



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Arquitetura da informação no desenvolvimento de aplicação web

Guilherme Fischmann Ferreira

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Fernando Albuquerque

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Homero Luiz Piccolo

Banca examinadora composta por:

Prof. Fernando Albuquerque (Orientador) — CIC/UnB
Prof.^a Dr.^a Fernanda Lima — CIC/UnB
Prof.^a Dr.^a Genaina Nunes Rodrigues — CIC/UnB

CIP — Catalogação Internacional na Publicação

Ferreira, Guilherme Fischmann.

Arquitetura da informação no desenvolvimento de aplicação web /
Guilherme Fischmann Ferreira. Brasília : UnB, 2015.

64 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. Arquitetura da informação. Engenharia de Software. Aplicação
web. Processo de desenvolvimento de software

CDU 004

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedico a todos os professores que passaram na minha vida dos quais pude tirar os conhecimentos necessários que contribuíram para este e outros trabalhos em minha vida. Dedico a meus Pais e meus amigos que me ajudaram nessa longa jornada.

Agradecimentos

Agradeço primeiramente a Deus por ter me ajudado nessa caminhada, a minha família pelo apoio e ajuda e principalmente a meus professores da universidade, em especial ao professor Fernando Albuquerque por aceitar ser o meu orientador e pela grande ajuda.

Resumo

Em processos de desenvolvimento de software para a *Web*, a falta de atividades voltadas para a arquitetura da informação pode ter impacto negativo sobre a qualidade do software desenvolvido. Este trabalho visa entender a arquitetura da informação no contexto de desenvolvimento de aplicações *web*. Para atingir esse objetivo, um estudo de caso é desenvolvido para colocar em prática processos de desenvolvimento que incluem atividades focadas em arquitetura da informação.

Palavras-chave: Arquitetura da informação. Engenharia de Software. Aplicação *web*. Processo de desenvolvimento de software

Abstract

In software development processes for the Web, lack of activities aimed at information architecture can have a negative impact on the quality of the developed software. This work aims to understand the information architecture in the context of web application development. To achieve this goal, a case study is developed to put into practice development processes that include activities focused on information architecture.

Keywords: Information Architecture. Software Engineering. Web application. Software Development Process

Sumário

1	Introdução	1
1.1	Motivações	1
1.2	Problema	2
1.3	Objetivos de projeto	2
1.4	Estrutura desta monografia	2
2	Arquitetura da informação em aplicações <i>web</i>	4
2.1	Histórico	4
2.2	Definições de arquitetura da informação	5
2.3	Responsabilidades de arquitetos da informação	6
3	Componentes de arquitetura da informação	8
3.1	Introdução	8
3.2	Sistemas de organização	9
3.3	Sistemas de rotulagem	10
3.4	Sistemas de navegação	11
3.5	Sistemas de busca	13
4	Modelos de componentes da arquitetura da informação na <i>Web</i>	16
4.1	<i>UML-based Web Engineering (UWE)</i>	17
4.2	Modelagem do conteúdo	17
4.3	Modelagem da navegação	17
4.4	Modelagem da apresentação	20
5	Abordagem prática: Modelagem	23
5.1	Visão e escopo do sistema	23
5.2	Principais funcionalidades	24
5.2.1	Procurar álbum	25
5.2.2	Comprar álbum	25
5.2.3	Visualizar perfil	25

5.2.4	Avaliar álbum	26
5.2.5	Manter álbuns e usuários	26
5.2.6	Visualizar compras	27
5.2.7	Autenticar usuário	27
5.3	Modelo de conteúdo do sistema	28
5.4	Modelo de navegação do sistema	29
5.5	Modelo de apresentação do sistema	31
6	Abordagem prática: <i>Design</i> e implementação	34
6.1	<i>Oracle Application Express (APEX)</i>	34
6.2	Implantação do sistema	35
6.3	Projeto do banco de dados	36
6.4	Aspectos da implementação	37
6.5	O sistema desenvolvido	40
7	Análise de resultados	46
7.1	Aspectos da Arquitetura da Informação	46
7.2	Aspectos positivos do método <i>UWE</i>	47
7.3	Aspectos negativos do método <i>UWE</i>	47
7.4	Avaliação da utilização do <i>APEX</i> no desenvolvimento do sistema	48
8	Conclusão	50
8.1	Considerações Finais	50
8.2	Trabalhos Futuros	51
	Referências	52

Lista de Figuras

2.1	Um wireframe delineando <i>layout</i> de navegação e <i>design</i> [3].	7
3.1	Estrutura hierárquica [21].	9
3.2	Estrutura polihierárquica [21].	10
3.3	Sistema de navegação embarcado e seus componentes [13].	12
3.4	<i>Website</i> e seus componentes de navegação [8].	12
3.5	<i>Megamenu</i> em <i>website</i> [9].	13
4.1	Modelo de casos de uso e modelo de conteúdo [19].	18
4.2	Modelo de navegação simplificado [19].	18
4.3	Modelo de conteúdo e modelo de navegação. [19].	19
4.4	Página de apresentação de um sistema de revisão [19].	21
5.1	Relacionamento entre atores.	24
5.2	Diagrama de caso de uso para a compra de álbum.	25
5.3	Diagrama de caso de uso para a visualização de perfil.	26
5.4	Diagrama de caso de uso para a avaliação de album.	27
5.5	Diagrama de caso de uso para a manutenção de albuns e usuários.	27
5.6	Esquema organizacional do conteúdo do sistema.	28
5.7	Modelo de conteúdo do sistema.	29
5.8	Modelo de navegação do sistema para clientes e funcionários.	30
5.9	Modelo de navegação do sistema para funcionários.	31
5.10	Exemplos de <i>wireframes</i> do modelo de apresentação do sistema.	32
5.11	Exemplos de <i>wireframes</i> do modelo de apresentação do sistema.	33
6.1	Diagrama de implantação.	35
6.2	Esquema de banco de dados do sistema.	36
6.3	Tela do <i>APEX</i> para edição da página para recarregar créditos.	37
6.4	Telas do <i>APEX</i> para edição de <i>tab sets</i> e <i>tabs</i>	38
6.5	Tela do <i>APEX</i> para edição da região "Mostra créditos".	40
6.6	Tela de pesquisar álbuns.	41

6.7	Tela com álbuns de uma pesquisa.	42
6.8	Tela para editar álbum.	43
6.9	Tela para visualizar informações do álbum.	43
6.10	Tela para visualizar informações do álbum.	44
6.11	Tela para recarregar créditos.	44
6.12	Tela para usuário ver a quantidade de créditos e os álbuns que ele comprou.	45
6.13	Tela para avaliar álbum.	45

Lista de Tabelas

3.1	Sistemas na arquitetura da informação [13].	8
-----	---	---

Capítulo 1

Introdução

Atividades de arquitetura da informação são realizadas no projeto de desenvolvimento de *software* para a *web*. O objetivo dessas atividades é descrever a estrutura de um *website* ou de uma *intranet*. *Websites* e *intranets* são espaços de informação que devem permitir que o usuário encontre as informações necessárias com rapidez e facilidade [13].

A arquitetura de informação é a parte estrutural que direciona a interação do usuário. Antes do usuário poder interagir, no entanto, a informação tem de ser organizada de forma eficaz. Isso é feito dividindo o conteúdo em categorias, usando menus e caminhos para facilitar o acesso ao conteúdo disponibilizado pelo *website* [3]. No entanto, a arquitetura da informação não descreve apenas conceitos, mas o processo de como o espaço foi projetado.

Equipes responsáveis pelo desenvolvimento de aplicações *web* podem conter arquitetos da informação. O Arquiteto de informação tem que a partir de necessidades dos usuários e requisitos de aplicação criar modelos que representam as categorias de conteúdo e a navegação do *website* [8]. Além da criação desses modelos, esses profissionais são tipicamente responsáveis por desenvolver sistemas de organização, sistemas de rotulagem, sistemas de navegação e sistemas de busca empregados em aplicações *web*.

1.1 Motivações

Atividades de arquitetura da informação usadas no desenvolvimento tem impacto positivo na qualidade do *software* desenvolvido. No desenvolvimento de aplicações, ao visualizar modelos de arquitetura da informação, é possível, por exemplo, identificar e resolver problemas antes da implementação do *software* [19].

1.2 Problema

Ao utilizar determinadas aplicações *web*, usuários tem dificuldade para encontrar informações e realizar operações. A interface e a navegação dessas aplicações precisam ser modificadas porque não oferecem suporte para a descoberta de informações. No entanto, essas mudanças podem ser dispendiosas e levar muito tempo.

Para evitar esses problemas, o processo de desenvolvimento de aplicações *web* pode ter metodologias para a criação de artefatos (modelos) de arquitetura da informação. Conhecimentos de arquitetura da informação podem ser utilizados para identificar problemas na navegação e na interface com o usuário.

1.3 Objetivos de projeto

Os objetivos do projeto relatado nesta monografia foram:

- descrever conceitos acerca de arquitetura da informação;
- descrever metodologia para a construção de modelos relevantes à arquitetura da informação;
- utilizar a metodologia para a construção de modelos e utilizar modelos construídos no desenvolvimento de uma aplicação *web*;
- avaliar a metodologia, os modelos criados e a tecnologia utilizada na implementação da aplicação.

1.4 Estrutura desta monografia

Esta monografia está dividida em fundamentação teórica, abordagem prática e conclusão. A fundamentação teórica é composta por quatro capítulos e resultou de pesquisas a diversas fontes de informação. No capítulo 2 são apresentados conceitos de arquitetura da informação, histórico e responsabilidades de arquitetos da informação. No capítulo 3 são descritos componentes de arquitetura da informação em um *website*. No capítulo 4 são descritos os modelos *UWE* (UML-based Web Engineering), os quais especificam os componentes de arquitetura da informação. Também é apresentado no capítulo 4, o método para a construção dos modelos *UWE*.

A descrição da abordagem prática é composta por dois capítulos. No capítulo 5 encontram-se a descrição dos requisitos da aplicação e dos modelos construídos em seu desenvolvimento, enquanto no capítulo 6 são descritos aspectos da implementação. No

capítulo 7 encontram-se análises sobre os modelos, sobre o método utilizado para obter os modelos e sobre a tecnologia usada na implementação da aplicação. Finalmente, no capítulo 8 (conclusão), são apresentadas considerações acerca do trabalho desenvolvido e são sugeridos possíveis trabalhos futuros.

Capítulo 2

Arquitetura da informação em aplicações *web*

Neste capítulo, que tem o intuito de apresentar conceitos acerca de arquitetura da informação em aplicações *web*, é apresentado um histórico do uso do termo "arquitetura da informação", definições possíveis para esse termo e possíveis responsabilidades do arquiteto da informação.

Uma aplicação *web* é qualquer software que é executado em um *web browser*. Ele é criado em uma linguagem de programação suportada por *browser*, como a combinação de *JavaScript*, *HTML* e *CSS*. Aplicações *web* são mantidas e atualizadas sem a necessidade de distribuir e instalar *software* em computadores clientes. Correio eletrônico, vendas de varejo *online* e *wikis* são exemplos de aplicações *web* [1].

2.1 Histórico

Embora a origem do termo "arquitetura da informação" seja atribuída a Richard Wurman, que em conferência do *American Institute of architecture* no ano de 1976 empregou esse termo para descrever a necessidade de transformar dados em informação para as pessoas usarem [5], outros acontecimentos foram também importantes na definição desse termo.

Por exemplo, anos antes, no centro de pesquisa da Xerox em Palo Alto (PARC), um grupo de especialistas desenvolveu tecnologias para apoiar a "arquitetura da informação". Esse grupo foi responsável por contribuições importantes para o que hoje se denomina interação humano computador, como o primeiro computador pessoal com uma interface amigável para o usuário e a impressão a laser [17]. A Xerox contribuiu para a visão da "arquitetura da informação" como um processo de *design* e apresentação de documentos, tendo sido uma das primeiras organizações a manifestar essa noção de estrutura da

informação e a usar o termo "o arquiteto da informação" para definir a sua missão corporativa [22].

Um marco relevante ao surgimento da arquitetura da informação como disciplina ou comunidade de prática foi a realização de uma conferência pela *American Society of Information Science & Technology* (ASIS&T) em maio de 2000 acerca de arquitetura da informação [3]. Conferências acerca de arquitetura da informação dirigidas pela ASIS&T são realizadas anualmente desde então. Também foi relevante o surgimento de grupos profissionais dedicados à arquitetura da informação. Por exemplo, o "The IA Institute". Esse grupo originalmente era denominado *Asimolar Institute of Information Architecture* (AIIA) e foi formado em 2003 com o intuito de promover o campo de arquitetura da informação [3].

A partir da década de 1990, a arquitetura da informação passou a ser relevante no desenvolvimento aplicações *web* devido à necessidade de se organizar a informação acessada digitalmente [3]. Nesse contexto, um marco foi a publicação de um livro sobre o tema por Rosenfeld e Morville em 1998 onde os autores tiveram o intuito de "aplicar princípios de arquitetura e biblioteconomia ao *design* de *websites*" [13]. Atualmente atividades relacionadas à arquitetura da informação tem sido introduzidas no *design* de dispositivos móveis, jogos eletrônicos, entre outros [18].

2.2 Definições de arquitetura da informação

Existem várias definições para o termo "arquitetura da informação". Em [13], Rosenfeld e Morville sugerem as seguintes definições: combinação de organização, rotulagem e esquemas de navegação em um sistema de informação; *design* estrutural de um espaço de informação com o intuito de acessar o conteúdo; arte e ciência de estruturar e classificar *sites* na *Web* (*World Wide Web*) e *intranets* para auxiliar as pessoas a encontrarem e gerenciarem informações; disciplina emergente e comunidade de prática focada em trazer princípios de *design* e arquitetura para o cenário digital. Por sua vez, [2] sugere que "arquitetura da informação é o termo utilizado para descrever o processo de projetar, implementar e avaliar espaços de informação". Para [18], a arquitetura da informação visa facilitar a assimilação da informação e também o entendimento das tarefas feitas pelos usuários.

Alguns membros da comunidade de arquitetura da informação acreditam que seu trabalho é baseado exclusivamente em torno da *Internet*, outros acreditam em aplicações mais amplas da arquitetura da informação. Embora a *Web* seja um domínio de aplicação da arquitetura da informação, o arquiteto da informação pode vê-la como um possível domínio e não uma limitação do seu âmbito profissional [2]. Nesta monografia, a aborda-

gem seguida é a arquitetura da informação em aplicações *web*. Abordagem essa descrita em [13], onde são apresentados métodos que podem ser usado no *design* de *websites* [17].

