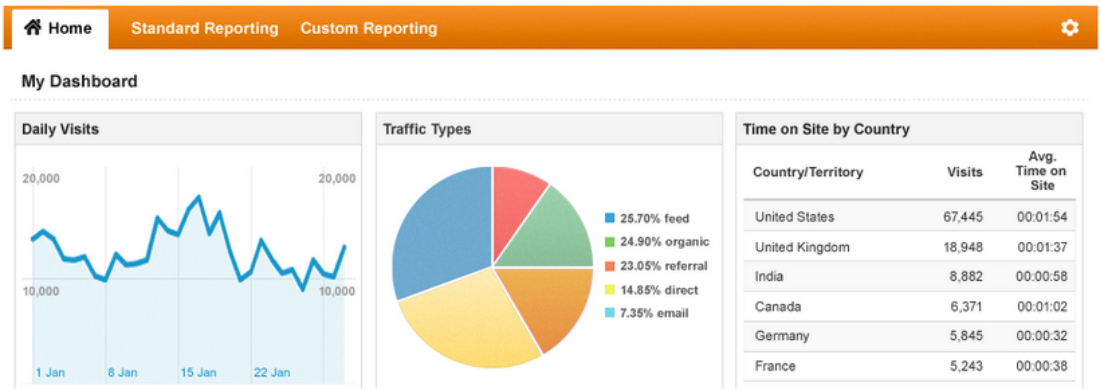

Figura 2.6: Relatórios do Google Analytics. Disponível em: [9]

2.3 Experimentos Controlados

Experimentos controlados, também chamados experimentos randomizados, permitem a avaliação de ideias em tempo real atribuindo de forma aleatória usuários do sistema a uma das suas diferentes versões em teste. Alterações no comportamento dos usuários diante das mudanças são avaliadas com base na coleta de métricas de interesse e aplicação de testes estatísticos [19]. Kohavi *et al.* apresentam dois métodos de experimentos controlados: o teste A/B, que é mais simples, e o teste multivariado [19]. Ambos serão detalhados a seguir.

2.3.1 Teste A/B

O teste A/B (Figura 2.7) é um experimento controlado simples onde usuários são aleatoriamente expostos a uma das duas variantes: i) a versão original (*Control*); ii) a nova versão (*Treatment*) [19]. Seu objetivo é avaliar se (ii) é estatisticamente diferente de (i), como explicado a seguir.

Para cada versão é calculado o critério de avaliação geral (do inglês, *Overall Evaluation Criterion*, OEC), que pode ser por exemplo o total de unidades vendidas, o lucro obtido ou até mesmo a combinação destes dados. O OEC nada mais é que uma medida quantitativa do objetivo do experimento. Como usuários são escolhidos de forma aleatória, a diferença observada nesses valores é dada como consequência da diferença entre as versões. Através do teste de hipótese [4], tomando como hipótese nula a inexistência de diferença nos OECs, a nova versão é aceita como estatisticamente diferente se a hipótese nula for rejeitada [19].

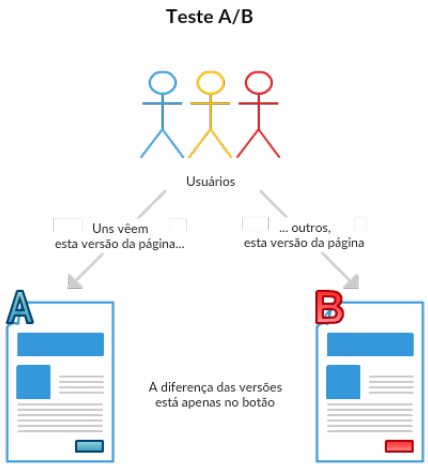

Figura 2.7: Exemplo de teste A/B em que a versão original e uma versão modificada são apresentadas, cada uma, a um grupo de usuários

2.3.2 Teste Multivariável

Enquanto o teste A/B permite que a diferença entre variantes seja a de apenas uma variável, o teste multivariável (do inglês, *Multivariable Testing*, MVT) [19] permite que sejam alteradas diversas variáveis, como ilustrado na Figura 2.8.

Figura 2.8: Exemplo de teste multivariável em que cada seção (imagem, botão e texto) é testada com duas variações cada uma. Disponível em: [15]

Devido a suas características, o teste multivariável apresenta algumas vantagens diante do teste A/B. Uma delas é o período de experimentação, que pode

ser mais curto, uma vez que várias modificações são testadas simultaneamente. Isso acelera tanto a identificação quanto a aplicação da melhoria. Outra vantagem do MVT é permitir estimar as relações entre as variáveis em teste, uma vez que mais de uma variável pode ser avaliada por vez. No entanto, estas interações podem ser prejudiciais para a experiência de usuário e também tornam a interpretação e análise mais difíceis [19].

Holzmann e Hutflesz apresentam um *framework* de experimentação para aplicações nativas de celulares (principalmente Android [12]), que utiliza tanto o teste A/B quanto o multivariável [15]. Já Tang *et al.* apresentam um processo de experimentação aplicado no Google para melhoria do motor de busca e que pode ser utilizado em outras aplicações *web*. A ideia é apresentar um mecanismo que facilite a experimentação, além de torná-las melhores, mais robustas e rápidas [36].

Kohavi *et al.* trazem alguns problemas que podem ser encontrados durante a realização dos experimentos [20, 22] bem como situações inesperadas. Resultados surpreendentes, como o do Office Online [22], demonstram a importância da experimentação para o negócio. Apesar do novo *design* proposto ser mais moderno e limpo de forma que usuários sejam direcionados a clicar nos botões de *download* do Microsoft Office, ele teve um desempenho pior (redução de 64% no número de cliques nesses botões).

3 TRABALHOS RELACIONADOS

A utilização de algoritmos de inteligência artificial na criação de *designs* de interfaces gráficas *web* é o foco do trabalho de alguns autores. No que tange à criação do *design*, alguns lidam com o *layout*, isto é, a forma como os elementos que compõe a página são dispostos [29]; outros, tanto com *layout* quanto com o estilo, que diz respeito à apresentação visual dos elementos [28, 44].

Em seu trabalho, Peñalver e Merelo aplicam o algoritmo genético para posicionar artigos de jornal em uma página *web*, como pode ser visto na Figura 3.1 [29]. O problema, caracterizado como um *page layout problem*, consiste em posicionar retângulos de tamanhos fixos e distintos numa área de largura fixa e altura variável de forma a minimizar os espaços em branco deixados e, consequentemente, maximizar a área ocupada. O algoritmo genético posiciona cada artigo na página, em tempo real. A técnica, aprimorada com um algoritmo guloso, é utilizada para melhorar o resultado obtido pelo *simulated annealing* de Lagus [24].

Oliver *et al.* utilizam um algoritmo genético interativo (AGI) para otimizar a aparência de uma página *web* [28]. *Layout* e estilo são tratados de maneira independente, ou seja, cada um possui seu próprio algoritmo genético. Baseado em seus gostos pessoais, usuários avaliam as páginas candidatas que serão selecionadas pelo algoritmo na etapa de cruzamento. A Figura 3.2 apresenta o exemplo de uma geração com as telas, geradas pelo método, exibidas para avaliação dos usuários. A proposta do presente trabalho é remover a dependência de avaliação assistida para selecionar as soluções candidatas, utilizando para isso um algoritmo genético não-interativo.

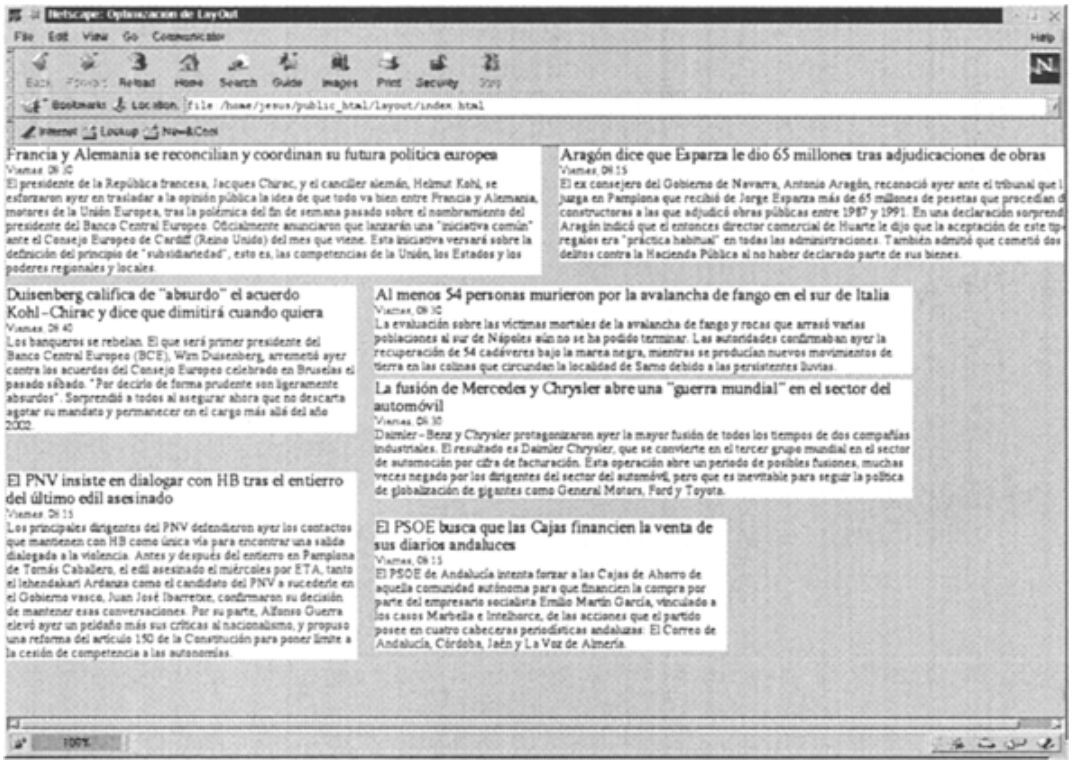

Figura 3.1: Resultado final encontrado pelo algoritmo genético no trabalho de Peñalver e Merelo. Disponível em: [29]

Figura 3.2: Apresentação dos *layouts* de uma geração para seleção. Disponível em: [28]

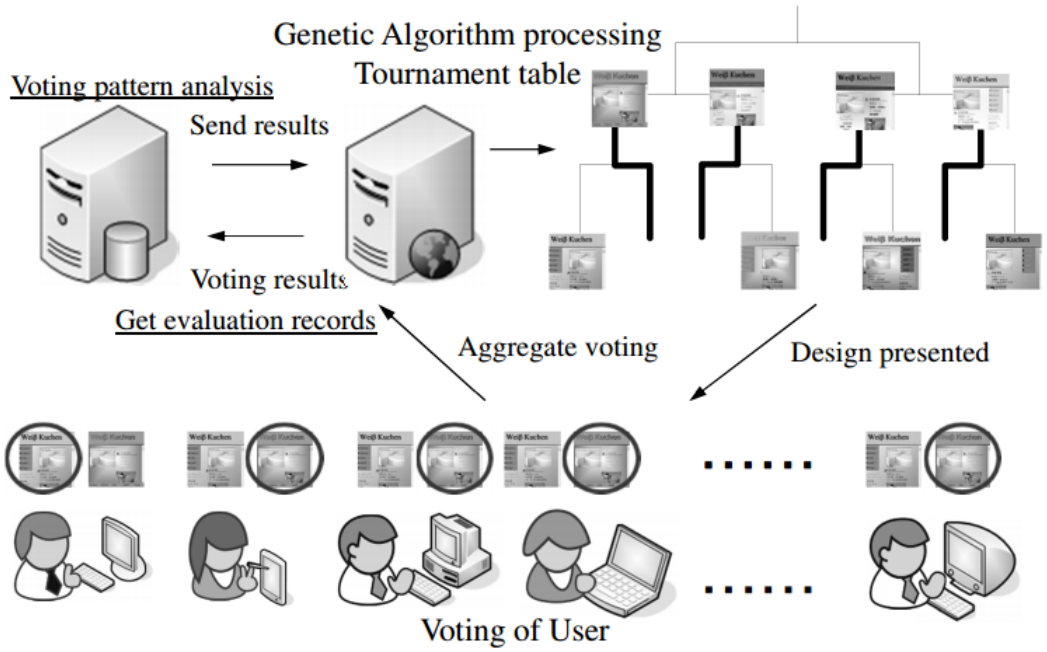

Figura 3.3: Estrutura do funcionamento do sistema de *design* de *website*. Disponível em: [44]

Yokoyama *et al.* em seu artigo trabalham no desenvolvimento de um sistema de *design* de *website* que adota em seu algoritmo genético interativo um novo método de torneio para seleção de uma página candidata [44]. Cada usuário, exposto a duas interfaces geradas pelo AGI, seleciona a que mais o agrada. Os votos são analisados e utilizados para avaliar uma solução do algoritmo genético, como é apresentado na Figura 3.3. O trabalho foi testado com usuários reais apresentando resultados significativos. Uma das melhorias propostas foi uma maneira mais interessante e fácil de avaliação pelo usuário das páginas candidatas.