O crescimento da *Web* transformou-a em um grande ambiente informacional. A sua diversidade de conteúdos, formatos e audiência dificultam a localização da informação [16]. Atender às necessidades de informação dos usuários é um objetivo da arquitetura da informação em aplicações *web*. Nesse contexto, a arquitetura da informação visa a construção de espaços de informação onde os usuários sejam capazes de encontrar informação que desejam de forma eficiente [23].

2.3 Responsabilidades de arquitetos da informação

O arquiteto da informação é responsável pela arquitetura da informação, tipicamente trabalha nas etapas iniciais do desenvolvimento de aplicações interativas usando ferramentas para representar, analisar e avaliar a arquitetura da informação [7]. Geralmente as atividades realizadas por arquitetos da informação estão relacionadas às seguintes áreas de esforço: criação de unidades de conteúdo de informação; construção de associações entre essas unidades; desenvolvimento de funcionalidades de pesquisa e navegação; projeto de gráficos, interfaces e técnicas de interação para permitir que usuários acessem as informações [3]. Além dessas atividades, os arquitetos da informação ilustram passos ou conceitos fundamentais através de gráficos; projetam mapas de *websites*; criam metáforas para identificar conteúdos e promover navegação; desenvolvem estilo e formatação dos elementos de informação; criam cenários e *storyboards*; constroem taxonomias ; testam as experiências dos usuários. A variedade de atividades sugere que o arquiteto de informação, devido a limitações de habilidade ou a restrições de projeto, tipicamente executa apenas algumas dessas atividades. Eles devem compreender e atender usuários, suas necessidades de informação e tarefas realizadas usando *websites* [3, 13]. Considerando que usuário, conteúdo e contexto, e suas interdependências são únicos para cada *website*, o arquiteto procura balanceá-los para que a informação certa seja acessada pela pessoa certa no momento certo.

No desenvolvimento de aplicações *web*, para determinar aspectos visuais dos *websites*, arquitetos da informação podem construir *wireframes* para mostrar como o conteúdo será exibido, localizações de menus, tamanhos de botões, etc. Um *wireframe* tem como objetivo mostrar os requisitos funcionais de um *website*, é um guia visual usado para sugerir a estrutura de um *website* e relacionamentos entre suas páginas [11]. *Wireframe* é uma ilustração dos elementos fundamentais da interface com o usuário, um desenho básico de uma interface. Os *wireframes* de um *website* devem retratar a arquitetura de informação do *website* [11].

A figura 2.1 é um exemplo de *wireframe*. As interfaces também podem ser criadas usando *toolkits* de interação visual. Pode-se construir e testar funcionalidades da aplicação, como a seleção de item ou formulários de pesquisa. Existe um corpo significativo de literatura sobre a interação com *websites* vindo do campo da interação homem-computador, o qual pode ser empregado no processo de arquitetura da informação [7].

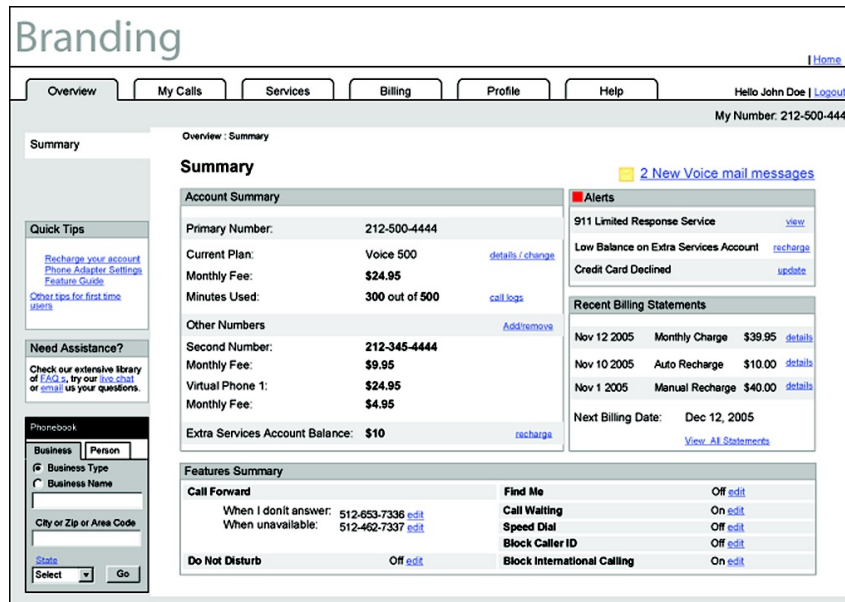


Figura 2.1: Um wireframe delineando *layout* de navegação e *design* [3].

Capítulo 3

Componentes de arquitetura da informação

Neste capítulo, que tem o intuito de descrever componentes integrantes da arquitetura da informação em aplicações *web*, são descritos conceitos acerca de sistema de organização, sistema de rotulagem, sistema de navegação e sistema de busca.

3.1 Introdução

A arquitetura da informação de um *website*, segundo [13], é constituída pelos seguintes componentes interdependentes: sistema de organização, sistema de navegação, sistema de rotulagem e sistema de busca. Esses componentes são elementos de interação do usuário com um *website*. A tabela 3.1 relaciona definições simplificadas dos componentes mencionados.

A divisão da arquitetura da informação nesses quatro sistemas é conceitual e destina-se a organizar o trabalho do arquiteto da informação [16]. As seções seguintes descrevem esses sistemas.

Tabela 3.1: Sistemas na arquitetura da informação [13].

Sistemas	Definição
Sistema de organização	Define o agrupamento e a categorização do conteúdo.
Sistema de navegação	Especifica as maneiras de navegar, de se mover pelo espaço.
Sistema de rotulagem	Estabelece as formas de representação, de apresentação, da informação.
Sistema de busca	Determina as pesquisas que o usuário pode fazer.

3.2 Sistemas de organização

É necessário organizar a informação para que os usuários sejam capazes de recuperá-la mais tarde, de forma eficiente. Sistemas de organização, juntamente com sistemas de rotulagem, são usados no projeto dos sistemas de navegação e busca de um *website*. Sistemas de navegação são constituídos por esquemas de organização e estruturas organizacionais [13]. Um esquema de organização é formado por grupos de conteúdo (grupos lógicos), esses grupos possuem itens de conteúdo (elementos) com características comuns. Esses esquemas podem ser exatos, um elemento só pode ser membro de um grupo lógico exclusivo, por exemplo, uma cidade relaciona-se exatamente a um município, ou podem ser ambíguos [21]. Esquemas de organização ambíguos podem ser por tema, por tarefa, por público ou por metáfora. Nesses esquemas, um elemento pode pertencer a mais de um grupo lógico.

Uma estrutura organizacional define como o esquema de organização está organizado. As estruturas organizacionais não necessariamente tem uma mesma forma, diferentes estruturas organizacionais representando diferentes perspectivas sobre um esquema de organização podem ser obtidas [21]. A estrutura organizacional mostra como os grupos lógicos se relacionam, os grupos podem estar organizados, por exemplo, de forma hierárquica, ou lista.

A estruturas hierárquica e polihierárquica são estruturas organizacionais muito utilizadas. A figura 3.1 mostra uma estrutura hierárquica. Nesse tipo de estrutura, todos nós filhos são acessíveis apenas através de seu nó pai. A restrição é a estrita relação de 1:n entre os grupos de um esquema de organização. Cada grupo pode ter várias subgrupos e cada subgrupo tem um único grupo pai. Por exemplo, "professores"(grupo pai) tem os subgrupos "professor adjunto" e "professor substituto".

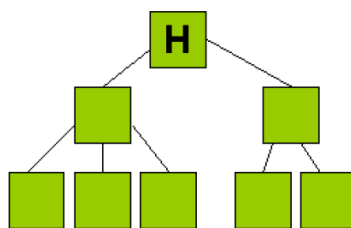


Figura 3.1: Estrutura hierárquica [21].

Uma estrutura polihierárquica (figura 3.2) é uma variação de uma estrutura hierárquica. A restrição 1:n nas relações entre nós é substituída por um relação n:m. Nesse tipo de relação, um nó pode ter vários nós filhos e um nó filho pode ter várias nós pai. Como consequência, um subgrupo pode pertencer a diferentes grupos. Um *website* que

usa a estrutura polihierárquica necessita informar ao usuário em que contexto se encontra um item de conteúdo, informar o grupo pai ao qual o item está relacionado [21].

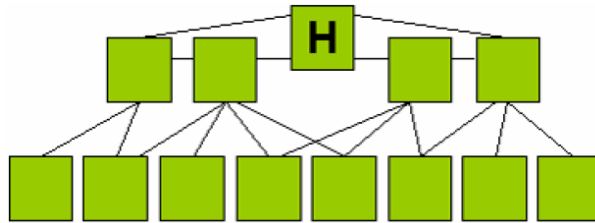


Figura 3.2: Estrutura polihierárquica [21].

O sistema de organização é afetado pela perspectiva do seu criador, sua cultura e sua visão de mundo [16]. Um fato que aumenta a complexidade do sistema de organização é que diferentes usuários têm diferentes perspectivas, o arquiteto da informação precisa evitar que suas perspectivas pessoais influenciem a organização da informação [13].

3.3 Sistemas de rotulagem

Rótulos são identificadores usados para representar informações [13]. Um sistema de rotulagem provê uma representação visual do esquema de organização, é composto por um conjunto de rótulos em um espaço de informação. Os rótulos podem ser textuais ou iconográficos, textual, quando composto por uma ou mais palavras, ou não textual, quando composto de imagens ou sons.

Um exemplo de rótulo textual são títulos de páginas de *websites*, que informam o conteúdo presente nessas páginas. Rótulos iconográficos usam o aspecto visual de um ícone ou um gráfico para representar o conteúdo. Os ícones representam funções mais complexas e expressam mais conteúdo do que o texto, mostram mais informação ao usuário sobre o conteúdo presente na página representada pelo rótulo. Porém, o usuário pode ter uma noção equivocada sobre o conteúdo representado pelo ícone. Isso é um problema de interpretação, esse problema também ocorre se vocabulários utilizados para rótulos textuais são jargões, termos organizacionais especializados não utilizados pelos usuários [13].

Na concepção de sistemas de rotulagem é preciso estar ciente de diferentes linguagens e percepções, os rótulos devem ser conhecidos pelos usuários [13]. A rotulagem do sistema pode ser inconsistente, o *website* pode ter rótulos diferentes para *links* que direcionam para uma mesma página *web*. Para evitar essas inconsistências, tabelas podem ser utilizadas para listar os *links* de navegação com os seus respectivos rótulos.

O processo de *design* de sistemas de rotulagem é complexo pois esses sistemas devem comunicar conceitos sem ocupar muito espaço na página e sem demandar muito esforço cognitivo dos usuários para a compreensão desses conceitos [13].

3.4 Sistemas de navegação

Sistema de navegação de um *website* é uma coleção de componentes de interface com o usuário. O principal objetivo da navegação é ajudar os usuários a encontrar informações e funcionalidades, e incentivá-los a tomar ações desejáveis [8].

O sistema de navegação é responsável por tornar o conteúdo acessível ao público alvo. Para atingir esse objetivo, emprega os recursos visuais do sistema de organização e rotulagem. Segundo [13], esses sistemas devem contextualizar e dispor de caminhos complementares para se encontrar o conteúdo e completar tarefas. Sistemas de navegação são compostos por navegação embarcada e navegação utilitária.

A navegação utilitária pode consistir de mapas do *website*, guias ou assistentes com informações sobre a localização do conteúdo no *website* e também pode consistir de *links* para conteúdo relevante ao público alvo do *website*.

Navegação embarcada aparece nas páginas de *websites* em conjunto com o conteúdo da página, indicando onde o usuário está, onde esteve e onde pode ir a partir da página *web*. A navegação embarcada consiste de navegação global, navegação local, navegação contextual.

A navegação global (menu principal) são *links* para as páginas que podem ser acessadas a partir de qualquer página do *website*, permite ao usuário ir para as principais áreas do *site*.

A navegação local (submenu) é um conjunto de páginas. Uma página desse conjunto pode ser acessada a partir de qualquer outra página que pertença ao conjunto. As páginas de uma navegação local são acessadas a partir de um único membro da navegação global (menu principal). Um *website* é formado por uma navegação global (menu principal) que possui navegações locais (submenus).

A navegação contextual de uma página é o relacionamentos entre páginas. Consiste em *links*, mostra as páginas (caminho) que o usuário passou para chegar até a página que está no momento [13]. A figura 3.3 mostra como um sistema de navegação embarcado está organizado.

Determinar o componente de navegação mais utilizado é importante. Por exemplo, os usuários navegam pelo *website* principalmente usando a navegação local e não os *links* da navegação contextual;

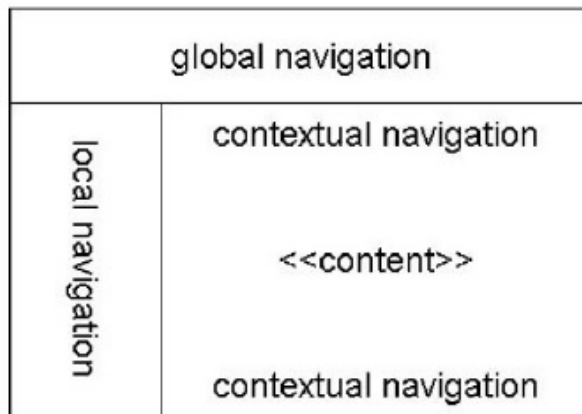


Figura 3.3: Sistema de navegação embarcado e seus componentes [13].

A figura 3.4 mostra alguns exemplos de componentes de navegação: 1) navegação utilitária; 2) navegação global; 3) navegação contextual; 4) navegação local.

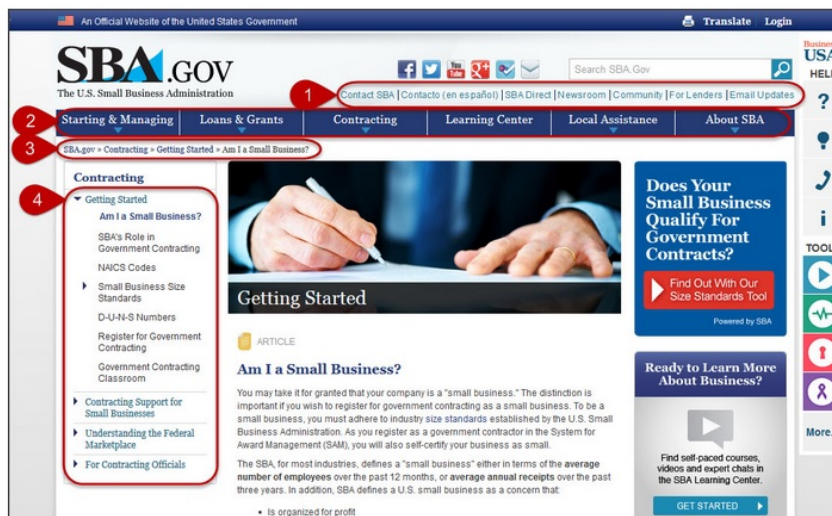


Figura 3.4: Website e seus componentes de navegação [8].

Para cada componente de navegação, as seguintes decisões devem ser tomadas [8]:

- Determinar as páginas que o componente deve estar presente e onde o mesmo deve ser colocado dentro do *layout* de uma página. (por exemplo, canto superior, do lado esquerdo, do lado direito, em baixo);
- Determinar os padrões de projeto de navegação que oferecem melhor suporte para a descoberta de informações, por exemplo, *tabs*, *megamenus* ou outras opções.

Megamenus são grandes painéis bidimensionais divididos em grupos de opções de navegação. As opções de navegação são estruturadas por meio de tipografia e ícones e são visíveis de uma só vez [9]. A figura 3.5 mostra um *megamenu* presente em um *website*.

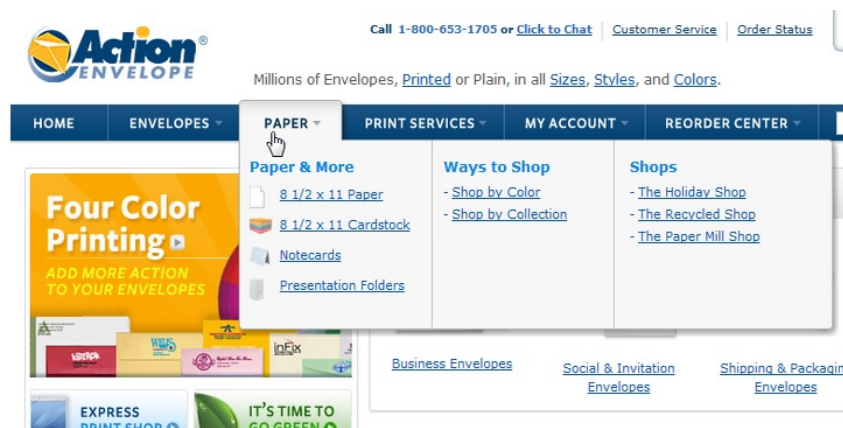


Figura 3.5: *Megamenu* em *website* [9].

Em [13] são relacionados os seguintes princípios de bons sistemas de navegação: o usuário sempre deve saber onde está no *website*; o usuário deve saber para onde ir a partir de uma página; informar ao usuário as páginas onde já esteve; ser óbvio o que fazer para chegar em uma página; indicar o que acontecerá ao selecionar um *link*.

3.5 Sistemas de busca

Sistema de busca é o componente da arquitetura da informação que possibilita o usuário encontrar a informação que precisa. Geralmente, a busca em um *website* consiste de caixas para a entrada de dados (*string* de pesquisa) e botões que ao serem selecionados mostram resultado da consulta, os itens de conteúdo relacionados à *string* informada pelo usuário [20].

O usuário pode ter acesso ao conteúdo utilizando o sistema de navegação definido na seção anterior ou utilizando o sistema de busca. A partir da busca que o usuário terá de forma mais direta a possibilidade de satisfazer suas necessidades informacionais, terá acesso ao conteúdo que deseja sem precisar passar por uma ou mais páginas do *website*.

Possibilitar a busca de conteúdos é um recurso desejável em *websites*. A busca é útil nos seguintes casos [21]: *websites* com grande quantidade de informação; *websites* nos quais os conteúdos são frequentemente modificados.

Um sistema de busca não é uma solução para melhorar a navegação em um *website*, não vai informar ao usuário onde ele está e nem como encontrar a informação [21].

A busca em *websites* normalmente é usada quando o usuário tem noção razoável do que procura. A navegação é utilizada em outros contextos, quando o usuário não tem uma necessidade de informação claramente definida ou tem uma noção vaga sobre o conteúdo. Nesse caso, o usuário utiliza a navegação para ter um maior conhecimento sobre o conteúdo. Como exemplo, uma pessoa utiliza a busca de um *website* para obter um determinado artigo e caso deseja saber sobre uma área de conhecimento, utiliza a navegação para ir na seção do *website* com artigos dessa área.

Sistemas de busca usam algoritmos de indexação, metadados e vocabulários controlados para recuperar a informação procurada pelos usuários [21].

Metadado é definido como os dados que fornecem informação sobre um ou mais aspectos do dado. No caso do sistema de busca, metadado é a informação sobre itens de conteúdo de um grupo do esquema de organização. Esses tipos de informação são chamados de "campos de um metadado" [15]. Como exemplo, campos do metadado de um artigo são atributos como autor, palavras-chave, língua, data de publicação.

Um vocabulário controlado é uma lista restrita de palavras ou termos normalmente utilizados para catalogação descritiva, etiquetagem ou indexação [6]. Um possível objetivo de um vocabulário controlado é relacionar os termos pertencentes a um campo de metadado. Um vocabulário controlado pode ser estruturado de maneira que possua termos preferidos e termos alternativos. Quando uma busca é feita, o sistema de busca sugere itens de conteúdo ou faz traduções do termo informado pelo usuário para o termo preferido. O sistema faz isso usando as associações entre termos preferidos a termos alternativos. O vocabulário é controlado porque somente sob certas condições específicas e processos de revisão, os termos dentro de um vocabulário controlado podem mudar ou outros termos podem ser adicionados. Isso é da responsabilidade de um editor de vocabulário controlado ou taxonomista, não dos usuários. Vocabulários controlados podem ter estrutura de lista de termos, taxonomia hierárquica e tesouros [6].

Listas de termos são frequentemente utilizadas como elementos de metadados. Considerando formato de arquivo como um campo de metadado, possíveis formatos de arquivo é um exemplo de lista de termos. Vocabulários controlados eventualmente podem ser muito grandes e complexos para serem representados por simples listas de termos. Nesses casos, podem ser utilizados outras estruturas.

Uma taxonomia hierárquica é um vocabulário controlado em que todos os termos

pertencem a uma única estrutura hierárquica e tem relacionamentos do tipo pai/filho ou do tipo mais amplos/mais específico com outros termos [6]. A estrutura é geralmente referida como uma "árvore".

Um tesouro para a recuperação do conteúdo é uma estrutura que lista termos semelhantes para cada termo de entrada de um vocabulário controlado. Todos os termos associados (sinônimos) podem ser usados no lugar do termo de entrada. Os sinônimos ou quase sinônimos devem ser equivalentes em todas as circunstâncias. Um tesouro de recuperação de conteúdo é mais estruturado do que outros tipos de vocabulários controlados, pois fornece informações sobre cada termo e seus relacionamentos com outros termos dentro do tesouro [6]. Além de especificar os termos que podem ser usados como sinônimos, um tesouro também indica quais termos são mais específicos, quais são mais amplos e que termos não são hierarquicamente relacionados.

Sistemas de busca não utilizam apenas a busca por palavras-chave (*keywords*) para encontrar informação. Procuram entender o significado e contexto da *string* de pesquisa e podem considerar relacionamentos entre os termos de um vocabulário controlado. Um vocabulário controlado pode trazer informação sobre sinônimos, relacionamentos hierárquicos e relacionamento entre elementos. Ao considerar esses relacionamentos, sistemas de busca podem entender como resultado de uma consulta não só as correspondentes sintáticas exatas da *string* de pesquisa [4].

Capítulo 4

Modelos de componentes da arquitetura da informação na *Web*

No capítulo anterior foram definidos os seguintes componentes da arquitetura da informação: sistemas de organização, navegação, rotulagem e busca. Existem metodologias para a implementação desses componentes em aplicações *web* [13]. A metodologia de desenvolvimento de aplicações *web* baseada em modelos é uma alternativa [19]. Modelos são usados pelos desenvolvedores na implementação, assim como também são produto da atividade de modelagem, são um meio de se chegar ao produto final. A partir de métodos para modelar aplicações, são obtidos o modelo de conteúdo, o modelo de navegação e o modelo de apresentação, os quais são usados para a implementação do sistema de organização, navegação e rotulagem de um *website*. O uso de modelos visa evitar problemas como o não cumprimento de requisitos [19]. Neste capítulo, os modelos descritos são:

- modelo de conteúdo;
- modelo de navegação;
- modelo de apresentação.

O modelo de conteúdo visa mostrar os tipos de conteúdo, apresentados por um *website*. O modelo de navegação representa o sistema de navegação, as possibilidades de navegação com base no conteúdo; o modelo de apresentação representa a rotulagem, mapeia estruturas de navegação para páginas e suas ligações, assim, representa a interface com o usuário [19]. Esse capítulo também apresenta métodos para a construção desses modelos. A *UML-based Web Engineering (UWE)* é utilizada para a modelagem.

4.1 *UML-based Web Engineering (UWE)*

UWE é um método de engenharia *Web* baseado em *UML* usado para a especificação de aplicações *web* [19]. O *UWE* usa modelos para representar a navegação e apresentação de aplicações *web*, usa notação e diagramas *UML* para análise e *design* de aplicações. Para as características específicas de aplicações *web*, como nós e ligações da estrutura de navegação, *UWE* usa estereótipos, *tags* e restrições definidas para os elementos de modelagem [19].

Os requisitos de uma aplicação *web* são o ponto de partida para a construção dos modelos. O conteúdo, a navegação e a apresentação da aplicação são dependentes dos requisitos. O modelo de conteúdo pode ser obtido a partir de um modelo de casos de uso ou de uma descrição de alto nível da aplicação *web*. Esse modelo é usado para o desenvolvimento do modelo de navegação, o qual é usado para a criação do modelo de apresentação [19].

4.2 Modelagem do conteúdo

O modelo de conteúdo apresenta os tipos de conteúdo do *website* e a relação entres esses tipos [19]. O modelo de conteúdo compreende os aspectos estruturais do conteúdo, por exemplo, sob a forma de um diagrama de classe.

O conteúdo procurado pelo usuário ao acessar uma aplicação *Web* pode ser composto por textos, imagens, videos, etc. O conteúdo é dividido em categorias de conteúdo, as quais estão representadas no modelo de conteúdo por classes [19]. Na modelagem do conteúdo, é preciso pensar nos conjuntos de (classes) da aplicação *web*, seus atributos e relacionamento entre essas classes. A figura 4.1 mostra modelo de casos de uso e modelo de conteúdo de uma aplicação *web*.

4.3 Modelagem da navegação

O objetivo do modelo de navegação é apresentar a navegabilidade baseada no conteúdo de uma aplicação *web*, ou seja, os caminhos de navegação à disposição de usuários. Esse modelo define as classes do modelo de conteúdo que podem ser visitadas pela navegação, os elementos de acesso.

A estrutura da navegação é formada por nós, também chamados de páginas ou documentos, e ligações entre esses nós. O ponto de partida utilizado para a criação de um modelo de navegação é o modelo de conteúdo, que contém as classes que podem ser nós da navegação. O modelo de navegação é também chamado de visão de navegação [19].

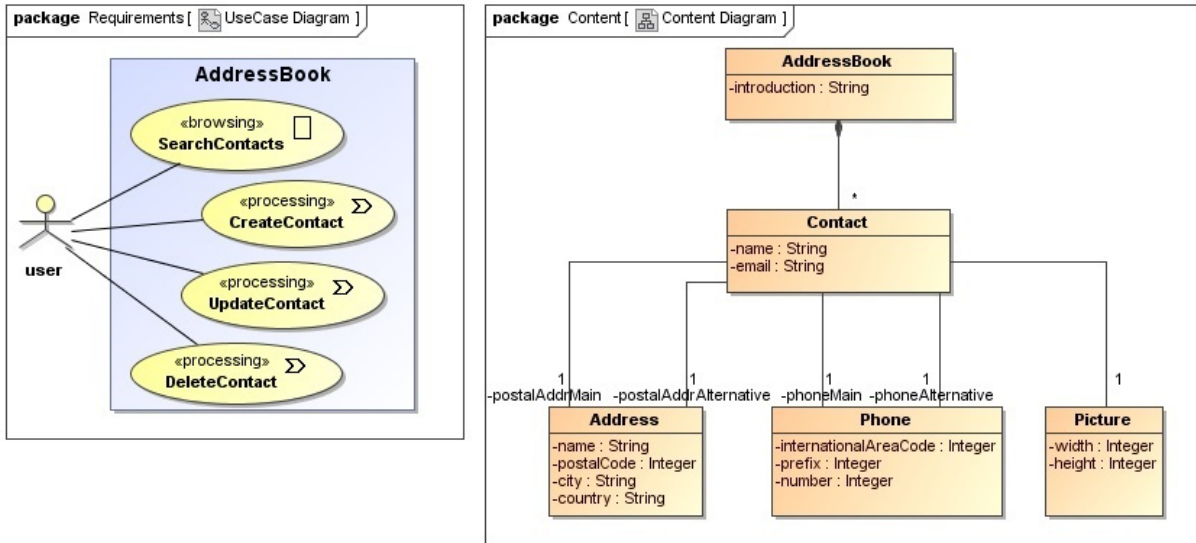


Figura 4.1: Modelo de casos de uso e modelo de conteúdo [19].

Podem ser obtidas visões de navegação personalizadas levando em consideração os direitos de acesso de diferentes usuários em um sistema.