Quanto ao algoritmo utilizado, com exceção do trabalho de Peñalver e Merelo, todos os demais utilizam o algoritmo genético interativo, onde pessoas participam do processo de avaliação das páginas. Um problema observado nestes métodos é

a ausência de uma análise do comportamento dos usuários na utilização da página para que as modificações sejam baseadas em dados estatísticos ao invés de opiniões pessoais, otimizando o uso da página [23].

Por outro lado, Barbosa [2] em seu projeto faz uso destes dados para aprimorar uma página *web*. Ele analisa as interações dos usuários com a página por meio de dados coletados com *web analytics* e, a partir deste *feedback*, gera novas páginas. Essas são submetidas ao mesmo processo para, ao final, obter aquela considerada ótima, ou seja, a que alcança os indicadores desejados atendendo os objetivos do produto [2].

A geração da interface, isto é, a combinação dos elementos e suas características é feita automaticamente. No entanto, a configuração, ou seja, os possíveis estilos que podem ser assumidos pelos elementos são configurados previamente por um usuário administrador.

Uma vez criados os *designs*, estes são apresentados para usuários e dados são coletados (como por exemplo, o número de cliques num link ou o tempo de permanência de um usuário na página). A partir destas informações, o usuário administrador analisa os resultados obtidos e decide aplicar ou não as modificações. Esta etapa de aceitação ou descarte das mudanças é feito manualmente, ainda que embasada em dados estatísticos.

4 FERRAMENTA DE EXPERIMENTAÇÃO

Como apresentado no Capítulo 1, para validar a hipótese, a estratégia deste trabalho é combinar três técnicas para elaborar uma ferramenta chamada UIDesign, aplicá-la a um sistema *web* real e analisar os resultados obtidos. O objetivo deste capítulo é apresentar UIDesign com mais detalhes, enquanto o Capítulo 5 mostra sua aplicação em um sistema *web* real.

A Seção 4.1 aborda UIDesign sucintamente, descrevendo seu objetivo, como ela faz para atingi-lo e ainda suas limitações. Já a Seção 4.2 aborda as técnicas utilizadas para sua elaboração, enquanto a Seção 4.3 trata da arquitetura da ferramenta e apresenta como as técnicas se combinam para o desenvolvimento de UIDesign.

4.1 O que é UIDesign?

UIDesign é uma ferramenta *web* que visa encontrar um *design* para uma página de um sistema *web* que apresente melhores resultados para o negócio. Por exemplo, no caso do *website* da campanha de Barack Obama apresentado no Capítulo 1, é aquele que proporcionou o maior número de cliques no botão.

UIDesign também objetiva um processo de decisão automático. Isso significa que, dada uma configuração inicial, ela é capaz de criar *designs* e encontrar, sem intervenção humana, aquele com as características citadas anteriormente. Desta forma, a ferramenta se diferencia de outras que demandam constante interação, como as apresentadas no Capítulo 3.

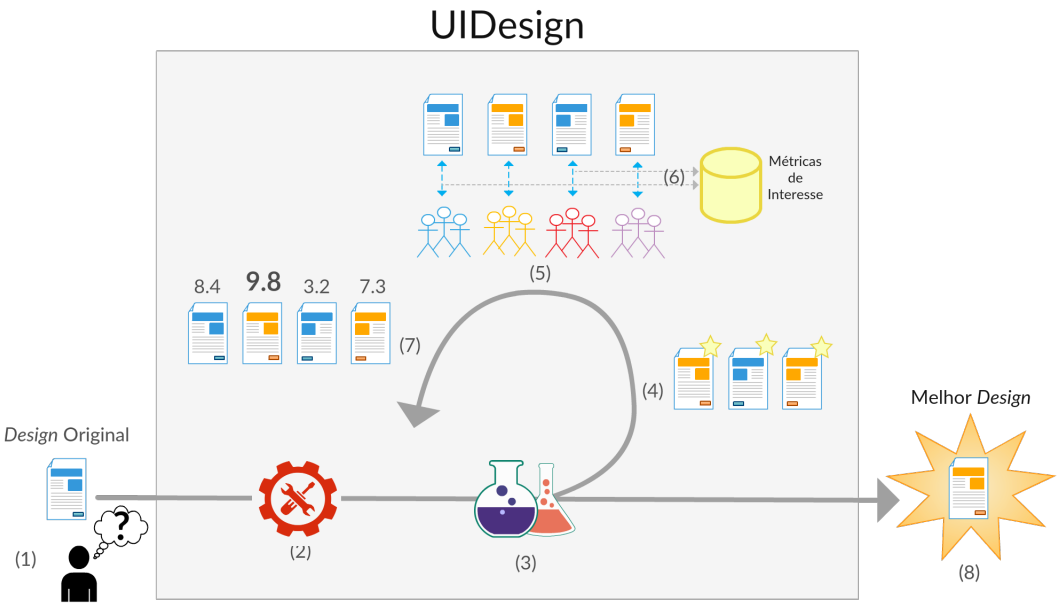

Figura 4.1: Visão geral do funcionamento de UIDesign. (1) Dada a página de um sistema *web*, será possível obter um *design* melhor que o atual? UIDesign objetiva encontrá-lo. (2) Após a configuração da ferramenta (Seção 4.3.1), (3) UIDesign tenta encontrar o melhor *design* a partir de experimentos sucessivos. (4) No início de cada experimento, a ferramenta cria diversos *designs*. (5) Ao longo de um período, ela aleatoriamente escolhe quais usuários interagem com quais *designs*. (6) Dados de relevância para o negócio são coletados a partir dessas interações, como por exemplo o tempo de permanência na página, número de cliques etc. (7) Ao final do experimento, UIDesign avalia cada um dos *designs* com base nos dados coletados. (8) Ao final, a ferramenta indica aquele que obteve os melhores resultados.

Para atingir seu objetivo, UIDesign se apóia numa experimentação iterativa, ilustrada na Figura 4.1. Cada iteração é um experimento onde ela cria diferentes *designs*, ou seja, diferentes combinações de estilos para os elementos da página. Em seguida, durante um certo período, ela expõe a grupos aleatórios de usuários uma das versões do experimento. Ao final desse período, cada *design* passa por um processo de avaliação que utiliza os dados extraídos do comportamento dos usuários enquanto eles interagiam com a nova versão da página. Estes dados estão relacionados com os objetivos do negócio e são o foco da otimização. Após algumas iterações, UIDesign

indica o que obteve os melhores resultados.

É importante destacar que UIDesign lida apenas com alterações nos estilos dos elementos de uma página, como por exemplo cor, tamanho, fonte etc [40, 41]. Alterações nos elementos que a compõe e na informação veiculada não estão no escopo de experimentação previsto pela ferramenta. UIDesign, portanto, não otimiza os *designs* com base na inclusão e/ou exclusão de elementos tampouco na alteração do conteúdo, mas sim através da alteração das propriedades dos elementos já existentes na página.

4.2 Técnicas

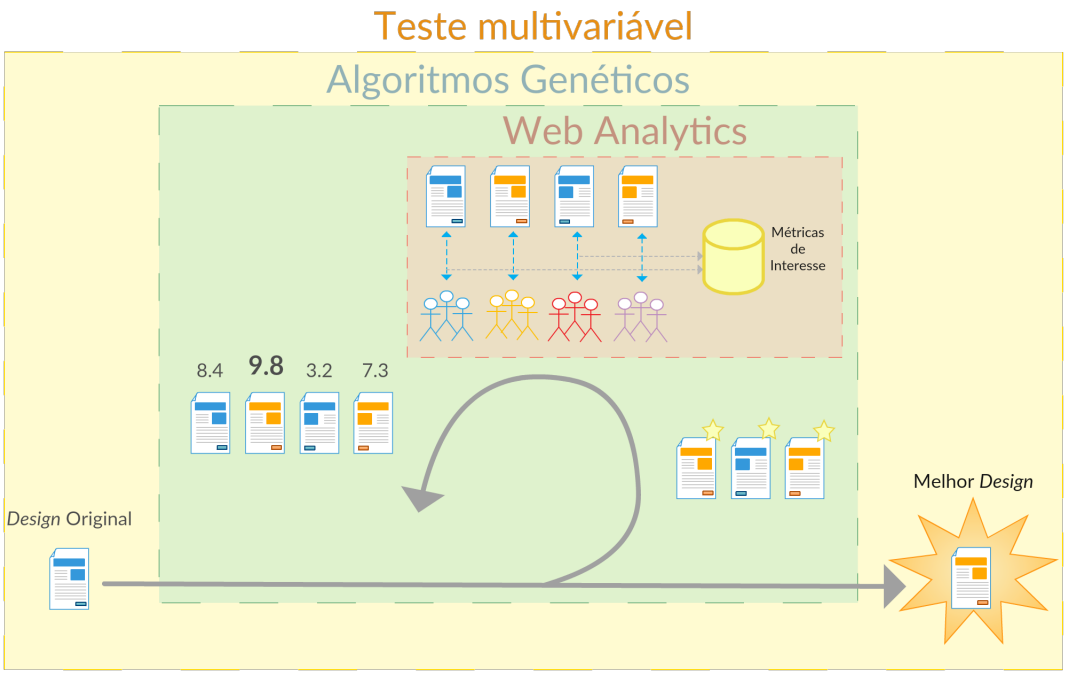

Figura 4.2: Alicerces de UIDesign

Três técnicas servem de alicerces para UIDesign: algoritmos genéticos (Seção

2.1), *web analytics* (Seção 2.2) e experimentos controlados - em especial, o teste multivariável (Seção 2.3.2). A Figura 4.2 ilustra a combinação dessas técnicas para a construção e o funcionamento da ferramenta.

Algoritmos genéticos oferecem uma técnica para resolver problemas de otimização onde o espaço de busca é extenso. Como elementos possuem uma variedade de propriedades modificáveis [41], é possível criar diversas combinações, diversos *designs* para uma única página *web*. Em UIDesign, os algoritmos genéticos gerenciam os experimentos, ou seja, criam as combinações de estilos para os elementos da página, avaliam cada uma e iniciam um novo ciclo de experimentação até encontrar a melhor.

Já *web analytics* atua na coleta e armazenamento das informações acerca das interações dos usuários com os *designs* gerados. Esses dados são posteriormente utilizados na etapa de avaliação do algoritmo genético permitindo, portanto, que ela ocorra sem a necessidade de intervenção humana e considerando apenas os interesses do negócio.

Por fim, o teste multivariável embasa o processo de experimentação em que usuários, escolhidos aleatoriamente, interagem com os *designs* criados pela ferramenta. A avaliação de cada um tem como base dados de interesse do negócio, obtidos através dos usuários.

A combinação destas técnicas permitiu desenvolver uma ferramenta que:

- aplica um processo de experimentação não assistido de diversas versões simultâneas;
- cria *designs* que evoluem a cada iteração;

- entrega um *design* melhor de acordo com os dados obtidos dos próprios usuários sem a intervenção de opiniões pessoais.

4.3 Arquitetura

Cada técnica (Seção 4.2) necessita de um conjunto de informações para funcionamento. Por exemplo, o algoritmo genético precisa da definição do número de indivíduos de cada geração, das especificações da função de adaptação, da definição do método de seleção etc. Como as necessidades variam de aplicação para aplicação, esses parâmetros também sofrem variações. Por isso, UIDesign possui duas etapas: a configuração e a experimentação (Figura 4.3).