Para que os usuários acessem os nós (páginas), eles precisam da navegação. São formuladas estruturas de acesso, um índice é uma estrutura de acesso que permite aos usuários selecionar um recurso de informação, ou seja, um item de um tipo de conteúdo, por exemplo, um artigo entre os artigos do sistema. Em contraste, um menu é uma estrutura de acesso que permite aos usuários acessar nós heterogêneos, por exemplo, artigos, autores, revisores ou outros menus. Outra estrutura de acesso é a consulta, a qual permite aos usuários procurar recursos de informação, por exemplo procurar artigos [19]. A figura 4.2 mostra um modelo de navegação de um sistema de revisão de artigos.

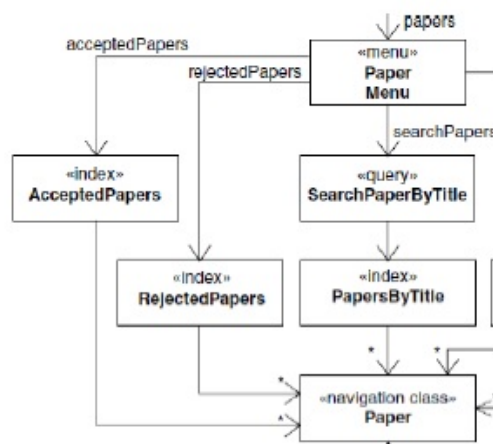


Figura 4.2: Modelo de navegação simplificado [19].

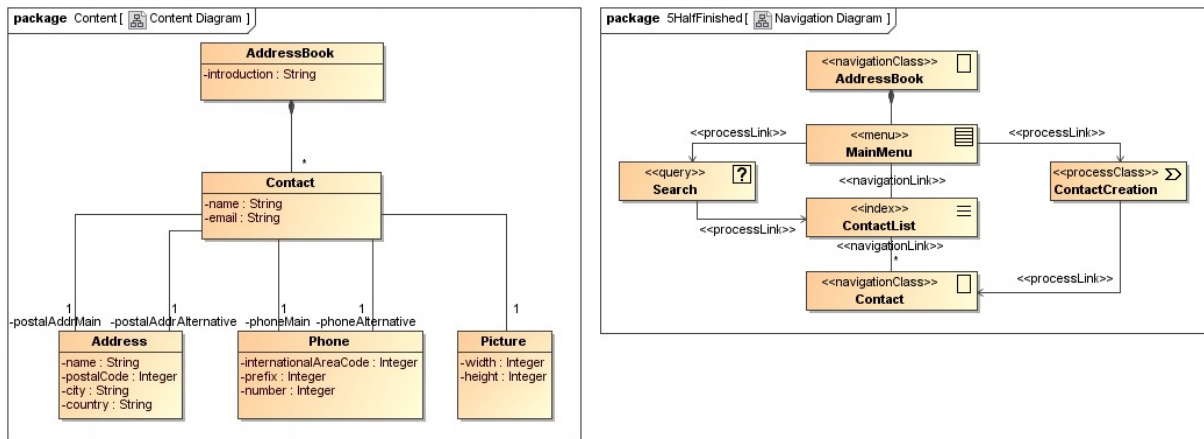


Figura 4.3: Modelo de conteúdo e modelo de navegação. [19].

Atividades definidas são usadas para obter o modelo de navegação a partir do modelo de conteúdo. A figura 4.3 mostra o modelo de conteúdo e o modelo de navegação obtido a partir dele. Para cada classe do modelo de conteúdo existe uma classe de navegação correspondente no modelo de navegação. O modelo de navegação mostra as classes do modelo de conteúdo (tipos de conteúdo) que podem ser acessadas através de uma outra classe, mostra os caminhos de acesso para chegar ao conteúdo especificado pelas classes do modelo de conteúdo. Modelos de navegação possuem classes com os estereótipos *navigation class*, *index*, *menu*, *query* e *process class*. Essas classes são chamadas estruturas de acesso [19]. Uma classe *navigation class* é uma classe do modelo de conteúdo representada no modelo de navegação. Algumas das classes do modelo de navegação estão descritas a seguir:

- uma classe *index* está sempre ligada a uma *navigation class*, representa muitos objetos dessa classe;
- uma classe *menu* pode está ligada a qualquer tipo de classe (*navigation class*, *index*, etc.);
- uma classe *query* está normalmente ligada a um *menu*, representa uma busca que retorna objetos de uma *navigation class*;
- as classes de processo (*process class*) representam operações da aplicação *web*, tais como adicionar, excluir ou modificar os objetos de uma *navegation class* do modelo de navegação.

Na modelagem da navegação, as primeiras atividades são criar uma classe do tipo *menu* e criar uma classe de navegação para cada classe presente no modelo de conteúdo.

Deve-se analisar os requisitos e o modelo de casos de uso para identificar as principais operações da aplicação envolvendo as classes de navegação criadas. Caso exista a funcionalidade de procurar itens de uma categoria de conteúdo (objetos de uma *navigation class*), é preciso ligar uma classe do tipo *query* ao menu criado e criar uma classe do tipo *index* e ligá-la entre a *query* e a *navigation class* [19]. Nesse caso, a classe *index* representa os objetos da *navigation class* que são resultado de uma busca. Uma *navigation class* geralmente esta ligada a uma classe do tipo *menu* ou a uma classe *index*. Se existir a operação de criar um objeto da *navigation class*, a *navigation class* é ligada à uma *process class*.

Em seguida, deve-se identificar as operações envolvendo as *navigation classes*. São criados outros *menus* para representar as navegações locais. Classes do tipo *process class*, que representam as operações, são geralmente ligadas às *navigation classes*. Quando objetos de uma classe fazem parte de outra classe (relacionamento um-para-muitos ou um-para-um), classes do tipo *index* ou outras *navigation classes* também podem estar ligadas à uma *navigation class* [19].

4.4 Modelagem da apresentação

O modelo de navegação não mostra quais classes de navegação e classes de processo pertencem a qual parte da página de uma aplicação *web*. Um diagrama de apresentação é usado a fim de fornecer essa informação.

O modelo de apresentação é representado por mais de um diagrama UML. As propriedades que estão contidas por composição são apresentadas nesses diagramas como retângulos que estão contidos na figura da classe que contém essas propriedades [19].

A modelagem da apresentação visa projetar a estrutura e o comportamento da interface com o usuário. O modelo de apresentação possui elementos presentes nas páginas, por exemplo, cabeçalhos e rodapés e mostra a composição de cada página e o *design* dos campos e textos nessas páginas.

Além da estrutura das páginas, o modelo de apresentação descreve os aspectos comportamentais da interface com o usuário, por exemplo, qual botão selecionar para ativar uma função da aplicação. Devido à variedade de opções de navegação, é preciso dar aos usuários a ajuda e orientação adequada sobre o nível de apresentação. Isso pode ser obtido, exibindo o caminho de navegação, as páginas visitadas durante a sessão ativa [19].

A figura 4.4 mostra uma página do modelo de apresentação de um *website*. O modelo apresenta classes com certos estereótipos. As classes do tipo *text* são textos descrevendo algum atributo, por exemplo, a data de envio do artigo, título, resumo; classes do tipo *anchor*, ao serem selecionadas, direcionam o usuário para outra página. Classes do tipo

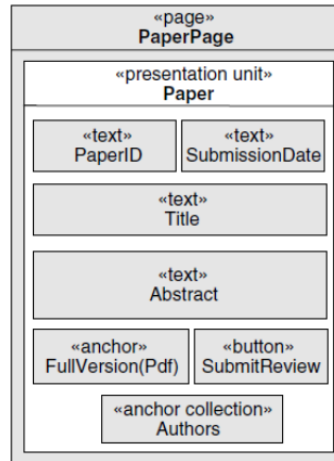


Figura 4.4: Página de apresentação de um sistema de revisão [19].

button ativam uma função da aplicação, por exemplo, "enviar revisão". Uma classe do tipo *presentation group* é um *wireframe*, representa o *layout* das páginas de uma aplicação *web*. *Presentation groups* e classes de outros tipos podem estar contidas em uma *presentation group*. Algumas classes do modelo de apresentação estão descritas a seguir:

- uma *presentation page* (página principal) é a página que possui as operações do *menu* principal do modelo de navegação, além disso, possui *presentation groups* e *links* para *presentation groups* da aplicação;
- uma *text* representa o texto presente em alguma página;
- uma *anchor* representa um *link* para uma *presentation group*;
- uma *button* representa uma operação presente em uma *presentation group* ou *presentation page*;
- uma *inputForm* representa um formulário para a entrada de dados, essas classes estão presentes em *presentation groups* ou na *presentation page*.

O modelo de apresentação é construído a partir do modelo de conteúdo e do modelo de navegação. Na modelagem da apresentação, as primeiras atividades a serem feitas são a criação de uma *presentation page* e a criação de uma *presentation group* para cada *navigation class* do modelo de navegação. Deve-se observar os atributos da classe equivalente à *presentation group* no modelo de conteúdo e criar, para cada um dos atributos, elementos do tipo *text* ou *image* na *presentation group*. Se uma *process class* estiver ligada a um *menu* do modelo de navegação, criar um *button* na *presentation group*. O usuário selecionará o *button* para realizar a operação definida pela *process class*.

Caso haja uma classe do tipo *query* ligada a um *menu*, é preciso criar na *presentation group* um *inputForm* com um *textInput* e com um *button*. O *textInput* é o texto (parâmetro de pesquisa) informado pelo usuário. Ao selecionar o *button*, o resultado da consulta aparece. No modelo de navegação, os resultados da busca são representados por uma classe *index* ligada à uma classe *query*. No modelo de apresentação, são representados por uma lista de *anchors* (relatório) com links para *presentation groups* que representam a *navigation class* do *index*. Para determinar outros elementos de uma *presentation group* é preciso identificar os tipos de classes ligadas à sua *navigation class* correspondente.

Capítulo 5

Abordagem prática: Modelagem

Neste capítulo será apresentada a abordagem prática realizada para estudo do processo de arquitetura da informação no desenvolvimento de aplicação *web*. O objetivo é o desenvolvimento de componentes de arquitetura da informação (conteúdo, navegação, apresentação, busca) para um sistema de compras de álbuns musicais. Os modelos vistos nos capítulos anteriores foram utilizados para especificar e documentar os componentes de arquitetura da informação. As navegações, buscas e interfaces (telas) do sistema precisam ter boa qualidade. Para alcançar esse objetivo, conhecimentos de Arquitetura da Informação foram utilizados na modelagem do sistema.

5.1 Visão e escopo do sistema

Os objetivos do sistema a ser desenvolvido são permitir a compra de álbuns musicais e manter o conteúdo de uma loja de música virtual. Álbuns musicais, músicas, usuários e avaliações feitas por usuários são alguns tipos de conteúdo da loja. Os usuário devem ter facilidade para encontrar o conteúdo e facilidade para realizar as operações do sistema. Os atores envolvidos nesse sistema são dos seguintes tipos:

- clientes;
- funcionários da loja.

Os clientes são quaisquer usuários capazes de comprar e avaliar álbuns. Os funcionários da loja desempenham o papel de administradores do sistema. O conteúdo do sistema pode ser visualizado, incluído, excluído e modificado por eles. Esse conteúdo pode ser álbuns, usuários, avaliações de álbuns ou compras realizadas. Exemplos de operações feitas por funcionários são mudar preço de álbum, adicionar funcionário como administrador e excluir álbum. A figura 5.1 mostra os atores do sistema.

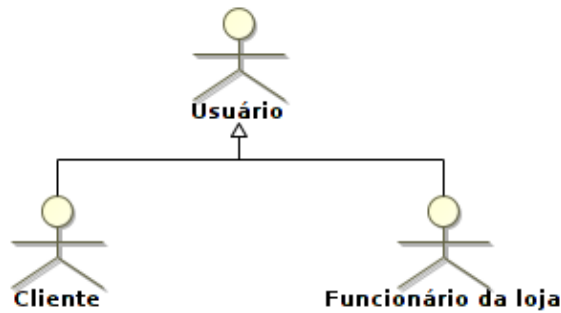


Figura 5.1: Relacionamento entre atores.

A solução proposta é que, utilizando os componentes da arquitetura da informação na *web*, construa-se um sistema em que usuários tenham facilidade em encontrar álbuns, músicas, compras realizadas e outros usuários. Além disso, o sistema deve ser flexível para a inserção de novos álbuns, novos usuários e novas avaliações.

Alguns requisitos não funcionais que deverão ser providos pelo sistema são: deverá existir autenticação dos usuários provendo integridade e confiança para utilização do sistema; será provido tratamento de erros na execução das operações.

5.2 Principais funcionalidades

As principais funcionalidades do Sistema de compra de álbuns musicais, são as seguintes:

- procurar álbum;
- comprar álbum;
- visualizar perfil;
- avaliar álbum;
- manter álbuns e usuários;
- visualizar compras;
- autenticar usuário.

Cada funcionalidade será detalhada para a compreensão de como será utilizado o sistema.

5.2.1 Procurar álbum

Cada cliente pode procurar por álbuns pelo seu nome, nome do artista, estilo musical e ano de lançamento. O resultado da pesquisa é apresentado como uma lista de álbuns. Um cliente pode ver as informações detalhadas de um álbum. Essas informações são o título do álbum, o nome do artista, a lista de músicas, avaliações do album e preço do álbum.

5.2.2 Comprar álbum

Ao acessar a página com os detalhes de um álbum, um cliente pode comprar o álbum. Caso o cliente já tiver comprado o álbum em uma outra ocasião, pode fazer o *download* do álbum. Não é possível comprar um subconjunto das músicas de um album, álbuns são comprados com todas as suas músicas.

Cada cliente registrado no sistema tem crédito, o qual é usado para comprar álbuns. O crédito pode ser recarregado por pagamento. Para isso, o cliente tem que inserir seus dados de pagamento e a quantidade que quer recarregar. Esses dados são validados e o cliente tem que confirmar a transação antes que seja cobrado e que o crédito seja recarregado. O diagrama de caso de uso 5.2 mostra os casos de uso no contexto dessas funcionalidades.

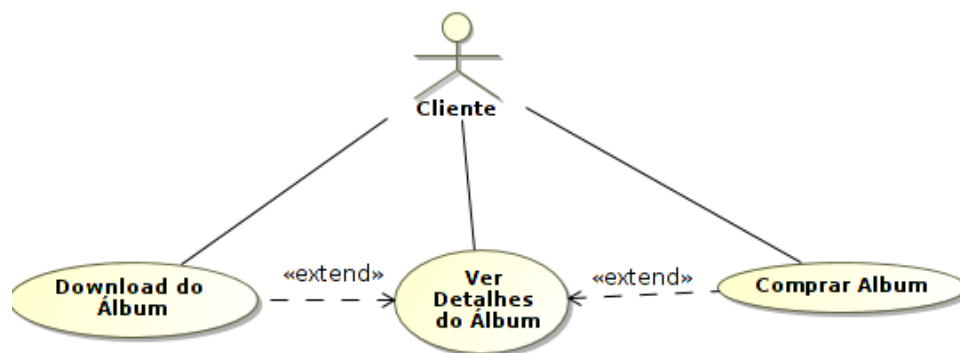


Figura 5.2: Diagrama de caso de uso para a compra de álbum.

5.2.3 Visualizar perfil

Um cliente pode navegar até a página do seu perfil, que mostra seus dados pessoais (nome, e-mail e quantidade de créditos). Nessa página, ele pode ver álbuns comprados por ele e informações desses álbuns.

Um funcionário da loja pode ver o perfil de todos os clientes e também pode ver os álbuns comprados por eles. O diagrama de caso de uso 5.3 mostra o relacionamento entre atores e os casos de uso no contexto dessas funcionalidades

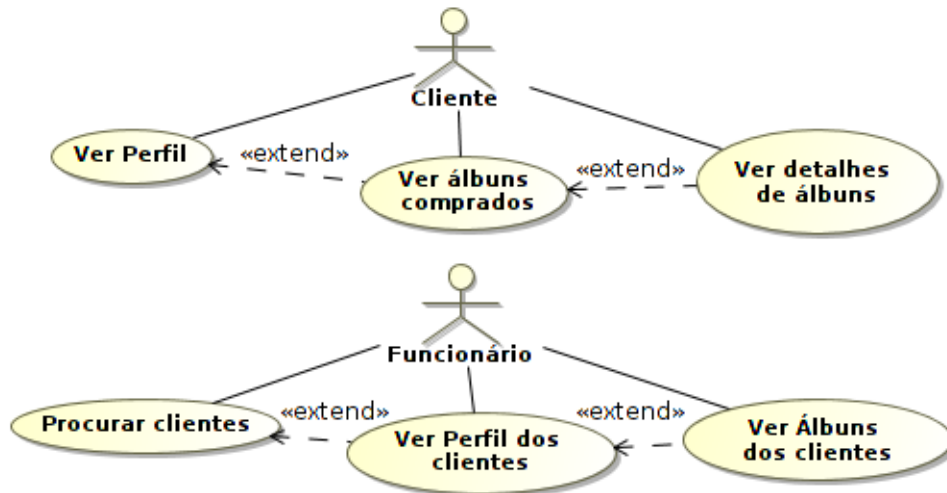


Figura 5.3: Diagrama de caso de uso para a visualização de perfil.

5.2.4 Avaliar álbum

Um cliente pode avaliar os álbuns comprados por ele, os quais aparecem na página do seu perfil. Na avaliação, o usuário informa o seu grau de satisfação com o álbum e pode escrever um comentário.

O usuário (cliente ou funcionário) pode ver avaliações dos álbuns do sistema. O usuário faz a busca e o sistema retorna o resultado da consulta, que é uma lista de zero ou mais álbuns. O usuário pode ver as avaliações dos álbuns dessa lista. O diagrama de caso de uso 5.4 mostra o relacionamento entre atores e os casos de uso no contexto dessas funcionalidades.

5.2.5 Manter álbuns e usuários

Funcionários registrados no sistema, além de procurarem por álbuns e ver suas informações, podem mudar preço dos álbuns, excluir e adicionar álbuns. Eles também podem procurar, excluir e incluir clientes e funcionários. O diagrama de caso de uso 5.5 mostra os casos de uso no contexto dessas funcionalidades.

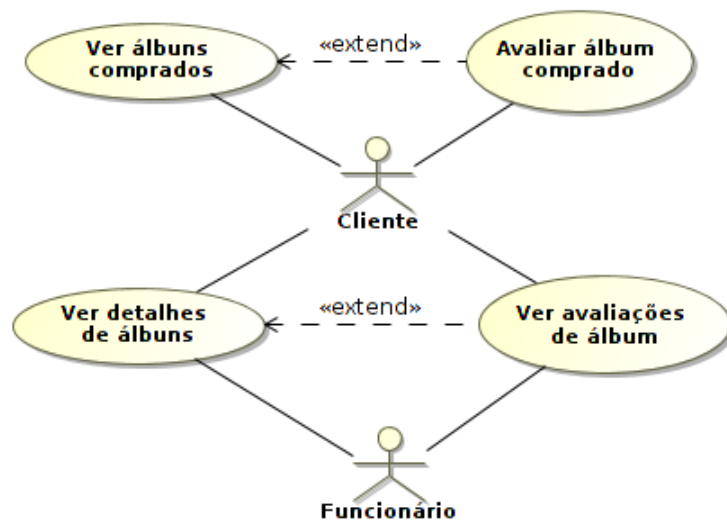


Figura 5.4: Diagrama de caso de uso para a avaliação de álbum.

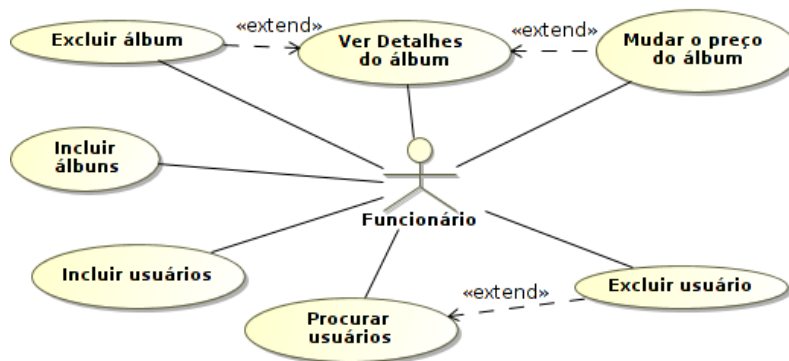


Figura 5.5: Diagrama de caso de uso para a manutenção de álbuns e usuários.

5.2.6 Visualizar compras

Funcionários da loja podem visualizar as compras feitas por quaisquer usuários.

5.2.7 Autenticar usuário

Somente usuários registrados podem utilizar o sistema. Clientes não registrados podem se cadastrar com um nome de usuário que não tenha sido tomado por outro cliente e uma senha livremente escolhida.

5.3 Modelo de conteúdo do sistema

Como visto nos capítulos anteriores, a modelagem do conteúdo visa analisar os requisitos e as funcionalidades do sistema para identificar os grupos (tipos) de conteúdo e as relações entre esses grupos. Identificou-se os seguintes grupos de conteúdo: Compra; Avaliação; Álbum; Usuário; Música. Identificou-se as relações entre os grupos de conteúdo. Tanto Compra quanto Avaliação estão relacionadas a Álbum e Usuário. Álbum está relacionado a Música.

Após identificar os grupos de conteúdo e a relação entre eles, foi desenvolvido um esquema organizacional. A figura 5.6 mostra o esquema organizacional com os grupos de conteúdo do sistema.

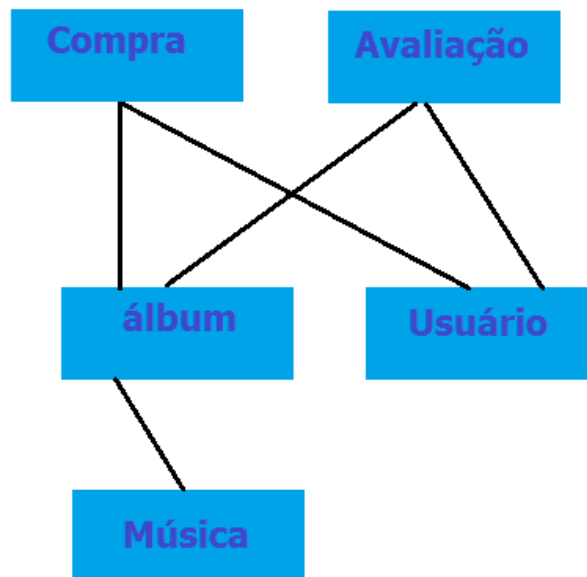


Figura 5.6: Esquema organizacional do conteúdo do sistema.

O modelo *UWE* de conteúdo visto no capítulo 5 é um refinamento do esquema de organização, traz maiores detalhes sobre os grupos de conteúdo e sobre as relações entre os grupos. Nesse modelo, os grupos de conteúdo são classes. São definidos os atributos de cada classe e a multiplicidade das relações entre as classes. Após o refinamento do esquema de organização criou-se o modelo *UWE* de conteúdo (figura 5.7) para o sistema de compra de álbuns musicais.

O modelo mostra os atributos e os tipos dos atributos das classes. Além disso, traz mais detalhes sobre as relações entre classes. Mostra que uma ou mais músicas devem estar vinculadas a um álbum. Uma compra deve estar vinculada a álbum e usuário. Os

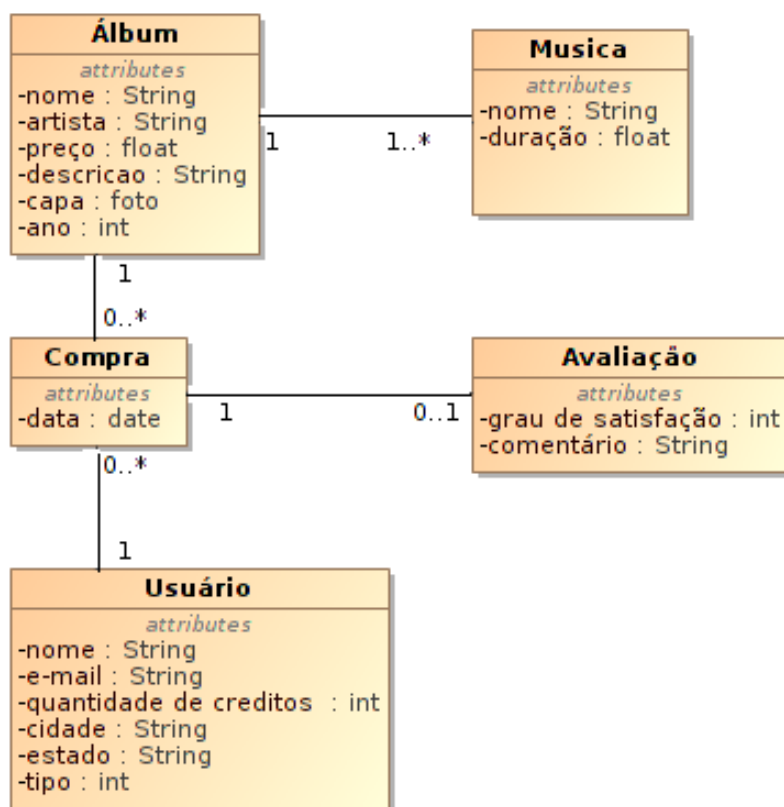


Figura 5.7: Modelo de conteúdo do sistema.

usuários podem fazer várias compras, cada compra possui um álbum e não é possível o mesmo usuário comprar o mesmo álbum mais de uma vez.

Sabendo as classes relacionadas no modelo de conteúdo, chega-se na etapa de construir o modelo de navegação para determinar como os grupos de conteúdo, representados por classes, serão acessados.

5.4 Modelo de navegação do sistema

Para a construção do modelo *UWE* de navegação, deve-se analisar os casos de uso, usar o modelo *UWE* de conteúdo e o método definido no capítulo anterior. A identificação da navegação global e identificação das navegações locais é feita durante a construção do modelo. No modelo de navegação, os *menus* ligados ao *menu* principal pertencem a navegação global, os elementos ligados a um *menu* que não é o principal fazem parte de uma navegação local.

De acordo com os requisitos, os funcionários da loja visualizam o conteúdo visto pelos clientes, porém, o conteúdo visto pelos clientes da loja não é o mesmo visto pelos funcio-

nários. O cliente não pode, por exemplo, ver as compras realizadas no sistema. Portanto, o sistema tem dois perfis de acesso. É preciso construir um modelo de navegação para os usuários (clientes ou funcionários) e construir um modelo de navegação apenas para funcionários.

A Figura 5.8 é o modelo de navegação do sistema que mostra as possibilidades de navegação que clientes e funcionários possuem em comum. O modelo mostra que a partir do *menu* principal o usuário pode acessar o *menu* da sua conta e o *menu* de álbuns, esses *menus* fazem parte da navegação global, estão presentes em todas as páginas (telas) do sistema. A partir do *menu* de sua conta o usuário pode visualizar seu perfil (suas informações pessoais) e pode visualizar os álbuns comprados por ele. O modelo também mostra que o usuário pode recarregar os créditos a partir da página do seu perfil e que a partir do *menu* de álbuns o usuário pode procurar álbuns musicais. A partir da página com informações do álbum, é possível avaliar ou comprar o álbum.

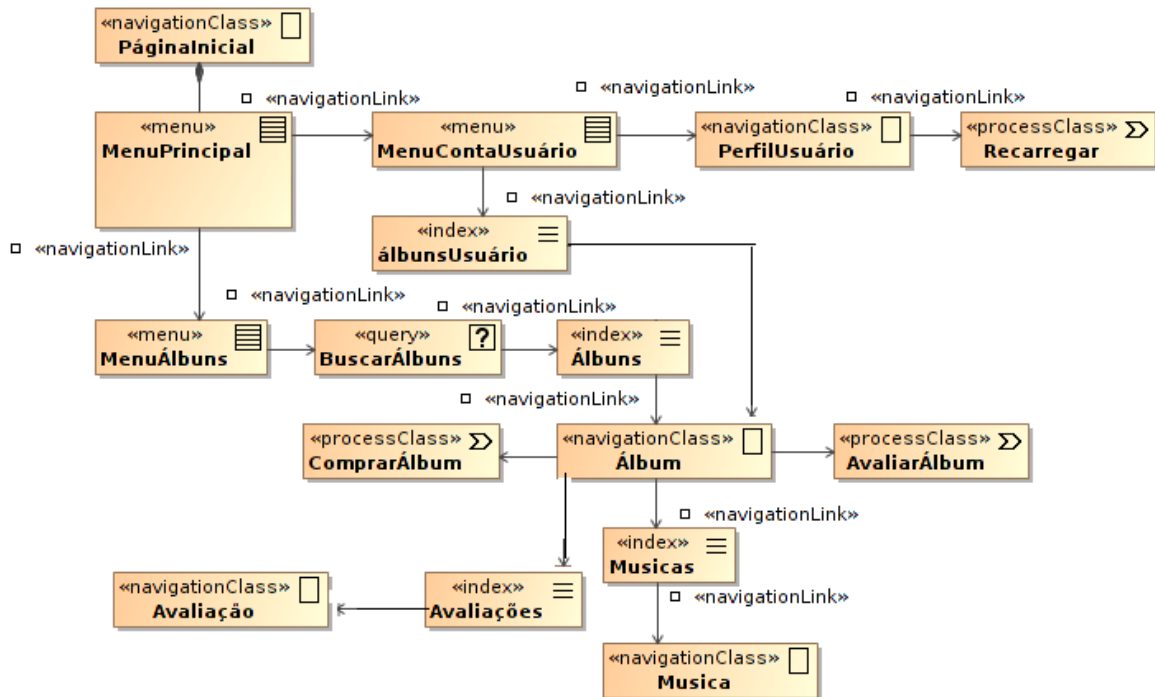


Figura 5.8: Modelo de navegação do sistema para clientes e funcionários.