Na primeira, a ferramenta recebe os parâmetros para inicializar as técnicas. Esses são, por exemplo, o número de *designs* a serem criados e avaliados em cada iteração, o tempo de duração de cada iteração (experimento), as métricas de interesse (Seção 2.2.1) etc. Já a segunda corresponde à experimentação propriamente dita, onde UIDesign executa uma série de experimentos de forma iterativa.

A etapa de configuração permite o reuso de UIDesign como ferramenta de experimentação por diversas aplicações *web*. Apesar de utilizar as mesmas técnicas em todos os experimentos (independente da aplicação), é nessa etapa que UIDesign obtém os dados para adequar-se às necessidades específicas de cada aplicação, dando à ferramenta a especialização necessária para realização do seu trabalho.

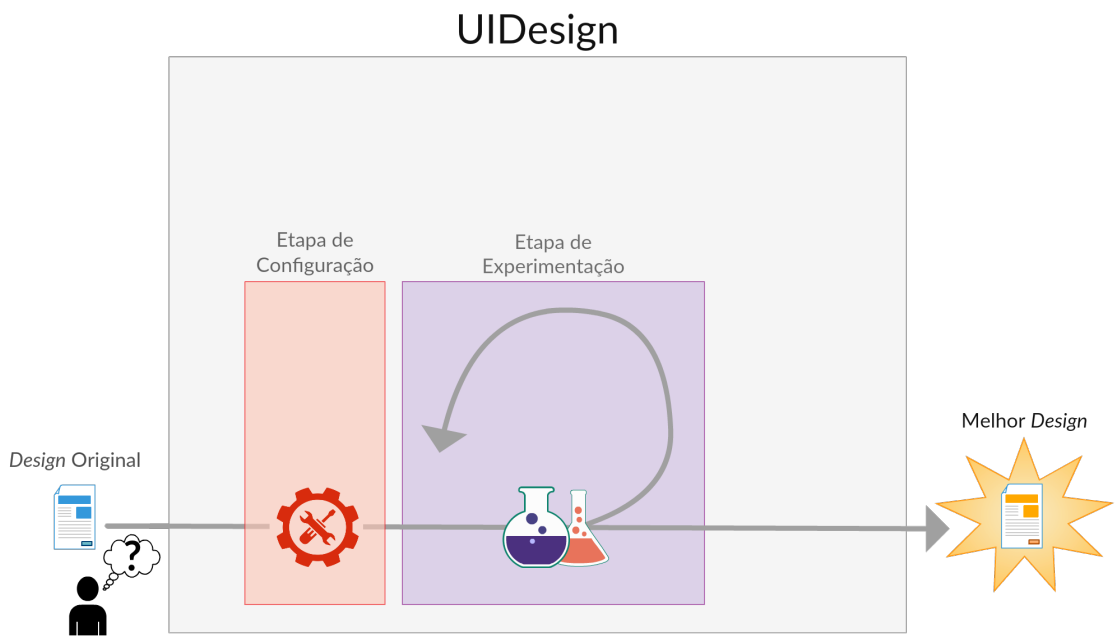

Figura 4.3: Etapas de UIDesign

4.3.1 Etapa de configuração

Na etapa de configuração, UIDesign recebe um conjunto de parâmetros para criar os experimentos específicos da aplicação. O usuário administrador é o agente responsável pelo fornecimento destes dados. Ele é a representação de cada cliente interessado na otimização da interface da aplicação *web*, como por exemplo, o dono da aplicação. Os parâmetros fornecidos à ferramenta são informações relevantes para a execução dos experimentos como: a) Número de versões/*designs* em cada experimento; b) Duração do experimento; c) Número máximo de experimentos; d) Mudanças a serem aplicadas às propriedades dos elementos; e) Ferramenta de *web analytics* escolhida para coletar informações da página; f) Dados de interesse do negócio para otimização.

Essa etapa é segmentada em quatro passos, como ilustrado na Figura 4.4. Em

cada um o usuário administrador cadastra um subconjunto dos parâmetros listados anteriormente.

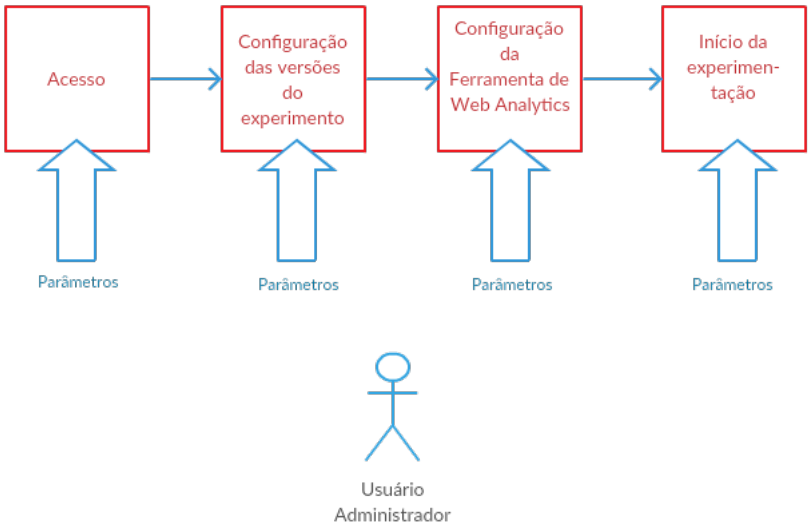

Figura 4.4: A etapa de configuração é dividida em passos, que recebem um conjunto de parâmetros. Quando o usuário administrador retorna ao sistema, ele prossegue a partir do último passo, não sendo possível alterar as informações fornecidas nos anteriores.

4.3.1.1 Primeiro passo: Acesso

No primeiro passo, ilustrado na Figura 4.5, o usuário administrador realiza o seu acesso à UIDesign, fornecendo um nome de usuário e uma senha previamente cadastrados. Caso seja o primeiro acesso, ele deve efetuar seu cadastro.

Assim que o acesso é permitido, o usuário é direcionado para o último passo não concluído ou, no caso de todos os passos concluídos, ele é direcionado para a página de visualização de resultados. Sendo o primeiro acesso, ele é encaminhado para o segundo passo.

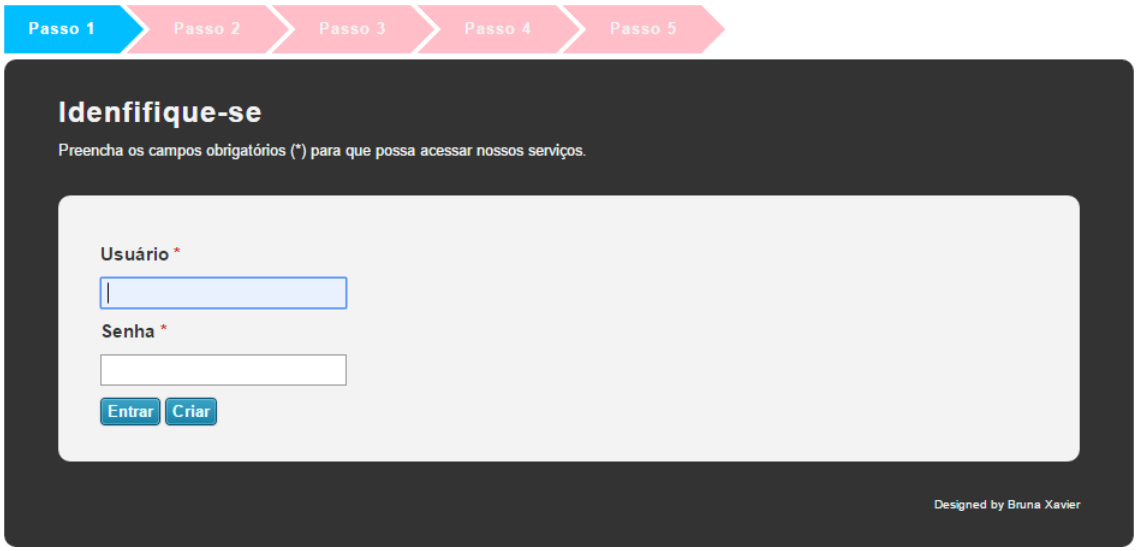

Figura 4.5: Primeiro passo da etapa de configuração

4.3.1.2 Segundo passo: Configuração das versões do experimento

No segundo passo (Figura 4.6), o usuário administrador informa dados para a criação do experimento. Os parâmetros são basicamente para a configuração do algoritmo genético.

Número de versões : número de *designs* por experimento ou o número de indivíduos de cada geração do algoritmo genético.

Tempo de duração de uma geração : tempo de cada iteração da etapa de experimentação.

Número máximo de gerações : número máximo de experimentos que devem ser realizados antes de determinar qual o melhor.

Classes de estilo : utilizadas para a criação dos indivíduos. Junto com elas, o usuário fornece ainda os elementos aos quais elas podem ser associadas. Por

exemplo

- elemento1: classe1, classe2;
- elemento2: classe3;

indica que nos *designs* gerados, o elemento1 pode ter somente classe1, somente classe2 ou nenhuma delas enquanto o elemento2 pode ter ou não a classe3. O uso de classes de estilo [42] permite alterações em diversas propriedades de um mesmo elemento. Isso dá uma flexibilidade maior para o usuário que visa experimentar uma combinação de alterações.

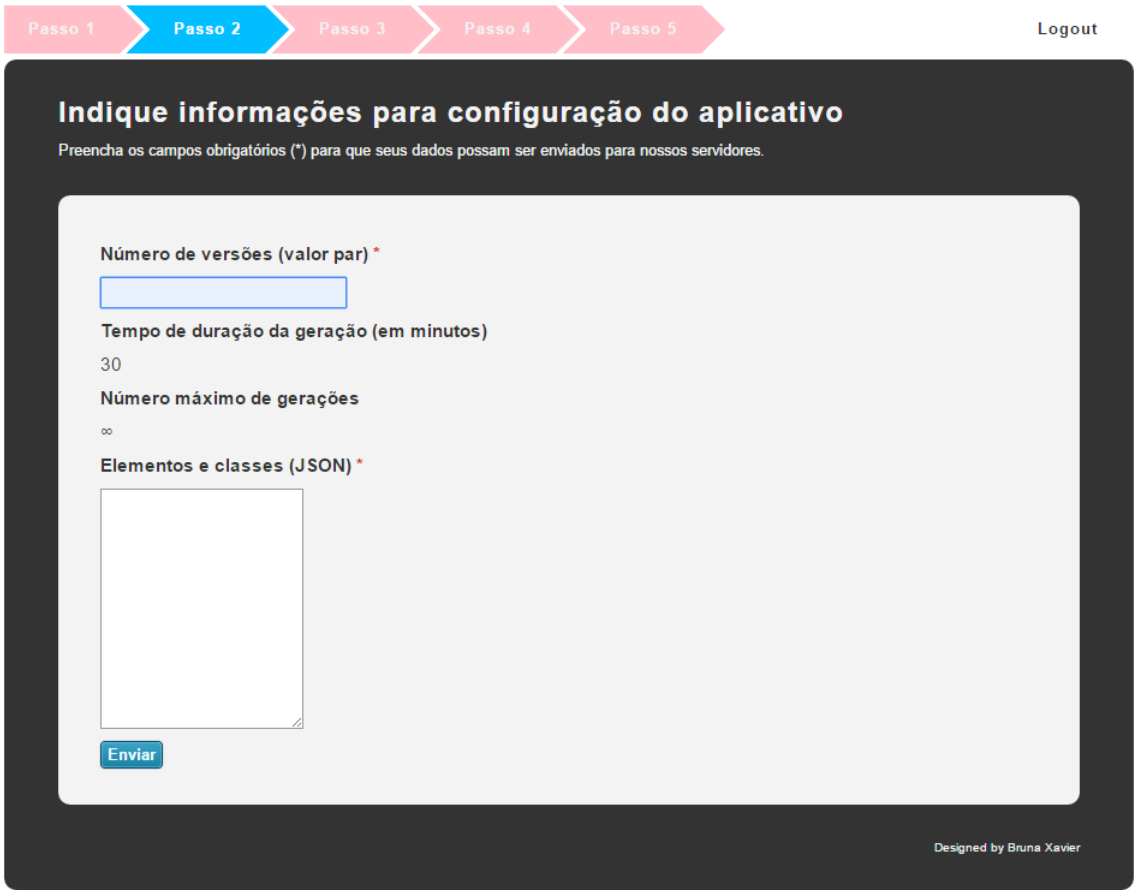

Figura 4.6: Segundo passo da etapa de configuração

4.3.1.3 Terceiro passo: Configuração da ferramenta de *web analytics*

O terceiro passo (Figura 4.7) recebe os parâmetros necessários para a avaliação dos *designs*. UIDesign utiliza um processo de avaliação transparente tanto para o usuário administrador, quanto para os usuários do sistema sob experimentação. Ela não utiliza informações advindas de um único usuário com base em seus interesses pessoais [28], tampouco utiliza votação para que seus usuários informem qual dentre as versões é a melhor [44]. Ao invés disso, ela utiliza informações capturadas dos usuários enquanto eles interagem com a página. O tipo de informação que deve ser utilizado nessa avaliação é configurado neste passo, bem como a indicação da ferramenta utilizada para capturar e armazenar esses dados.