A Figura 5.9 é o modelo de navegação do sistema que mostra as possibilidades de navegação dos funcionários. Além do *menu* de álbuns e do *menu* da sua conta, o funcionário tem acesso ao *menu* de compras e ao *menu* de usuários. Além da opção de buscar álbum, que aparece no modelo anterior, o *menu* de álbuns do funcionário possui a opção de incluir álbum. No modelo da figura 5.8 foram definidas algumas possibilidades de navegação dos funcionários. Essas possibilidades não aparecem no modelo da figura 5.9

porque foi decidido simplificar e dar prioridade às novas possibilidades de navegação e operação dos funcionários. O modelo não mostra o *menu* da conta porque o funcionário pode acessar a partir desse *menu* os mesmos elementos definidos no modelo da figura 5.8.

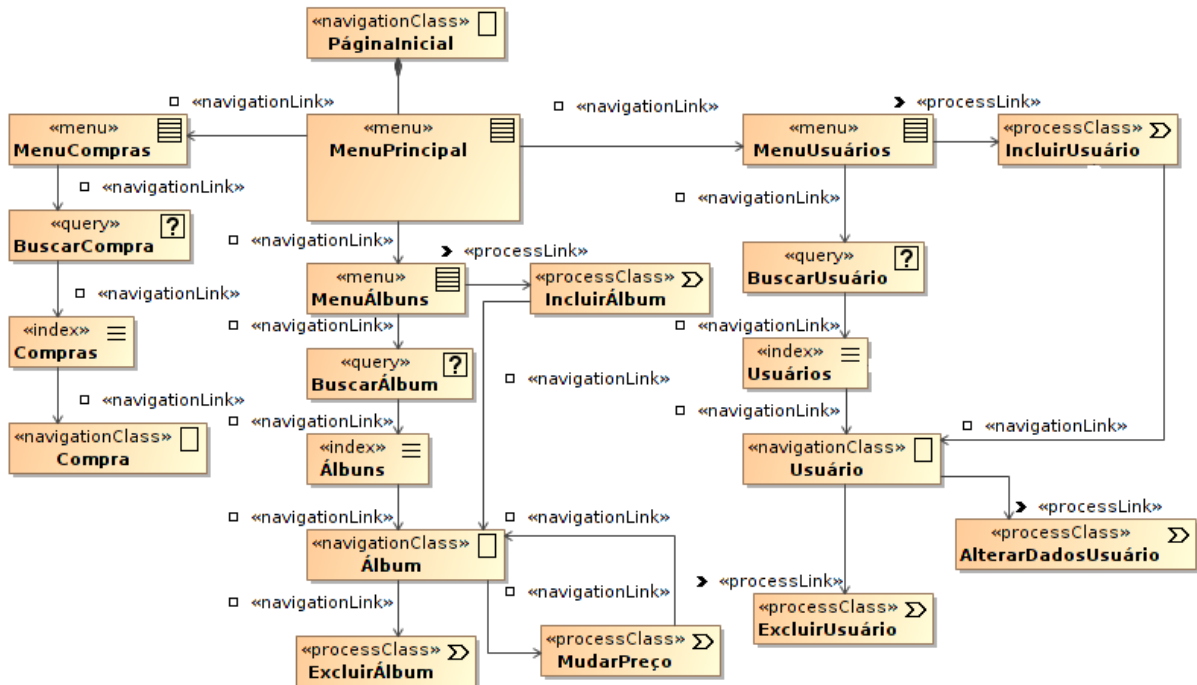


Figura 5.9: Modelo de navegação do sistema para funcionários.

5.5 Modelo de apresentação do sistema

Para a construção do modelo *UWE* de apresentação, deve-se usar os modelos *UWE* de conteúdo e de navegação obtidos e o método definido no capítulo anterior. O modelo de apresentação do sistema consiste de *wireframes*, os quais representam as páginas que formam o sistema. Os textos de *links* e botões do sistema precisam ser bem escolhidos. Como visto no capítulo 3, ao ler o texto, o usuário precisa saber o que acontecerá se o botão ou *link* for selecionado.

A figura 5.10 mostra seis *wireframes*. Na página chamada "MenuContaUsuário", estão posicionados elementos do tipo *anchor*, os quais representam *links*. A página possui um *link* para a página com todos os álbuns comprados pelos usuários (página "ÁlbunsUsuário") e um *link* para a página de recarga de créditos (página "Recarregar"), a qual recebe a quantidade a ser recarregada e informações do pagamento. A figura 5.11 mostra outros *wireframes* do modelo *UWE* de apresentação do sistema. A página "Álbum" tem as informações detalhadas de um álbum, a lista de músicas e a lista de avaliações do

álbum. Além disso, possui *links* para comprar o álbum, avaliar o álbum e fazer *download* do álbum. Os *links* para fazer *download* e avaliar estão disponíveis para usuários que compraram o álbum.

Os *wireframes* apresentados representam as páginas que compõe o sistema de compras de álbuns musicais. No modelo também estão especificadas todas as ligações entre as páginas do sistema. Após obter o modelo de apresentação, pode-se partir para implementação de cada página descrita nesta seção.

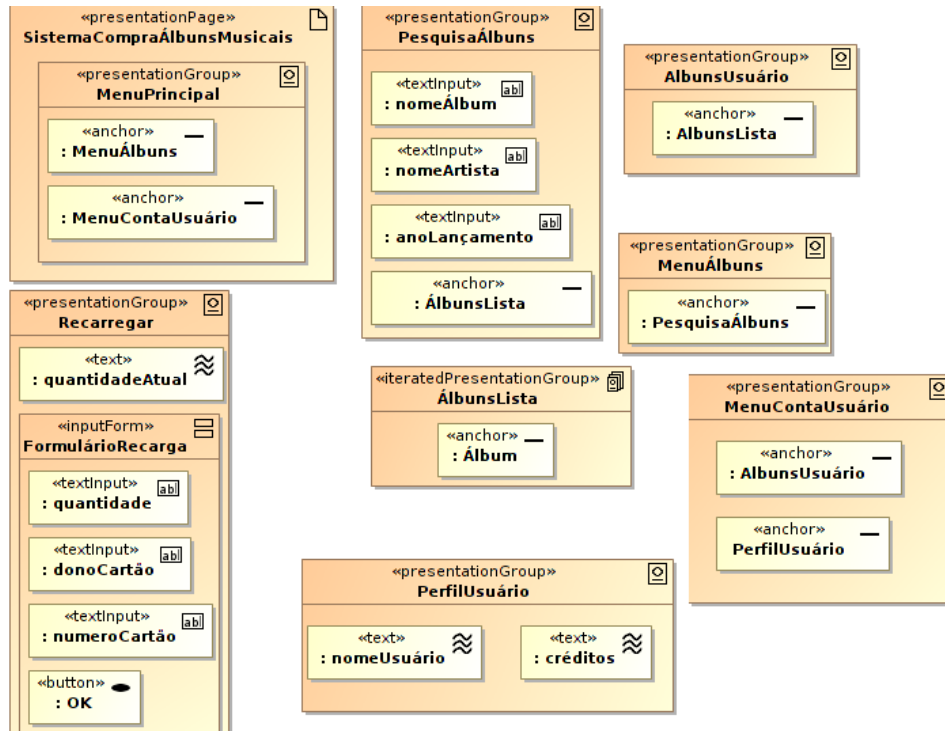


Figura 5.10: Exemplos de *wireframes* do modelo de apresentação do sistema.

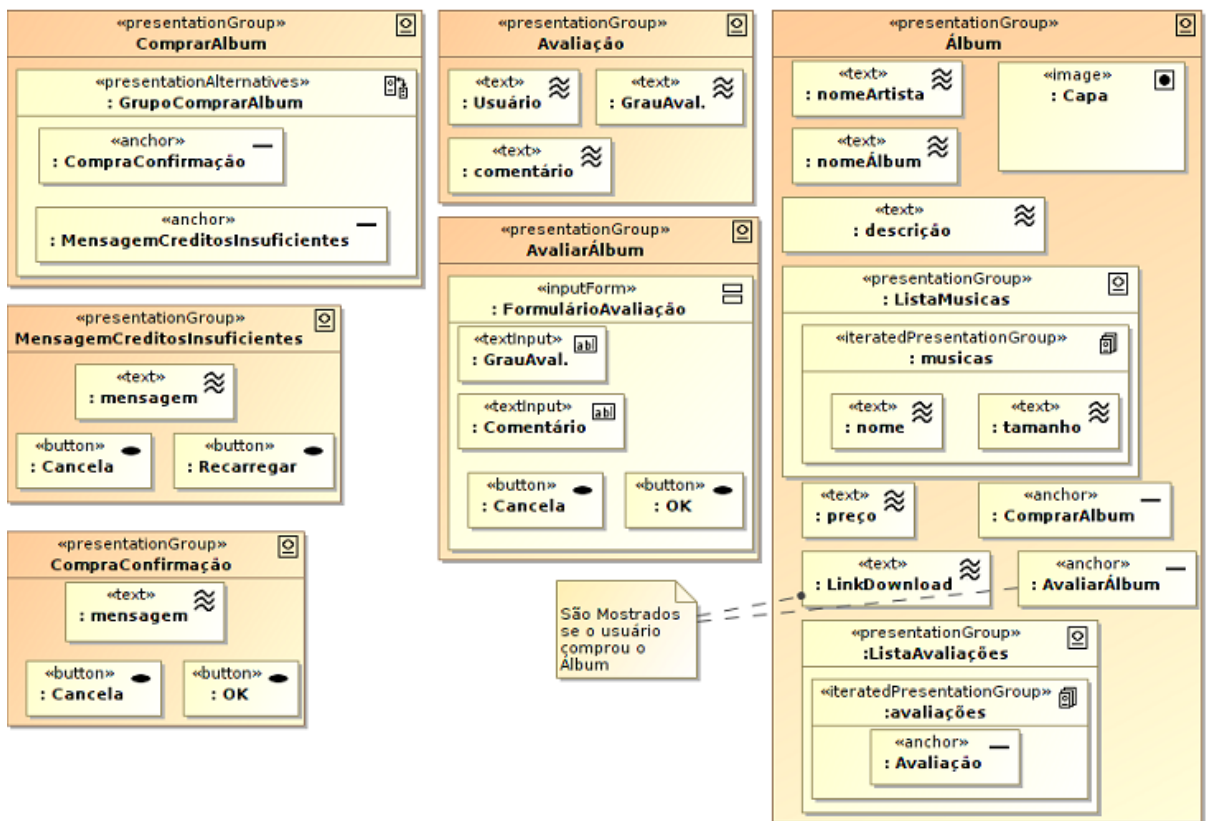


Figura 5.11: Exemplos de *wireframes* do modelo de apresentação do sistema.

Capítulo 6

Abordagem prática: *Design* e implementação

Após a fase de especificação e projeto, iniciou-se a implementação do sistema usando os modelos definidos, especificados e documentados no capítulo anterior. Antes que se iniciasse a implementação, foi necessário escolher tecnologias para criar o sistema de maneira prática e funcional. Além disso, um banco de dados precisou ser implementado.

Este capítulo apresenta uma breve descrição da tecnologia utilizada, detalhes sobre a implantação do sistema, aspectos da implementação, o esquema de banco de dados desenvolvido e uma breve descrição de uso do sistema.

6.1 *Oracle Application Express (APEX)*

Após a pesquisa de uma tecnologia viável para a implementação do sistema, chegou-se a conclusão que o *Oracle Application Express (APEX)* é a ferramenta que melhor encaixa no foco do projeto. Essa ferramenta permitiu a criação fácil e rápida das funcionalidades do sistema. As facilidades oferecidas pelo *APEX* possibilitaram desenvolver as páginas do sistema e a navegação especificada no modelo de apresentação e no modelo de navegação.

APEX é um ambiente de desenvolvimento de *software* baseado na web que é executado em um SGBD *Oracle*. A ferramenta vem por padrão (sem custo adicional) em todas as edições do SGBD *Oracle*. Pode ser usado para construir aplicações *web* que podem ser utilizadas em *browsers*. [12]. O ambiente de desenvolvimento *APEX* também é baseado em *browser*.

Aplicações *APEX* usam construções *PL/SQL* como a linguagem do servidor. Bem como acessam dados através de blocos *PL/SQL*, uma aplicação *APEX* vai usar *PL/SQL* para exibir elementos de uma página *web* [10]. Algumas vantagens do *APEX* são: a facilidade para a criação de protótipos; a facilidade para implantação, o usuário final usa

uma *URL* para acessar uma aplicação; o processamento e as validações serem feitas do lado do servidor; possuir uma grande comunidade de usuários (desenvolvedores) [14].

6.2 Implantação do sistema

Os usuário precisam de um *browser* e acesso à *Internet* para utilizar o sistema. Para a execução do sistema (aplicação *APEX*) é necessário um servidor *Web*, um servidor de aplicação, um *software* chamado *Apex listener* e um servidor de banco de dados *Oracle*.

A figura 6.1 mostra o diagrama de implantação do sistema.