UIDesign não implementa uma ferramenta de *web analytics* (Seção 2.2). Ao invés disso, na fase de avaliação (Seção 4.3.2.3), ela requisita as informações de uma escolhida pelo usuário administrador para avaliar os indivíduos. UIDesign portanto depende que o usuário administrador utilize uma ferramenta externa para coletar e armazenar dados.

Neste passo, o usuário informa qual ferramenta de *web analytics* ele utiliza para coletar dados. Atualmente, UIDesign permite apenas Piwik (Seção 2.2.2.1) e Google Analytics (Seção 2.2.2.2). Junto com a ferramenta, ele indica também quais métricas (Seção 2.2.1) devem ser consideradas, como por exemplo, o número de cliques num certo botão ou o tempo de permanência na página. Essas métricas refletem o desejo do negócio para com o sistema *web* permitindo que a avaliação dos *designs* seja tal que traduza a aproximação do sistema às metas do negócio.

Passo 1 Passo 2 Passo 3 Passo 4 Passo 5 Logout

Indique informações da sua ferramenta de Web Analytics para análise de resultados

Preencha os campos obrigatórios (*) para que seus dados possam ser enviados para nossos servidores.

Ferramenta utilizada *

Google Analytics (analytics.js)

ID do profile do Google Analytics *

ex.: ga:12345

Atenção!

- Cadastre o usuário "46756727074-h57fqdnom7g7tmfnadi8g7239hhs4pje@developer.gserviceaccount.com" com permissão de "Ler e analisar" no seu Google Analytics. VEJA COMO FAZER CLICANDO AQUI..

- Cadastre a dimensão personalizada 1 no seu Google Analytics. VEJA COMO FAZER CLICANDO AQUI..

Métrica 1 * **Peso para métrica 1** *

ex.: ga:sessions

Métrica 2 **Peso para métrica 2**

ex.: ga:sessions

Métrica 3 **Peso para métrica 3**

ex.: ga:sessions

Filtro

ex.: ga:county==Brazil

Validar

Designed by Bruna Xavier

Figura 4.7: Terceiro passo da etapa de configuração

4.3.1.4 Quarto passo: Início da experimentação

Este último passo dá início à etapa de experimentação. Uma vez iniciada, o usuário administrador pode acompanhar o seu andamento, visualizando os experimentos realizados e qual dos *designs* teve o melhor resultado.

4.3.2 Etapa de experimentação

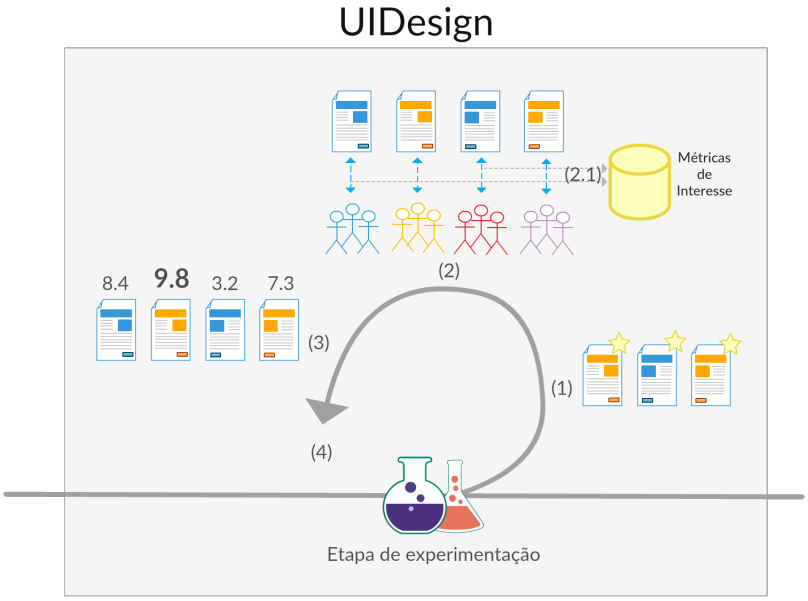

Figura 4.8: Etapas da fase de experimentação: (1) Criação dos *designs*; (2) Apresentação aos usuários; (2.1) Coleta de métricas de interesse; (3) Avaliação dos *designs*; (4) Finalização do experimento

Como apresentado na Figura 4.8, a etapa de experimentação possui quatro fases. A primeira corresponde à criação dos *designs* do experimento. A segunda, à apresentação de cada um a um grupo de usuários. Nesta fase, informações das interações usuário-sistema são capturadas e armazenadas. A terceira utiliza esses dados para avaliar cada *design*. Por fim, a última decide se UIDesign prossegue para

uma nova experimentação ou se o *design* que melhor aproxima sistema e negócio foi encontrado.

UIDesign utiliza o algoritmo genético para gerenciar os experimentos, seguindo as orientações do teste multivariável. Desta forma, cada iteração de experimentação corresponde a uma iteração do algoritmo genético e cada indivíduo, um dos *designs* do experimento.

4.3.2.1 Primeira fase: Criação dos *designs*

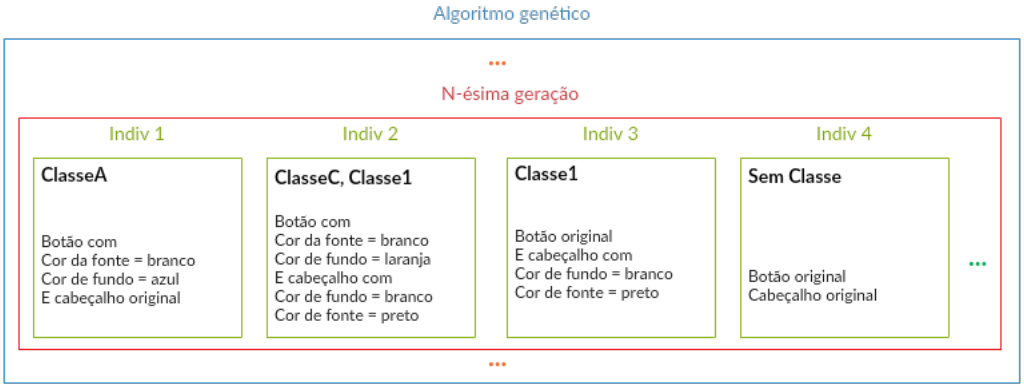

Figura 4.9: Cada indivíduo representa as classes de estilo que devem ser aplicadas a elementos do *design* original

A primeira fase corresponde à criação dos indivíduos do algoritmo genético. O indivíduo representa quais classes de estilo [17, 5] serão aplicadas aos elementos da página *web*, gerando um *design* para experimentação (Figura 4.9). Ele é representado por uma cadeia de bits (zeros e uns) cujo tamanho varia de acordo com as possíveis classes dos elementos, como exemplificado adiante.

Suponha que numa página existem três possíveis classes de estilo para o botão: a) classeA: azul para cor de fundo e branco para cor da fonte; b) classeB:

vermelho para cor de fundo e branco para cor da fonte; c) classeC: laranja para cor de fundo e branco para cor da fonte. E além disso, duas para o cabeçalho: i) classe1: branco para cor de fundo e preto para cor da fonte; ii) classe2: preto para cor de fundo e branco para cor da fonte.

Neste exemplo, os indivíduos possuem quatro *bits*. Os dois primeiros correspondem às mudanças do botão e os dois últimos às do cabeçalho. As Tabelas 4.1 e 4.2 mostram todos os possíveis indivíduos que podem ser gerados pelo algoritmo genético neste exemplo e as mudanças que cada um representa. Observando essas tabelas, o indivíduo 1010 (Figura 4.10), por exemplo, representa um *design* em que o botão aparece vermelho (cor de fundo) com fonte branca e o cabeçalho aparece com cor de fundo preta e fonte branca.

Tabela 4.1: Indivíduos e classes de estilo do botão

Indivíduo	**Botão**
00XX	não aplica mudança (mantém original)
01XX	aplica classeA
10XX	aplica classeB
11XX	aplica classeC

Tabela 4.2: Indivíduos e classes de estilo do cabeçalho

Indivíduo	**Cabeçalho**
XX00	não aplica mudança (mantém original)
XX01	aplica classe1
XX10	aplica classe2
XX11	não aplica mudança (mantém original)

É na etapa de configuração (Seção 4.3.1) que UIDesign recebe o número de indivíduos que devem ser gerados em cada iteração e as classes de estilo que devem ser atribuídas aos elementos.

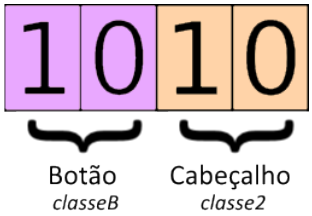

Figura 4.10: Indivíduo 1010

4.3.2.2 Segunda fase: Apresentação aos usuários

A segunda fase apresenta os indivíduos (*designs* representados por eles) aos usuários do sistema *web*. O objetivo é coletar informações sobre como usuários se comportam em um ou outro *design*.

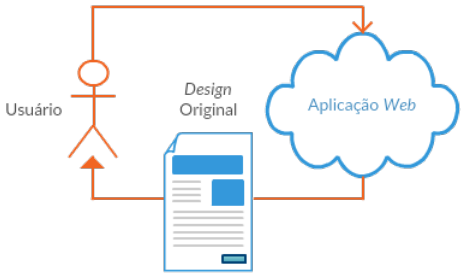

Figura 4.11: Usuário recebe a página *web* como resposta a sua requisição

Quando um usuário acessa o endereço de uma aplicação *web*, ele recebe como resposta uma página com a qual ele pode interagir (Figura 4.11). Com a utilização de UIDesign, antes de o usuário ver a página, a ferramenta seleciona aleatoriamente um dos indivíduos da geração vigente (um dos *designs* da experimentação vigente) e aplica as classes de estilo aos respectivos elementos da página (Figura 4.12). O usuário portanto tem a sensação de que está interagindo com a página em sua interface original.

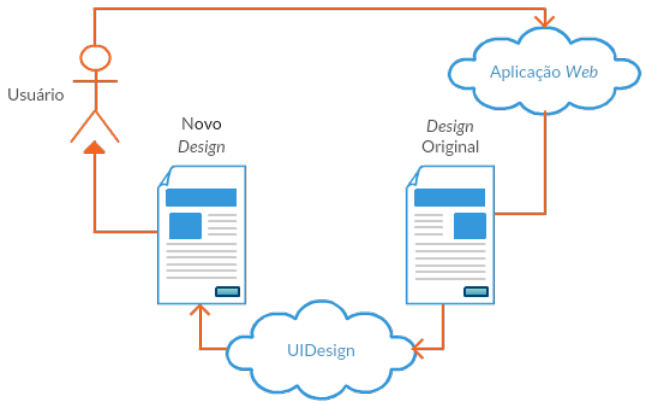

Figura 4.12: A aplicação *web* envia de volta a página após requisição do usuário. Só após UIDesign aplicar as classes de estilo aos elementos da página, modificando o *design*, é que usuário pode interagir com a interface.

A Figura 4.13 mostra com mais detalhes a atuação de UIDesign desde a requisição do usuário à aplicação *web*, passando pela escolha de um *design* para experimentação até a apresentação do mesmo para o usuário.

Aplicações *web* não recebem requisições de apenas um usuário, mas de vários simultaneamente. Já que aplicações *web* recebem requisições simultâneas, UIDesign também lida com isso. Como ilustrado na Figura 4.14, a cada nova requisição que chega, UIDesign responde com um dos *designs* do experimento vigente.

O fato de UIDesign estar desacoplada da aplicação *web* sob experimentação traz algumas vantagens para ambas. Uma delas é a independência das arquiteturas. Isso traz liberdade para UIDesign, que pode ser reutilizada em diversos sistemas *web* sem necessidade de modificações. Além disso, como é a ferramenta que lida com o processo de experimentação, a aplicação sob experimentação pode abstrair todo o processo, necessitando de poucas alterações (Seção 4.3.1). Outra vantagem é que, devido à praticidade de criação e execução dos experimentos, estes podem ser mais frequentes no sistema, o que traz ganhos para o negócio.

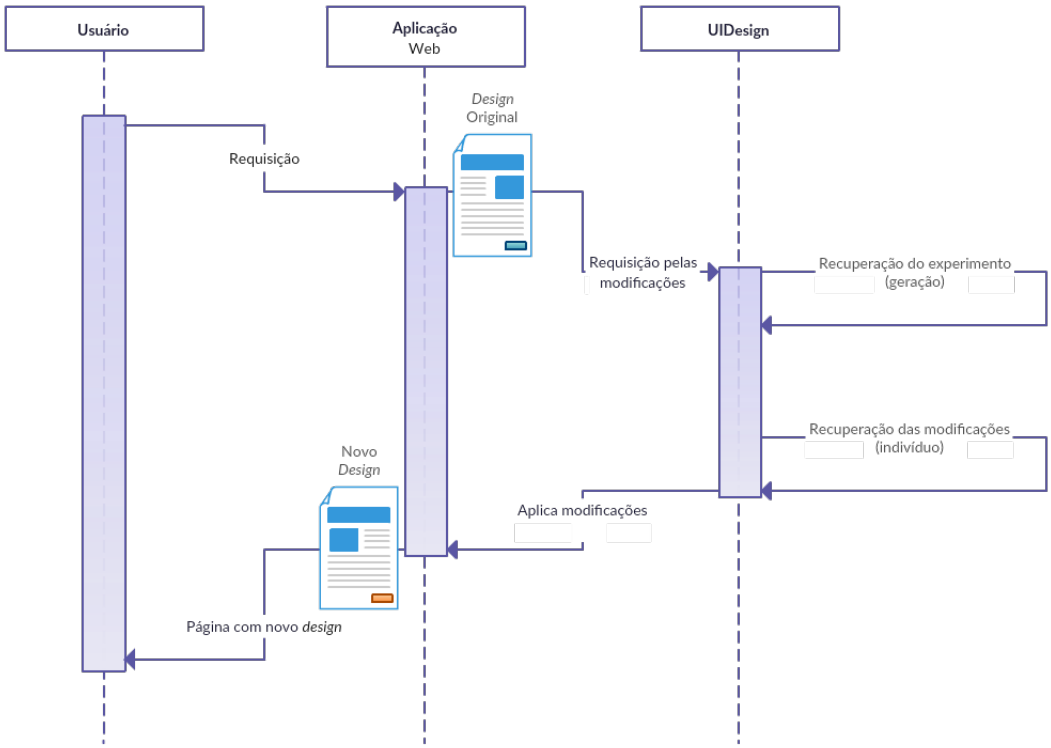

Figura 4.13: Diagrama de sequência da resposta a uma requisição de usuário quando a aplicação *web* está sob experimentação de UIDesign

Todo esse processo de experimentação em que usuários aleatórios recebem uma versão com a qual interagem é embasado no experimento controlado (Seção 2.3). No caso específico de UIDesign, ela utiliza o teste multivariável (Seção 2.3.2) que permite mais de uma alteração na versão em teste e ainda, a experimentação simultânea de múltiplas versões.

Assim que as respectivas alterações são aplicadas à interface original da página, o usuário pode interagir com ela. A partir de então, toda informação relevante para o negócio é coletada por uma ferramenta de *web analytics* (Figura 4.15). Essas informações são as métricas de interesse (Seção 2.2.1), como por exemplo, o número de cliques num certo botão, o tempo de permanência do usuário na página etc, que

dependem da aplicação em experimentação e são definidas na etapa de configuração de UIDesign (Seção 4.3.1).

A ferramenta de *web analytics* armazena esses dados e os associa ao identificador do indivíduo, formado pelo número da sua geração e pelo seu cromossomo. Isso permite identificar, por exemplo, quantos cliques foram feitos num botão numa certa iteração quando ele tinha a cor de fundo azul ou quando a cor de fundo era laranja, por exemplo.

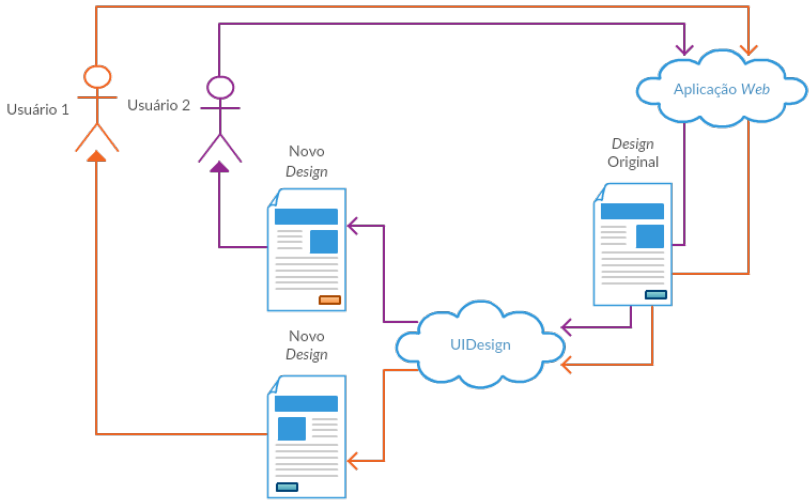

Figura 4.14: UIDesign responde cada requisição de forma independente e pode retornar, inclusive, o mesmo *design* para requisições distintas.

4.3.2.3 Terceira fase: Avaliação dos indivíduos

A terceira fase corresponde à avaliação de cada um dos indivíduos do algoritmo genético. Os dados coletados ao longo da fase anterior são obtidos da ferramenta de *web analytics* e consolidados por indivíduo. A função de adaptação, então, utiliza esses dados para avaliá-los, atribuindo a cada um o seu nível de adaptação (Seção 2.1). Com isso, se o objetivo é obter um aumento no número de cliques

num certo botão, o indivíduo que apresentar o maior valor nesta métrica está mais próximo de ser considerado o melhor.

Figura 4.15: Após receber o novo *design*, o usuário interage com ele e a ferramenta de *web analytics* configurada coleta informações sobre o comportamento dos usuário no *website*

4.3.2.4 Quarta fase: Finalização do experimento

Terminada a avaliação dos indivíduos, a quarta fase tem início. Nesta, o algoritmo genético decide se a solução foi encontrada ou se é necessária nova iteração.

Para a criação de uma nova iteração, o algoritmo genético seleciona alguns indivíduos para o cruzamento através da função de seleção. No caso de UIDesign, a função de seleção utiliza o método da roleta em que os mais aptos tem uma chance maior de serem selecionados. Em seguida, ocorre o cruzamento (*one-point crossover*) e a mutação. A taxa de cruzamento atualmente está fixada em 80% e a de mutação, em 1%. Esses valores foram utilizados com base no algoritmo genético do Matlab [26]. Terminado o processo, um novo experimento com novos *designs* está criado.

5 EXPERIMENTO

O Capítulo 4 apesentou a ferramenta elaborada neste trabalho. No entanto, parte da estratégia de validação da hipótese corresponde à aplicação desta ferramenta a um sistema *web* real. O objetivo deste capítulo é apresentar o experimento conduzido com o auxílio da ferramenta elaborada (Capítulo 4) e analisar os resultados obtidos. A Seção 5.1 apresenta o sistema escolhido bem como seus objetivos, enquanto a Seção 5.2 aborda as configuração de UIDesign e o experimento conduzido. Já a Seção 5.3 encerra o capítulo com a análise dos resultados obtidos.

5.1 Portal Aluno

O Portal Aluno é um sistema *web* que permite aos alunos da UFRJ (Universidade Federal do Rio de Janeiro) acesso a serviços *on-line* oferecidos pela universidade. Alguns destes serviços são:

Inscrição em Disciplinas Neste serviço os alunos selecionam as disciplinas que desejam cursar no período. O acesso ao serviço é sazonal por causa do processo de inscrição em disciplinas, que ocorre em períodos específicos do ano.

Avaliação de Professores Neste serviço, alunos de graduação avaliam os professores das disciplinas cursadas no período anterior. A partir de um questionário, alunos proveem informações como assiduidade, cumprimento da ementa, pontualidade etc.

Emissão de Documentos Neste serviço, alunos selecionam diversos documentos oficiais, tais como boletim e histórico e também certidões como as de regular-

mente matriculado e concluinte. Dos serviços presentes no Portal Aluno, o de emissão de documentos é o segundo mais acessado no ano, perdendo apenas para o de inscrição em disciplinas.

Neste trabalho, será feito um estudo de caso utilizando o Portal Aluno. Por se tratar de um sistema bastante amplo, o serviço para aplicação de UIDesign na otimização de *designs* foi o de emissão de documentos. Além de ser o segundo mais acessado, não é um serviço sazonal. Diferente dos serviços de inscrição em disciplinas e de avaliação de professores que apresentam um período de funcionamento, alunos frequente e regularmente realizam a emissão de documentos.

É importante destacar que a aplicação de UIDesign nos demais serviços também seria possível (inclusive em paralelo). No entanto, neste trabalho, a decisão foi de utilizar a ferramenta em apenas um serviço.

5.1.1 Metas do negócio e métricas de interesse

Atualmente, o principal objetivo do serviço de emissão de documentos, cuja interface aparece na Figura 5.1, é reduzir o tempo médio que o usuário permanece na página. Cada usuário demanda recursos do servidor e quanto antes o usuário seleciona um documento para impressão, mais cedo recursos alocados a ele serão também liberados.

Para otimizar a interface do serviço dando foco às metas de interesse dos administradores do sistema, elas devem ser mapeadas em um conjunto de informações capturáveis por uma ferramenta de *web analytics*. No caso do serviço de emissão de documentos, o melhor *design* deve ser aquele que apresenta o menor valor para o tempo médio dos usuários entre o acesso à página e o clique em um dos ícones.

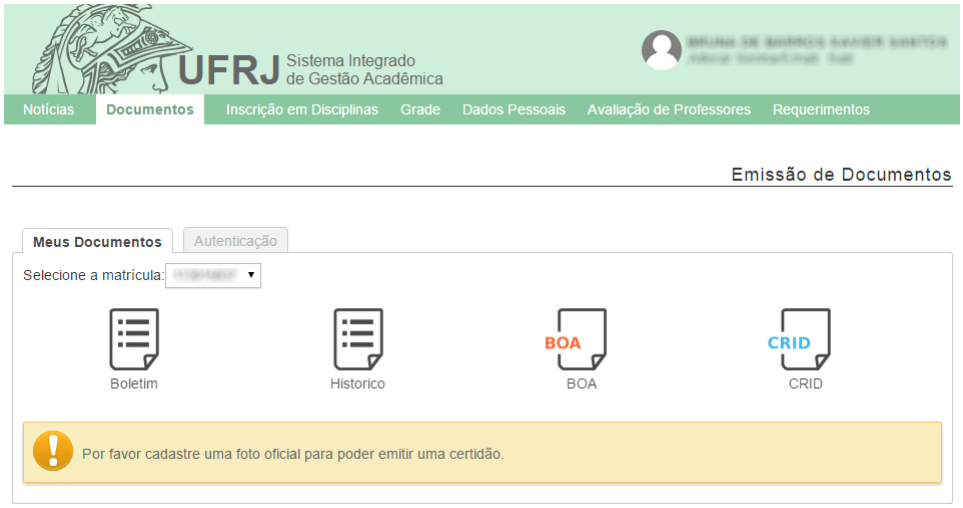

Figura 5.1: *Design* original da emissão de documentos

5.2 UIDesign e o Portal Aluno

Como descrito no Capítulo 4, o objetivo de UIDesign é encontrar um *design* para uma página de outra aplicação *web* que apresenta resultados melhores para o negócio. Para isso, o processo definido por UIDesign é dividido em duas etapas. Na primeira, o usuário administrador fornece os dados do negócio como fonte de informação para a ferramenta. Na segunda, com base nessas informações, UIDesign realiza uma série de experimentos criando e avaliando diferentes *designs*, a fim de obter o melhor.