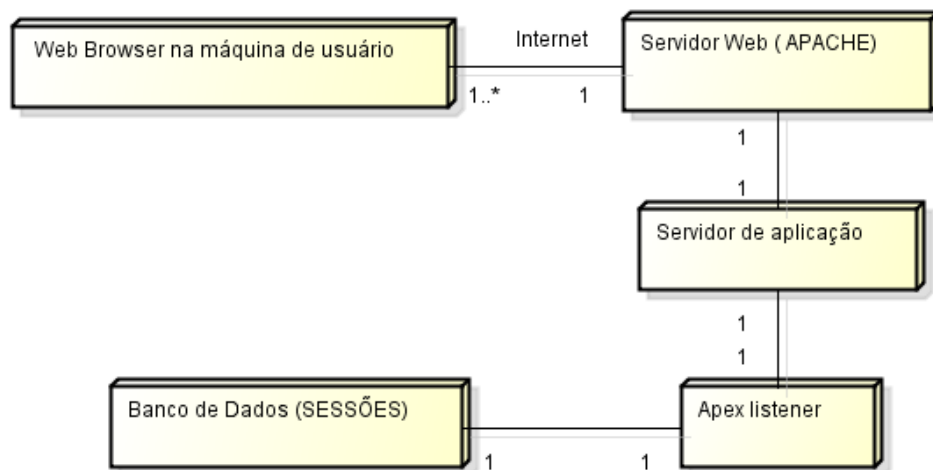


Figura 6.1: Diagrama de implantação.

Uma requisição (*request*) HTTP, que é um protocolo padrão WWW executado sobre o TCP/IP, é passada quando o usuário acessa uma página do sistema pelo navegador; o servidor web recebe o *request* da web e identifica que é uma requisição do *Apex*, redirecionando a requisição ao servidor de aplicação correto; o servidor de aplicação organiza os pedidos (filtros, cache, *timeouts*, filas etc) e transfere para o *Apex listener*; o *Apex listener* recebe pedido, cria a *thread* Java para efetivamente ler dados do banco e/ou executar códigos PL/SQL, que são a lógica de uma aplicação *APEX*. As *threads* interagem com sessões de banco, onde os dados são buscados. Faz-se o caminho inverso com os dados coletados, PL/SQLs executados e códigos HTML, Javascript, CSS, Ajax e jQuery repassados ao navegador que mostra a tela ao usuário.

Caso deseje-se desenvolver aplicações *APEX* e não colocar as aplicações em produção, a *Oracle* disponibiliza gratuitamente o *APEX* e a infraestrutura necessária para desenvolver e executar as aplicações.

6.3 Projeto do banco de dados

Após a permissão para usar a ferramenta ser obtida, iniciou-se o projeto de banco de dados do sistema. O modelo *UWE* de conteúdo da figura 5.7 foi usado para determinar as tabelas e os relacionamentos do esquema de banco de dados do sistema. As tabelas foram criadas no *APEX*. A figura 6.2 mostra o diagrama com o esquema de banco de dados do sistema. As colunas com "PK" antes do nome são chaves primárias e as colunas com "FK" antes do nome são chaves estrangeiras.

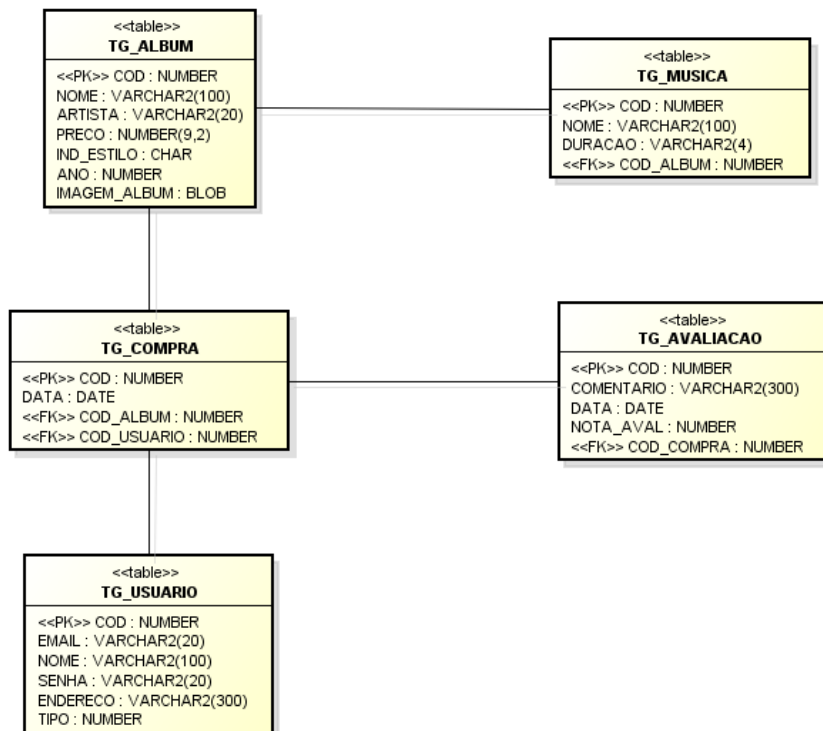


Figura 6.2: Esquema de banco de dados do sistema.

As classes "Álbum", "Música", "Compra", "Avaliação" e "Usuário" foram usadas para criar as tabelas "TG-ALBUM", "TG-MUSICA", "TG-COMPRA", "TG-AVALIACAO" e "TG-USUARIO".

Uma coluna com nome "COD" precisou ser criada em todas as tabelas, essa coluna é a chave primária das tabelas. As colunas da tabela "TG-ALBUM" são a chave primária e os atributos da classe "Álbum" do modelo *UWE* de conteúdo. Se o atributo da classe for "int", a coluna será do tipo "NUMBER". Se o atributo for "String", a coluna será do tipo "VARCHAR2".

O modelo de conteúdo especifica que "Álbum" precisa ter uma ou mais classes do tipo "Música". Por isso, uma coluna que é chave estrangeira foi criada na tabela "TG-ALBUM" para referenciar a tabela "TG-MUSICA". Como a tabela "TG-ALBUM" referencia

"TG-MUSICA", um relacionamento foi criado entre elas. As outras tabelas do esquema foram obtidas de maneira similar a descrita para a tabela "TG-ALBUM".

6.4 Aspectos da implementação

Nesta seção são descritos os elementos do *APEX* usados na implementação do sistema e como esses elementos foram utilizados para implementar o que foi especificado no modelo *UWE* de apresentação.

Após a criação das tabelas do esquema de banco de dados, a primeira atividade foi usar o *APEX* para a criação da aplicação e criação das páginas da aplicação, que foram especificadas no modelo de apresentação.

Numa aplicação *APEX*, alguns elementos de uma página são: regiões; itens; botões; processos. Uma página pode ter várias regiões e processos, cada região pode ter vários itens e botões. Esses elementos podem ser de vários tipos. *Form* (formulário para a entrada de dados) e *report* (relatórios criados usando SQL) são alguns tipos de regiões. Na figura 6.3 está a tela do *APEX* para a edição da página do sistema para recarregar créditos.

Page Rendering		
Page ✎ 📄 +		
Page Name: Recarregar	Template: Two Level Tabs	
Title: Recarregar	Header Text:	
HTML Header:	Footer Text:	
HTML Body:	Build Option:	
Help Text:	Authorization: No	
Page Group:	Cached: No	
Regions ✎ 📄 +		
Display Point Page Template Body (2)		
5 ⚙ Mostra créditos	PL/SQL	
10 🔗 <span style="color:DarkBlue;font-size:18	Static Content (Column 3)	
Buttons ✎ 📄 +		
Region Infor		
10 Cancela	Redirect to page 27	
30 Ok	Redirect to page 27	
Items ✎ 📄 +		
Region Informação Paga		
10 P29_QUANTIDADE	Quantidade	Number Field
20 P29_DONO_CARTAO	Dono do cartão	Text Field
30 P29_NUMERO_CARTAO	Numero do cartão	Text Field

Figura 6.3: Tela do *APEX* para edição da página para recarregar créditos.

A navegação utilizada no sistema é do tipo embarcada (ver capítulo 3). Essa navegação é composta por uma navegação global que possui navegações locais e por navegação contextual.

No modelo *UWE* de apresentação do sistema (figura 5.10), os elementos do tipo *anchor* presentes no "MenuPrincipal" representam os elementos da navegação global do sistema. Os elementos da navegação global do sistema foram implementados no *APEX* usando *Tab sets* (lista de *tabs*). As *Tabs* da navegação global estão presentes em todas as páginas do sistema e direcionam para páginas com informações essenciais ou páginas com operações fundamentais. No modelo de apresentação, as *tabs* e as *Tab sets* do sistema estão especificadas no *wireframe* "MenuContaUsuário" e no *wireframe* "MenuÁlbuns".

A figura 6.4 mostra a tela do *APEX* para edição das *tabs* do sistema. As *Tab sets* da navegação global são: "Usuários"; "Compras"; "Minha Conta"; "Álbuns". As *tabs* da *Tab set* "Álbuns" também aparecem na figura 6.4.

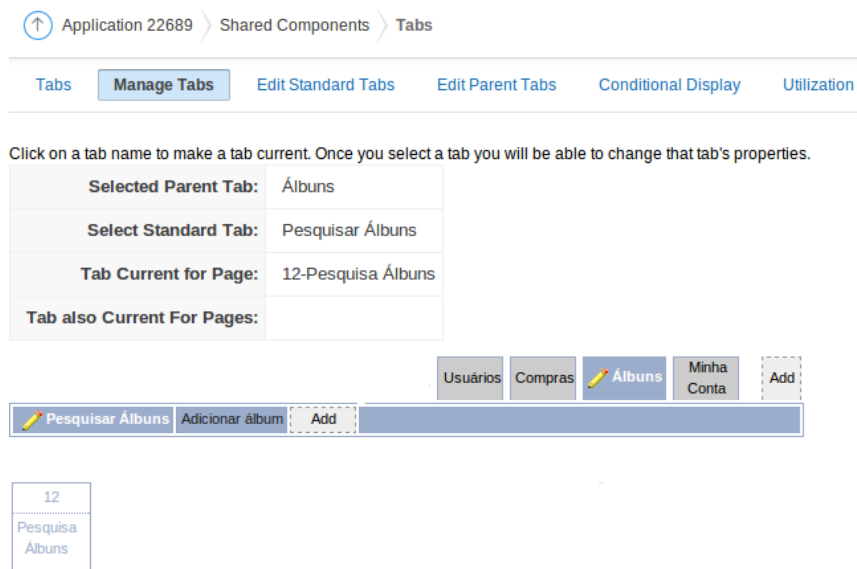


Figura 6.4: Telas do *APEX* para edição de *tab sets* e *tabs*.

No modelo de apresentação existe a classe "Recarregar" do tipo *presentation group* (*wireframe*). Esse *wireframe* possui o elemento "FormularioRecarga" do tipo *inputForm* e o elemento "quantidadeAtual" do tipo *text*. O *inputForm* possui elementos do tipo *textInput* ("quantidade", "donoCartao", "numeroCartao") e o elemento "Ok" do tipo *button*.

Uma página foi criada no *APEX* para implementar o que está definido em "Recarregar". O *inputForm* "FormularioRecarga" foi implementado usando uma região do tipo *static* que possui itens do tipo *Text Field*. Esses itens são utilizados para gerar espaços na tela para entrada de dados. No caso da página, os itens geram espaços para o usuário informar as

informações do cartão de crédito e a quantidade que deseja recarregar. O *button* "Ok" foi implementado usando um botão na região onde estão os itens. Para saber a tela que o usuário é direcionado ao selecionar o botão, é preciso analisar os modelos de navegação e apresentação. Os modelos especificam que o usuário deve ser direcionado para tela especificada pelo *wireframe* "AlbunsUsuário".

Foi criado o processo "Insere Creditos" e definida a condição para o processo executar. A condição é selecionar o botão "Ok". O valor de um dos itens é a quantidade de créditos informada pelo usuário. O processo usa esse item no comando *update* do SQL para atualizar a quantidade de créditos do usuário.

O modelo *UWE* de apresentação especifica que o *wireframe* "Recarregar" também possui o elemento "quantidadeAtual" do tipo *text*. Nesse texto está a quantidade de créditos do usuário, o texto muda se o usuário fizer uma recarga. A região "Mostra créditos" contém o código PL/SQL para mostrar a quantidade de créditos do usuário. A figura 6.5 apresenta o código PL/SQL para mostrar os créditos. O código usa o item "P12-X-NOME-USUARIO" de outra página. O valor desse item é o nome do usuário usando o sistema. O item é usado no comando *select* que acessa o registro com os créditos do usuário. O texto com os créditos do usuário é formatado usando *HTML*. A função "sys.htp.p()" mostra o texto na tela.

Os elementos do tipo *inputForm*, *textInput*, *button* e *text* presentes em *wireframes* do modelo de apresentação foram implementados de maneira similar a descrita para os elementos do *wireframe* "Recarregar". Os textos que podem mudar foram implementados usando código similar ao da região "Mostra créditos". Algumas telas do sistema possuem texto que não muda. Esses textos foram implementados usando regiões do tipo *Static Content*, essas regiões são usadas para mostrar textos estáticos nas telas de um aplicação.

O *wireframe* "PerfilUsuário" (figura 5.10) representa a tela do sistema onde aparecem as informações do usuário. A página "Minha Conta" foi criada no *APEX* para implementar o que está definido nesse *wireframe*. No *wireframe* estão os elementos "nomeUsuário" e "créditos", que são do tipo *text*. A região "Mostra usuário" foi criada na página "Minha Conta" para mostrar os textos na tela. Como os textos não são estáticos, o nome do usuário usando o sistema e a quantidade de créditos podem mudar, o código da região "Mostra usuário" é similar ao código da figura 6.5.