As seções a seguir mostram a aplicação de UIDesign no Portal Aluno. A Seção 5.2.1 apresenta as informações cadastradas pelo usuário administrador e a Seção 5.2.2, a experimentação propriamente dita.

5.2.1 Etapa de configuração

Este trabalho combina algoritmos genéticos, *web analytics* e teste multivariável para atingir seu objetivo. Essas técnicas necessitam de informações para inicialização e execução, como por exemplo o algoritmo genético que requer a definição dos indivíduos, do método de seleção etc. Alguns desses dados são definidos durante a etapa de configuração (Seção 4.3.1) em forma de parâmetros para UIDesign enquanto outros, atualmente, já estão pré-definidos. A seguir são apresentados cada um dos dados necessários para o funcionamento das técnicas e, consequentemente, do processo de experimentação elaborado para otimizar a página de emissão de documentos do Portal Aluno.

A ferramenta de *web analytics* utilizada foi o Google Analytics (Seção 2.2.2.2) já que o Portal Aluno já a utiliza para analisar outras informações.

Os indivíduos são sempre representados por uma cadeia de bits e codificam uma versão de *design* para experimentação, como descrito na Seção 4.3.2. Já as classes de estilo definidas pelo usuário administrador podem ser vistas no Apêndice A.

O número de indivíduos foi arbitrariamente escolhido como quatorze pelo usuário administrador.

O período de experimentação tem a duração de quatro dias. Isso quer dizer que cada iteração da experimentação (geração do algoritmo genético) coleta e analisa dados de seus usuários por quatro dias para, após este período, avaliar cada uma das versões (indivíduos). Em quatro dias, no serviço de emissão de documentos, é possível obter amostras de cerca de cem usuários para cada indivíduo (versão).

A função de adaptação está atrelada à métrica de interesse. No caso, como o objetivo é minimizar o tempo de permanência do usuário na página, a função de adaptação utiliza essa informação para calcular o nível de adaptação ao indivíduo. Assim, quando cada indivíduo é avaliado, requisições são feitas ao Google Analytics para obter esses valores.

O método de seleção adotado foi o método da roleta (Seção 2.1).

O cruzamento é feito em um único ponto (*one-point crossover*) e com taxa de oitenta porcento, como utilizado pelo Matlab [26].

A mutação tem taxa de um porcento, também como no Matlab [26].

O elitismo não foi utilizado.

5.2.2 Etapa de experimentação

Terminada a etapa de configuração, a etapa de experimentação teve início. Para o primeiro experimento, os indivíduos foram gerados aleatoriamente. Para os demais vinte e nove experimentos, seus indivíduos foram obtidos pelo cruzamento e mutação dos mais aptos. A cada quatro dias, indivíduos do experimento em vigor passaram pelo processo de avaliação, com base nos dados do Google Analytics, em seguida pela seleção, através do método da roleta, pelo cruzamento e mutação.

A Tabela 5.1 mostra os indivíduos da primeira geração e seus respectivos valores para a métrica de interesse a ser otimizada: o tempo médio que usuários levam entre o acesso à página e a seleção de um documento para emissão. Por se tratar da primeira geração, esses indivíduos foram gerados aleatoriamente pelo algoritmo genético, com base nos parâmetros fornecidos pelo usuário administrador na etapa de configuração (Seção 5.2.1).

Tabela 5.1: Indivíduos da primeira geração e o tempo médio dos usuários para selecionar um documento para impressão

Indivíduo	**Tempo médio (segundos)**
000001	98.15
000010	45.75
000110	77.52
000111	69.23
001100	149.60
001101	77.04
010000	81.08
011001	75.62
100011	44.51
101101	46.19
110000	84.03
111000	58.03
111101	64.68
111111	84.44

Como esperado, mudanças na página causaram impacto na percepção dos usuários. Cada conjunto de modificações pode causar um impacto maior ou menor. No entanto, elas afetam a maneira como eles interagem e, por consequência, as métricas de interesse.

Ainda analisando a Tabela 5.1, alguns indivíduos se destacam, como é o caso do indivíduo 100011 (Figura 5.2). Nesse experimento, ele apresentou o melhor tempo de resposta do usuário, garantindo um bom resultado para o negócio. Por outro lado, o pior desempenho foi do indivíduo 001100 (Figura 5.3).

Como característica dos algoritmos genéticos, todos eles tem uma chance de participar na geração do próximo experimento. Como o método de seleção é o da

roleta, indivíduos mais aptos, isto é, aqueles que apresentam menor tempo médio, tem mais chances de contribuir na geração de descendentes.

Figura 5.2: *Design* representado pelo indivíduo 100011

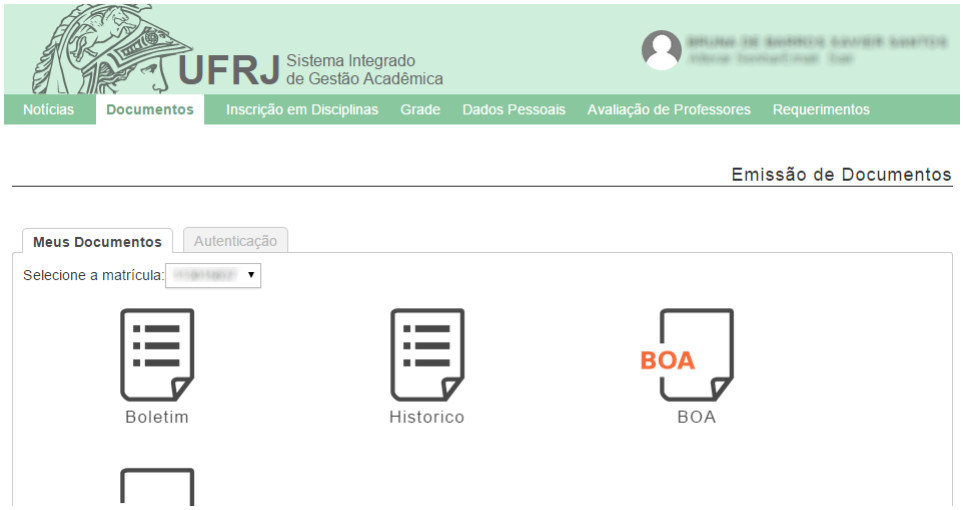

Figura 5.3: *Design* representado pelo indivíduo 001100

5.3 Resultados

A Figura 5.4 apresenta um gráfico com os melhores indivíduos (aqueles que apresentaram o menor tempo médio) para cada uma das gerações e os respectivos valores da métrica de interesse utilizados em sua avaliação.

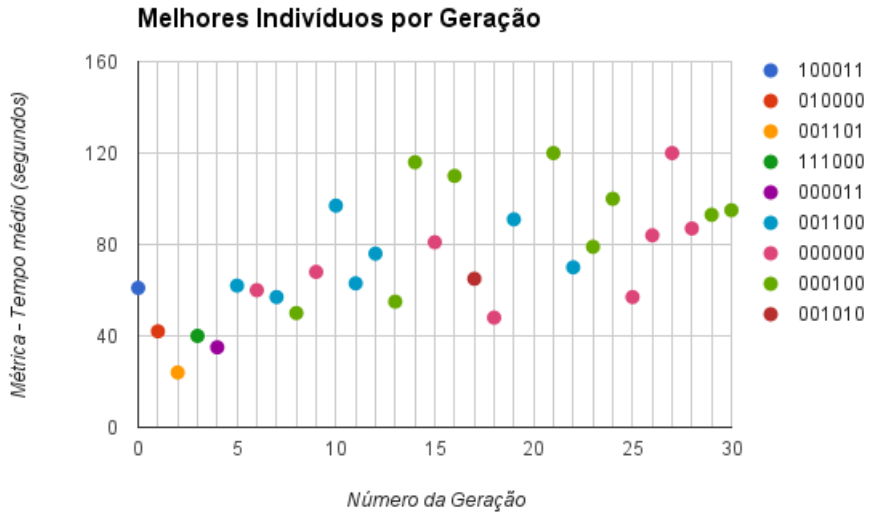

Figura 5.4: Indivíduos com o menor tempo médio encontrados em cada uma das gerações

A partir da Figura 5.4 é possível notar uma variação na métrica de interesse com o passar do tempo. Mais de uma hipótese pode ser formulada para explicar essa variação. Uma possibilidade é o comportamento natural dos usuários do sistema, que varia ao longo do tempo. Essa hipótese é reforçada pela Figura 5.5. Note que um mesmo indivíduo, por exemplo 000000, apresenta diferentes tempos médios ao longo das gerações. Essa variação de comportamento, portanto, é independente do experimento em questão.

Outra possibilidade é a existência de *outliers*. Eles podem estar prejudicando

as medidas feitas pela ferramenta de *web analytics* escolhida pelo usuário administrador (Google Analytics). Como ela trabalha com dados previamente totalizados, não se pode descartar esses valores [13].

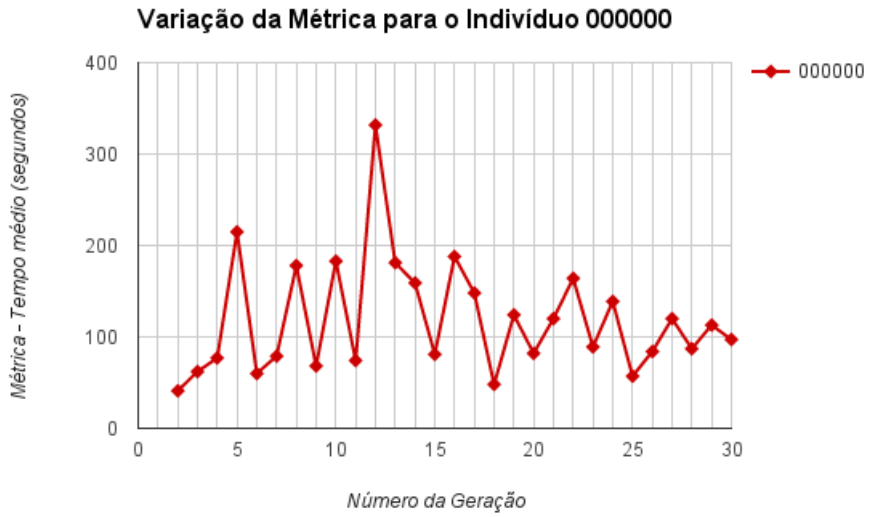

Figura 5.5: Indivíduo 000000 e a variação da métrica decorrente da variação no comportamento dos usuários ao longo do tempo.

Na Figura 5.4 é possível notar também que o indivíduo 001101, dado como o "melhor dos melhores", não se manteve como melhor nas gerações seguintes. Através da Tabela 5.2 vê-se que o indivíduo 001101 aparece nas gerações zero, dois e três mas que somente na geração dois ele se destaca dos demais. A hipótese é de que isso é consequência da variação do tempo médio analisada anteriormente. Com base nisso, acredita-se que a ausência do elitismo no algoritmo genético, nesse caso, teve pouca influência nos resultados uma vez que as variações na métrica influenciam variações nas decisões.

Três indivíduos podem ser destacados, como apresentado na Figura 5.6. O indivíduo 000000 que representa a versão original (ilustrado na Figura 5.1), o indi-

víduo 001100 (Figura 5.3) e o 000100 (Figura 5.7).

Tabela 5.2: Indivíduos 001101 e 111000 e seus respectivos valores para o tempo médio (em segundos) nas primeiras cinco gerações

	001101	**111000**
0	80,60	408,33
1	-	83,88
2	**24,47**	50,86
3	44,28	**40,01**
4	-	-

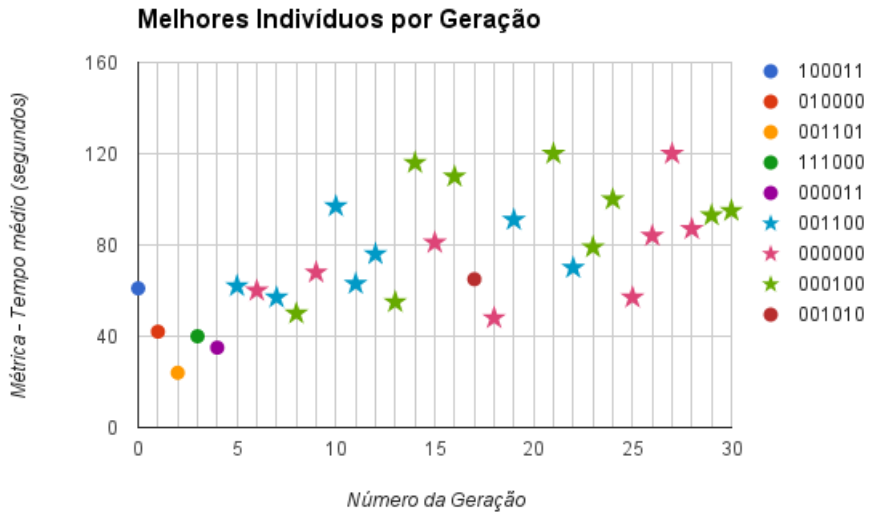

Figura 5.6: Indivíduos com o menor tempo médio encontrados em cada uma das gerações, com destaque para os das últimas gerações

Um fato bastante interessante observado é que esses dois últimos possuem uma característica em comum no *design*: os ícones são maiores que os da versão original. Uma possível explicação para esses indivíduos terem sobrevivido ao longo das iterações e obtido bons resultados está relacionada ao número de documentos que alunos têm para seleção. Como um aluno pode emitir, em média, cinco documentos,

ícones maiores ficam mais destacados, inclusive com o texto maior, o que ajuda a reduzir o tempo até o aluno selecioná-lo.

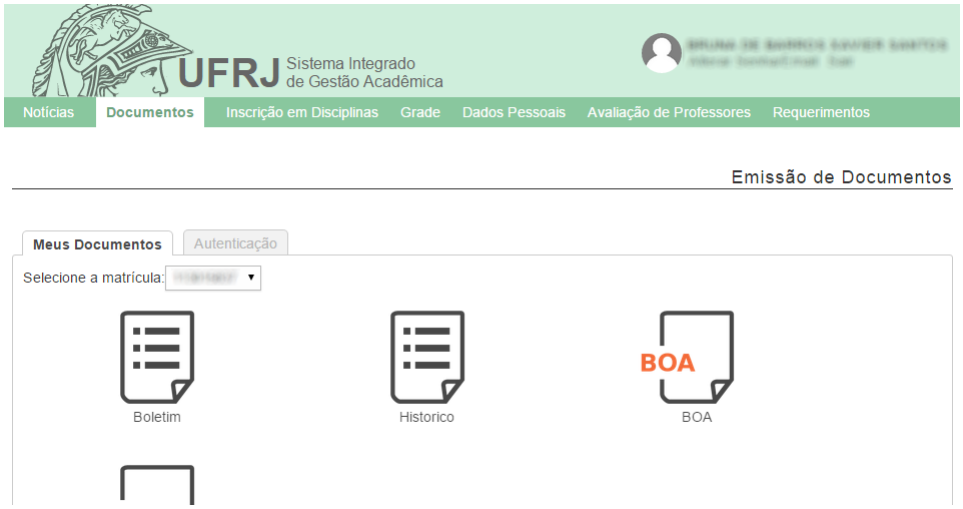

Figura 5.7: *Design* representado pelo indivíduo 000100

Outra possível análise é com relação à importância da experimentação. A Figura 5.8 mostra um gráfico comparativo entre os melhores indivíduos e o *design* original, que nesse experimento, foi representado pelos indivíduos 000000, 000110 e 000111. Note que na maioria das gerações, os indivíduos que obtiveram melhores resultados foram aqueles que apresentaram alguma modificação na interface original. Isso demonstra que não necessariamente o *design* inicialmente definido (no caso, o original) terá o maior apelo frente aos usuários. Caso hajam novas ideias sobre possíveis melhorias, elas devem ser experimentadas pois a menor das mudanças, pode trazer um grande impacto.

A hipótese deste trabalho é a de que é possível encontrar um *design* para uma página *web* de forma automática a partir de um conjunto pré-fixado de estilos e que apresente melhores resultados para o negócio. Esses resultados são observados a partir do comportamento dos usuários com a página que são de interesse para o negócio. Apesar de o comportamento dos usuários variar de geração para geração,

mesmo se observado o mesmo indivíduo, o algoritmo genético se manteve consistente. Isso quer dizer que, apesar de não ser possível descobrir o melhor *design*, o algoritmo genético permite mostrar os indivíduos que trazem melhorias para o negócio, como os destacados na Figura 5.6 e, além disso, identifica o melhor indivíduo em cada uma das gerações. Tudo isso é obtido sem qualquer intervenção humana.

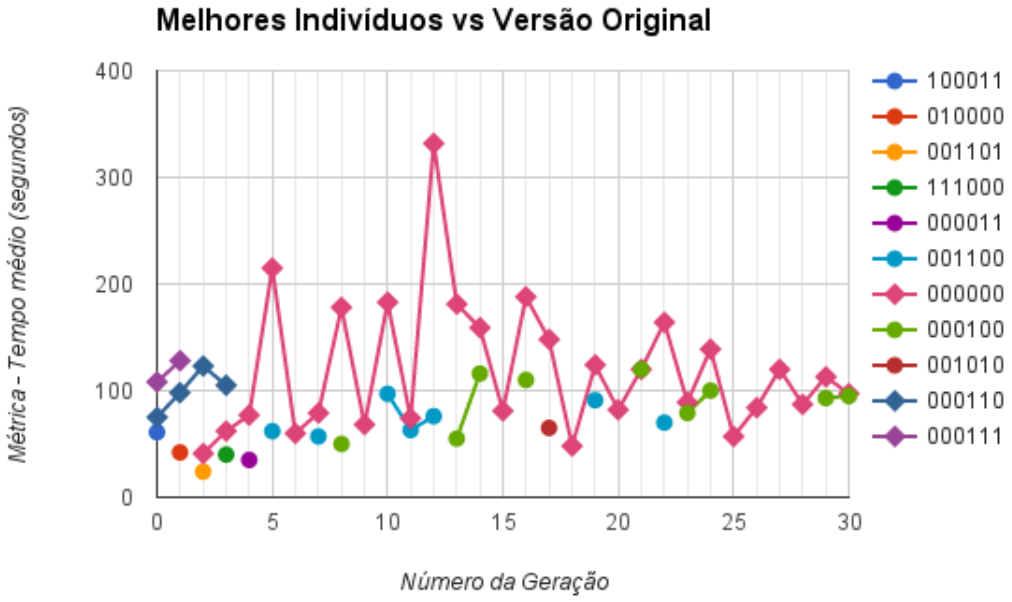

Figura 5.8: Comparativo entre os melhores indivíduos e o *design* original. Os indivíduos que representam o *design* original (losangos) são 000000, 000110 e 000111.

A grande dificuldade encontrada pelo algoritmo genético está em lidar com a variação da função de adaptação. Como ela varia com o tempo, como consequência da variação das métricas, um mesmo indivíduo numa geração pode ser muito bom e na geração seguinte, ruim.

6 CONCLUSÃO E TRABALHOS FUTUROS

Este capítulo apresenta na Seção 6.1 a conclusão do presente trabalho, com base na ferramenta desenvolvida e apresentada no Capítulo 4 e nos resultados observados de sua aplicação num sistema *web* real, vistos no Capítulo 5. Já a Seção 6.2 apresenta possíveis trabalhos que podem ser derivados deste.

6.1 Conclusão

Através da combinação dos algoritmos genéticos, *web analytics* e teste multivariável, este trabalho tenta validar a hipótese de que é possível encontrar uma combinação de estilos, a partir de um conjunto pré-fixado, tal que garante melhores resultados para o negócio, de forma não assistida e através da análise de interações de usuários. Para isso, ele elabora uma ferramenta *web* denominada UIDesign, que combina essas três técnicas, e a aplica em um sistema *web* real: o Portal Aluno [32].

A motivação para a elaboração deste trabalho foi a verificação da existência de outros que visam a geração de páginas *web* mas que apresentam um critério de avaliação extremamente dependente de seres humanos [28, 44, 35]. No seu processo de avaliação, um [28] ou mais usuários [44] escolhem a interface que mais lhes agrada com base em suas opiniões pessoais. No entanto, de acordo com alguns autores [23, 21, 34], esse tipo de avaliação pode gerar interpretações errôneas a respeito das necessidades dos usuários do sistema e podem impactar negativamente nos objetivos do negócio.

Para contornar este problema, a ideia deste trabalho é aplicar sucessivas expe-

rimentações para decidir pelo melhor *design*. A experimentação consiste da criação do *design*, da coleta de informações acerca do comportamento dos usuários e da avaliação de cada um com base nesses dados. Assim, após a criação dos *designs*, usuários "recebem" aleatoriamente uma das versões em teste e interagem com ela. Uma ferramenta de *web analytics* captura informações de relevância para o negócio (como o número de cliques, o tempo de permanência na página etc) que serão utilizadas no fim de um período para avaliar a versão. Essas versões evoluem ao longo do tempo, uma vez que as mais aptas, ou seja, as que obtém os melhores resultados, tendem a perpetuar suas informações.

Este trabalho abre um novo viés para a otimização do *design* de páginas *web*, uma vez que utiliza dados das interações dos usuários para avaliar ideias. Isso permite o desenvolvimento de um processo de experimentação sem supervisionamento. Os *designs* obtidos alinham sistema e objetivos do negócio e podem trazer o diferencial competitivo para empresas. Outra contribuição é a ferramenta (UIDesign) elaborada a partir de ferramentas e técnicas atuais, disponíveis no mundo acadêmico e comercial. Ela já está em funcionamento e pode ser utilizada por qualquer sistema *web*. Além destas, este trabalho originou um artigo [32] que foi publicado recentemente.

Após aplicar UIDesign ao Portal Aluno, resultados interessantes foram obtidos. Apesar de a ferramenta não conseguir identificar o *design* considerado o "melhor dos melhores", devido às variações nos dados capturados dos usuários, o algoritmo genético se manteve consistente e permitiu indicar quais seriam aqueles com mais chances de sucesso para o negócio. A cada iteração, o melhor *design* dentre os avaliados pôde ser identificado e suas características foram propagadas e evoluídas. Além disso, na maioria das iterações, os melhores *designs* foram aqueles que incluíram alguma modificação do *design* original. Com isso, esse trabalho destaca a importância de se experimentar ideias, pois pequenas mudanças podem gerar grande impacto.

6.2 Trabalhos futuros

Um primeiro trabalho consiste em modificar UIDesign, dando à ferramenta mais flexibilidade para permitir uma maior adaptação às preferências do usuário. Isso significa adicionar novos parâmetros à ferramenta, como por exemplo as taxas de cruzamento e mutação e o tempo de duração do experimento, hoje mantidos fixos. Alguns, poderiam depender inclusive de condições do próprio sistema em experimentação, como é o caso do tempo de duração do experimento que poderia variar dependendo do número de amostras coletadas para cada indivíduo. Além disso, o conceito de otimização poderia ser estendido para o conteúdo da página, ao invés de apenas avaliar o estilo dos elementos.

Um segundo trabalho consiste ainda do estudo da variação dos parâmetros de UIDesign. Dados das taxas de cruzamento e mutação bem como o tempo de duração de um experimento, por exemplo, poderiam ser variados. Essas variações poderiam ser comparadas a fim de achar a melhor combinação de parâmetros.

Um terceiro trabalho é a mudança nas técnicas que sustentam UIDesign. Ao invés de combinar algoritmos genéticos e teste multivariável, combinar o *simulated annealing* com o teste A/B, utilizando ainda *web analytics* para extração de dados. Cada novo experimento apresentaria apenas duas versões: a melhor versão encontrada até o momento e essa mesma versão com alguma das modificações propostas para experimentação. O *simulated annealing* em conjunto com dados de *web analytics* seria responsável por gerar e validar a variante. A comparação das duas abordagens também pode trazer resultados significativos.

Outro trabalho futuro consiste ainda da sugestão de modificações para a página, que pode englobar estilo e conteúdo. Essas sugestões podem estar embasadas num aprendizado feito sobre o comportamento dos usuários na página e até num

aprendizado acerca das relações entre os elementos, que podem ser obtidas através do teste multivariável, por exemplo.

Para enriquecer ainda mais os resultados, é possível desenvolver também uma mistura de opinião pessoal e automação. A ferramenta poderia indicar quais as interfaces que apresentaram melhores resultados e o usuário poderia selecionar aquelas que mais fossem do seu agrado.

REFERÊNCIAS

[1] BALUJA, S.; CARUANA, R. Removing the Genetics from the Standard Genetic Algorithm. In: ANNUAL CONFERENCE ON MACHINE LEARNING, 12., 1995, Lake Tahoe. **Proceedings...** San Francisco: Morgan Kaufmann Publishers, 1995. p. 38-46.

[2] BARBOSA, G. **Ferramenta assistida para otimização do design de uma página web baseada no feedback**. 2012. Projeto final de curso (Graduação em Ciência da Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

[3] BOOTH, D.; JANSEN, B. J. A review of methodologies for analyzing websites. In: JANSEN, Bernard J.; SPINK, Amanda; TAKSAI, Isak (Ed.). **Handbook of research on web log analysis**. Hershey: Information Science Reference, 2009. cap. 8, p. 143-164.

[4] MORETTIN, P. A.; BUSSAB, W. de O. **Estatística básica**. 6. ed. rev e atual. São Paulo: Saraiva, 2010.

[5] CCM. Disponível em: <http://ccm.net/contents/158-style-classes-css>. Acesso em: 1 ago. 2015.

[6] DIGITAL ANALYTICS ASSOCIATION. Disponível em: <http://www.digitalanalyticsassociation.org/about>. Acesso em: 4 dez. 2014.

[7] FACEBOOK. Disponível em: <https://developers.facebook.com/docs/platforminsights/page>. Acesso em: 22 mar. 2015.

[8] GNU General Public License. Disponível em: <http://www.gnu.org/copyleft/gpl.html>. Acesso em: 4 dez. 2014.

[9] GOOGLE. Disponível em: <http://www.google.com.br/intl/en_uk/analytics/index.html>. Acesso em: 2 nov. 2015.

[10] GOOGLE. Disponível em: <http://www.google.com/analytics/>. Acesso em: 22 mar. 2015.

[11] GOOGLE. Disponível em: <https://developers.google.com/analytics/devguides/reporting/core/dimsmets>. Acesso em: 2 nov. 2015.

[12] GOOGLE. Disponível em: <https://www.android.com/>. Acesso em: 30 ago. 2015.

[13] GOOGLE. Disponível em: <https://support.google.com/analytics/answer/2637192?csw=1#confidenceInterval>. Acesso em: 2 nov. 2015.

[14] GROUT, I.; A'AIN, A. K. B. Adapting an on-line tutorial tool with web analytics to incorporate analysis of tutorial use. In: INTERNATIONAL CONFERENCE ON INTERACTIVE COLLABORATIVE LEARNING, 15., 2012, Villach. **Proceedings...** New York: IEEE, 2012. p.1-7.

[15] HOLZMANN, C.; HUTFLESZ, P. Multivariate testing of native mobile applications. In: INTERNATIONAL CONFERENCE ON ADVANCES IN MOBILE COMPUTING AND MULTIMEDIA, 12., 2014, Kaohsiung. **Proceedings...** New York: ACM, 2014. p. 85-94.

[16] JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. **Neuro-fuzzy and soft computing**: a computational approach to learning and machine intelligence. Upper Saddle River: Prentice Hall, 1997.

[17] Jennifer Kyrnin. Disponível em: <http://webdesign.about.com/od/css/a/aa020899.htm>. Acesso em: 1 ago. 2015.

[18] KAUSHIK, A. **Web Analytics**: an hour a day. Indianapolis: Sybex, 2007.

[19] KOHAVI, R. et al. Controlled experiments on the web: survey and practical guide. **Data Mining and Knowledge Discovery**, v. 18, n. 1, p. 140-181, 2009.

[20] KOHAVI, R. et al. Trustworthy Online Controlled Experiments: five puzzling outcomes explained. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 18., 2012, Beijing. **Proceedings...** New York: ACM, 2012. p. 786-794.

[21] KOHAVI, R.; HENNE, R. M.; SOMMERFIELD, D. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 13., 2007, San Jose. **Proceedings...** New York: ACM, 2007. p. 959-967.

[22] KOHAVI, R.; LONGBOTHAM, R. Unexpected results in online controlled experiments. **SIGKDD Explorations**, New York, v. 12, n. 2, p. 31-35, 2011.

[23] KUMAR, L.; SINGH, H.; KAUR, R. Web Analytics and Metrics: a survey. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS, 2012, Chennai. **Proceedings...** New York: ACM, 2012. p. 966-971.

[24] LAGUS, K.; KARANTA, I.; YLÄ-JÄÄSKI, J. Paginating the generalized newspapers: a comparison of simulated annealing and a heuristic method. In: PARALLEL PROBLEM SOLVING FROM NATURE, 4., 1996, Berlin. **Proceedings...** Berlin: Springer, 1996. p. 594-603.

[25] LANDIM, T.; ALVES-SOUZA, S. N.; DE SOUZA, L. S. Improving a website with web analytics: a case study. In: IBERIAN CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGIES, 8., 2013, Lisboa. **Proceedings...** New York: IEEE, 2013. p. 1-5.

[26] MATLAB. Disponível em: <http://www.mathworks.com/help/gads/genetic-algorithm-options.html>. Acesso em: 29 out. 2015.

[27] MITCHELL, M. **An introduction to genetic algorithms**. Cambridge: MIT Press, 1996.

[28] OLIVER, A.; MONMARCH, N.; VENTURINI, G. Interactive design of web sites with a genetic algorithm. In: INTERNATIONAL CONFERENCE ON WWW/INTERNET, 2002, Lisboa. **Proceedings** [S.l.]: IADIS, 2002, p. 355-362.

[29] GONZÁLEZ PENALVER, J.; MERELO, J. Optimizing web page layout using an annealed genetic algorithm as client-side script. In: EIBEN, A. E. et al (Ed.). **Parallel Problem Solving from Nature — PPSN V**. Berlin: Springer, 1998. p. 1018-1027. (Lecture Notes in Computer Science, 1498).

[30] PIWIK. Disponível em: <http://piwik.org/docs/piwik-tour/#piwik-overview>. Acesso em: 2 nov. 2015.

[31] PIWIK. Disponível em: <http://piwik.org>. Acesso em: 2 nov. 2015.

[32] SANTOS, B.; CRUZ, A.; TOLEDO, R. de. Automated optimization for web page design based on user interaction analysis. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 17., 2015, Las Vegas. **Proceedings...** Athens: CSREA Press, 2015. p. 799-804.

[33] SINGAL, H.; KOHLI, S.; SHARMA, A. K. Web analytics: state-of-art & literature assessment. In: INTERNATIONAL CONFERENCE - CONFLUENCE THE NEXT GENERATION INFORMATION TECHNOLOGY SUMMIT, 5., 2014, Noida. **Proceedings...** New York: IEEE, 2014. p. 24-29.

[34] SIROKER, D.; KOOMEN, P. **A/B testing**: the most powerful way to turn clicks into customers. Hoboken: Wiley, 2013.

[35] SORN, D.; RIMCHAROEN, S. Web page template design using interactive genetic algorithm. In: INTERNATIONAL COMPUTER SCIENCE AND ENGINEERING CONFERENCE, 2013, Nakorn Pathom. **Proceedings...** New York: IEEE, 2013. p. 201-206.

[36] TANG, D. et al. Overlapping experiment infrastructure: more, better, faster experimentation. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 16., 2010, Washington. **Proceedings...** New York: ACM, 2010. p. 17-26.

[37] VIVEKANANTHAMOORTHY, N. et al. An effective e-learning framework model: a case study. In: INTERNATIONAL CONFERENCE ON ICT AND KNOWLEDGE ENGINEERING, 7., 2009, Bangkok. **Proceedings...** New York: IEEE, 2009. p. 8-14.

[38] W3C HTML. Disponível em: <http://www.w3.org/html/>. Acesso em: 23 jul. 2015.

[39] W3Schools. Disponível em: <http://www.w3schools.com/html/html_elements.asp>. Acesso em: 22 mar. 2015.

[40] W3Schools. Disponível em: <http://www.w3schools.com/html/html_styles.asp>. Acesso em: 22 maio 2015.

[41] W3Schools. Disponível em: <http://www.w3schools.com/cssref/>. Acesso em: 1 nov. 2015.

[42] W3Schools. Disponível em: <http://www.w3schools.com/css/css_selectors.asp>. Acesso em: 1 ago. 2015.

[43] W3Tech. Disponível em: <http://w3techs.com/technologies/overview/traffic_analysis/all>. Acesso em: 23 jul. 2015.

[44] YOKOYAMA, T. et al. Website design system based on an interactive genetic algorithm using tournament evaluation by multiple people. In: INTERNATIONAL CONFERENCE ON SOFT COMPUTING AND INTELLIGENT SYSTEMS, 6., 2012, Kobe and INTERNATIONAL SYMPOSIUM ON ADVANCED INTELLIGENT SYSTEMS, 13., 2012, Kobe. **Proceedings...** New York: IEEE, 2012. p. 2260-2263.

APÊNDICE A CLASSES DE ESTILO

Este apêndice trata das classes de estilo definidas para o Portal Aluno durante a etapa de configuração de UIDesign. Ele apresenta também os possíveis indivíduos do algoritmo genético, dado que sua representação está relacionada às classes e aos elementos aos quais elas podem ser atribuídas.

A Tabela A.1 apresenta os elementos e as possíveis classes de estilo que podem ser atribuídas a eles. Esses dados foram fornecidos pelo usuário administrador durante a etapa de configuração do experimento.

Tabela A.1: Elemento e classes de estilo possíveis para o elemento

Elemento	**Classes de estilo possíveis**
Caixa de seleção de matrícula	destaca-seletor-matricula
Textos dos ícones	titulo-marcado inverte-texto
Ícones	mais-icones-pequenos menos-icones-pequenos mais-icones-grandes menos-icones-grandes menos-icones-muito-grandes

Com base nesses dados, as Tabelas A.2, A.3 e A.4 mostram todos os possíveis indivíduos que podem ser gerados pelo algoritmo genético neste caso e as mudanças que cada um representa. A Figura A.1 mostra um exemplo de indivíduo, com as seções demarcadas.

Tabela A.2: Indivíduos e classes de estilo da caixa de seleção de matrícula

Indivíduo	**Caixa de seleção de matrícula**
0XXXXX	não aplica mudança (mantém original)
1XXXXX	aplica classe destaca-seletor-matricula

Tabela A.3: Indivíduos e classes de estilo do texto dos ícones

Indivíduo	**Texto dos ícones**
X00XXX	não aplica mudança (mantém original)
X01XXX	aplica classe titulo-marcado
X10XXX	aplica classe inverte-texto
X11XXX	não aplica mudança (mantém original)

Tabela A.4: Indivíduos e classes de estilo dos ícones

Indivíduo	**Ícones**
XXX000	não aplica mudança (mantém original)
XXX001	aplica classe mais-icones-pequenos
XXX010	aplica classe menos-icones-pequenos
XXX011	aplica classe mais-icones-grandes
XXX100	aplica classe menos-icones-grandes
XXX101	aplica classe menos-icones-muito-grandes
XXX110	não aplica mudança (mantém original)
XXX111	não aplica mudança (mantém original)

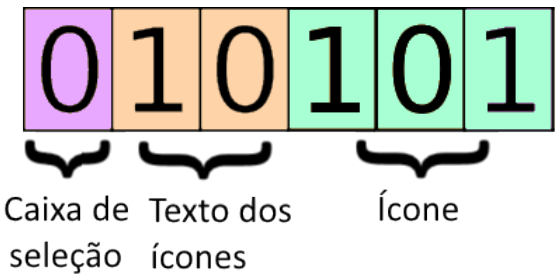

Figura A.1: Exemplo de um possível indivíduo do experimento do Portal Aluno