O *wireframe* "MenuContaUsuário", que representa o *menu* do usuário, também apresenta o elemento "AlbunsLista" do tipo *anchor*. Esse elemento representa um *link* para a tela onde aparece a lista de álbuns do usuário. A tela é representada por um *wireframe* do tipo *iteratedPresentationGroup* que contém um *anchor* (*link*). Esse *wireframe* (figura 5.11) representa uma tela com uma lista de álbuns. Cada elemento da lista tem um *link* para tela que contém informações detalhadas do álbum. Foi criada no *APEX*, a região

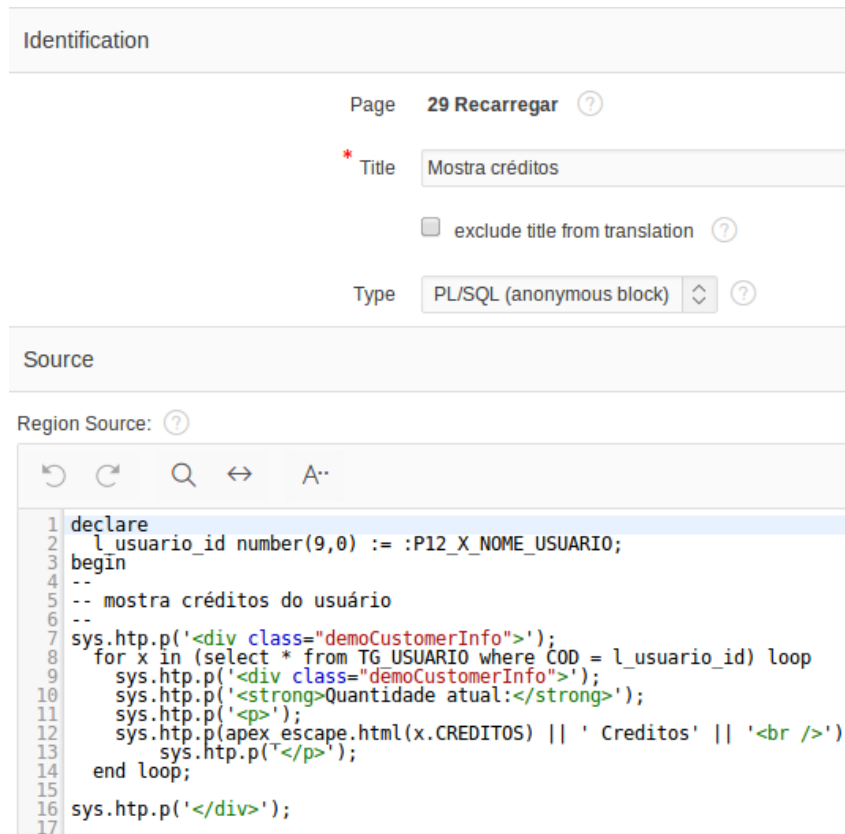


Figura 6.5: Tela do APEX para edição da região "Mostra créditos".

"Meus Álbuns" do tipo "relatório interativo" para implementar a lista de álbuns especificada no *wireframe* "AlbunsLista". A região contém o comando *select* do SQL. O comando é usado para recuperar os álbuns do usuário.

Os elementos do tipo *anchor* contidos em *wireframes* do tipo *iteratedPresentationGroup* foram implementados da mesma maneira que "AlbunsLista", usando regiões do tipo relatório interativo. No modelo de apresentação, existem elementos do tipo *anchor* que não estão presentes em *wireframes* do tipo *iteratedPresentationGroup*. Nesses casos, o *anchor* foi implementado usando o elemento do APEX que é do tipo *button* (botão). No APEX, o desenvolvedor define a tela que aparecerá ao selecionar o botão.

6.5 O sistema desenvolvido

Nesta seção, serão apresentadas algumas telas do sistema desenvolvido para exemplificar a sua utilização.

A figura 6.6 mostra a tela para a pesquisa de álbuns, nessa tela estão os parâmetros de pesquisa que podem ser utilizados. A tela apresenta as *tabs* da navegação global:

"Álbuns"; "Minha Conta"; "Usuários"; "Compras". A tela também apresenta a *Tab set* (navegação local) de álbuns: "Pesquisar álbuns"; "Adicionar álbuns". Ao selecionar o botão "Pesquisar", o usuário vai para tela com os álbuns que são resultado da pesquisa.

Figura 6.6: Tela de pesquisar álbuns.

A figura 6.7 mostra a tela que apresenta os resultados de uma pesquisa por álbuns. Nessa tela estão as informações dos álbuns que correspondem à pesquisa do usuário. Os nome dos álbuns podem ser selecionados e direcionam para a tela com informações detalhadas do álbum. A tela também apresenta a navegação contextual no canto superior. Essa navegação mostra a tela que o usuário está e as telas que ele passou. No caso da figura, o usuário passou pela página "Pesquisar álbuns" e está na tela "Álbuns encontrados".

A figura 6.8 mostra a tela para editar informações de um álbum. O usuário com perfil de "funcionário" pode usar essa parte para mudar informações de um álbum ou excluir o álbum. Os campos de um álbum em vermelho precisam ser informados. Caso o usuário não informe, um erro é reportado ao selecionar o botão "Salvar".

A figura 6.9 é parte da tela para visualizar informações de um álbum. Essa parte mostra as informações do álbum e a foto da capa do álbum. Ao selecionar o botão "Comprar" o usuário pode fazer o *download* do álbum caso tiver créditos. Caso o usuário já tenha comprado o álbum, o *link* para *download* aparece no lugar do botão "Comprar".

A figura 6.10 é outra parte da tela para visualizar informações de um álbum. Essa parte mostra as músicas e avaliações do álbum feitas por usuários. Se o álbum for um dos álbuns comprados pelo usuário, o usuário pode avaliar o álbum. A avaliação pode ser feita apenas uma vez e não pode ser alterada.

A figura 6.11 mostra a tela para recarregar os créditos do usuário. A tela mostra a quantidade de créditos que o usuário possui. Se o usuário não informar todos os campos, uma mensagem de erro aparece. Ao selecionar "Ok", o funcionário é direcionado a tela para confirmar transação financeira.

A figura 6.12 mostra a tela para o usuário visualizar a quantidade de créditos que ele possui e os álbuns que ele comprou.

A figura 6.13 mostra a tela para avaliar álbum. O usuário escolhe uma avaliação entre as disponíveis e pode fazer um comentário. No canto superior da tela aparece a navegação contextual, as páginas que o usuário passa para chegar até a tela para avaliar álbum.

SCAM - Sistema de compra de álbuns musicais

Álbuns Minha Conta Usuários Compras

Pesquisar Álbuns > Álbuns encontrados

Álbuns encontrados

Nome	Preço R\$	Capa	Estilo Musical	Artista
Never Mind the Bollocks, Here's the Sex Pistols	19.9		ROCK	Sex Pistols
25 Bach Favorites	20		Música Clássica	J. S. Bach

1 - 2

SCAM-Sistema de compra de álbuns musicais

Figura 6.7: Tela com álbuns de uma pesquisa.

Editar Álbum

Nome do Álbum

Artista

Descrição do álbum

Estilo

Ano

Preço

Imagem do álbum No file chosen

SCAM - Sistema de compra de álbuns musicais

Figura 6.8: Tela para editar álbum.

SCAM - Sistema de compra de álbuns musicais

[Álbuns](#) | [Minha Conta](#) | [Usuários](#) | [Compras](#)

[Pesquisar Álbuns](#) > [Álbuns encontrados](#) > [Informações do Álbum](#)

Informações do Álbum

25 Bach Favorites
Artista: J. S. Bach
 Bach é um dos maiores compositores de todos os tempos. Esse álbum apresenta as canções de Bach mais adoradas pelo público.
 R\$20

Capa



Figura 6.9: Tela para visualizar informações do álbum.

Músicas

Nome	Duração
1. Toccata In D Minor For Organ	2:56
2. Brandenburg Concerto No. 1 In F: I. Allegro	8:45
3. Brandenburg Concerto No. 3 In G: I. Allegro	7:40
4. Brandenburg Concerto No. 5 In D: II. Affetuoso	9:23

1 - 4

Avaliações de Clientes

Avaliação	Comentário
Bom	As músicas do álbum estão entre as melhores composições de Bach. Porém, o som das músicas não apresenta uma qualidade muito boa.

Figura 6.10: Tela para visualizar informações do álbum.

SCAM - Sistema de compra de álbuns musicais

Álbuns | Minha Conta | Usuários | Compras

Meus Álbuns | [Recarregar Créditos](#) |

Recarregar Créditos

Quantidade atual: 40 Créditos

Informações do Pagamento

Quantidade 40 Créditos ▾

Dono do cartão gif

Numero do cartão 67900

SCAM-Sistema de compra de álbuns musicais

Figura 6.11: Tela para recarregar créditos.

SCAM - Sistema de compra de álbuns musicais

[Álbuns](#) | [Minha Conta](#) | [Usuários](#) | [Compras](#) |

[Meus Álbuns](#) | [Recarregar Créditos](#) |

Olá, Uchego

Você tem 20 Créditos

Meus Álbuns

Nome	Artista	Ano De Lançamento
25 Bach Favorites	J. S. Bach	1996

1 - 1

SCAM-Sistema de compra de álbuns musicais

Figura 6.12: Tela para usuário ver a quantidade de créditos e os álbuns que ele comprou.

SCAM - Sistema de compra de álbuns musicais

[Álbuns](#) | [Minha Conta](#) | [Usuários](#) | [Compras](#) |

[Pesquisar Álbuns](#) > [Álbuns encontrados](#) > [Informações do Álbum](#) > [Avaliar Álbum](#)

Avaliar Álbum

Avaliação

Comentário

O melhor álbum deles. Os outros foram cópias deste.

Avaliação Excelente ▼

SCAM-Sistema de compra de álbuns musicais

Figura 6.13: Tela para avaliar álbum.

Capítulo 7

Análise de resultados

As atividades descritas no método *UWE* orientaram a construção dos modelos usados no desenvolvimento do sistema. Neste capítulo serão apresentados os aspectos da Arquitetura da Informação usados no projeto, será feita uma análise sobre o uso do método *UWE*, serão apresentadas as dificuldades encontradas e será feita a avaliação das tecnologias usadas para a implementação.

7.1 Aspectos da Arquitetura da Informação

Conhecimentos acerca da Arquitetura da Informação foram usados na modelagem do conteúdo, navegação e apresentação do sistema. Nesta seção serão apresentados os aspectos de AI abordados e não abordados no projeto.

O esquema organizacional foi um artefato de AI utilizado no projeto, esse artefato especifica as unidades de conteúdo de informação do sistema e as associações entre essas unidades. O esquema organizacional foi útil no projeto de banco de dados.

A rotulagem do sistema seguiu os princípios da AI, ou seja, os nomes de *links*, *tabs* e botões foram escolhidos para que os usuários não tenham uma noção equivocada sobre o conteúdo representado por esses elementos. Além disso, a rotulagem do sistema não é inconsistente, o sistema não tem rótulos diferentes para *links* que direcionam para uma mesma página.

A navegação do sistema, formada por navegação global, navegações locais e navegação contextual, ajuda o usuário a encontrar informações e funcionalidades. Como a navegação oferece caminhos complementares para encontrar o conteúdo, álbuns e usuários são fáceis de serem encontrados. Conhecimentos de AI foram usados na modelagem da navegação para que o usuário, ao utilizar o sistema, saiba onde esteve, onde pode ir a partir de uma página e saiba o que fazer para chegar em uma página.

A desvantagens em realizar atividades e utilizar artefatos de AI é que essas atividades tomam muito tempo. A maior dificuldade foi identificar problemas e arrumar os modelos antes da implementação. No entanto, se as atividades não fossem realizadas, mudanças teriam que ser feitas após o desenvolvimento, o que não é desejado.

Alguns aspectos de AI não foram abordados no projeto. A busca do sistema não usa algoritmos de indexação, metadados e vocabulários controlados para recuperar os álbuns procurados pelos usuários. No caso do sistema, os campos de metadado são o nome do artista, o nome do álbum e o estilo musical. O Sistema utiliza apenas a busca por palavras-chave (*keywords*) para encontrar os álbuns. Utilizando apenas palavras-chave, alguns resultados relevantes não aparecem como resultado de uma pesquisa. O sistema terá que considerar relacionamentos entre os termos de um vocabulário controlado. Ao considerar esses relacionamentos, a busca terá melhor recuperação da informação.

7.2 Aspectos positivos do método *UWE*

O método *UWE* é flexível, permitiu a criação de diferentes *wireframes* para uma mesma tela do sistema e produziu modelos que especificam bem o conteúdo, a navegação e apresentação do sistema. Ao analisar o modelo de conteúdo, os grupos de conteúdo do sistema e os atributos desses grupos são facilmente identificados. O modelo de conteúdo também foi útil para determinar as tabelas e os relacionamentos do esquema de banco de dados do sistema.

O método apresenta atividades bem definidas para obter as classes de navegação. Não houve dificuldade para obter o modelo de navegação a partir dos modelos de conteúdo e de casos de uso. Outro aspecto positivo do método é permitir o desenvolvimento de vários modelos de navegação para um mesmo conteúdo. O modelo de navegação especifica bem as possibilidades de navegação, as ligações entre as páginas do sistema foram facilmente identificadas.

7.3 Aspectos negativos do método *UWE*

Houve dificuldade para mapear algumas classes do modelo de navegação para *wireframes* do modelo de apresentação. De acordo com o método, uma classe do tipo *process class* ligada a uma classe do tipo *navigation class* pode gerar um elemento do tipo *button* no *wireframe* que representa a *navigation class* ou pode gerar um novo *wireframe* e um elemento do tipo *anchor*, que é um *link* para o novo *wireframe*. Em alguns casos, criar um novo *wireframe* não é a melhor opção. Por exemplo, é melhor criar um *button* para representar a operação "Excluir Álbum" do que um *anchor (link)* para outra página. Alguns

wireframes inadequados foram criados porque o método não mostra qual a melhor opção em cada caso. Os *wireframes* que não estavam adequados foram substituídos por outros.

Outro aspecto negativo foi que os *wireframe* do modelo de apresentação não definem o estilo (formato, cor etc) de botões, textos e *tabs* do sistema. Isso é ruim porque, normalmente, é desejado especificar o formato de textos e *tabs* e ter uma idéia de como esses elementos serão apresentados nas telas do sistema.

7.4 Avaliação da utilização do *APEX* no desenvolvimento do sistema

O uso do *APEX* no desenvolvimento do sistema teve vantagens e desvantagens. Foi fácil identificar os recursos do *APEX* necessários para implementar os elementos presentes nos *wireframes* do modelo de apresentação. O tratamento de erro na execução das operações do sistema foi facilmente realizado utilizando os elementos do tipo *validation* que são disponibilizados pelo *APEX*. O uso de *authorization schemes* (esquemas de autorização) facilitaram restringir o acesso de usuários com perfil de cliente a determinadas telas do sistema. O fato das *tabs* serem criadas uma vez e compartilhadas pelas páginas (telas) do sistema é bom porque não há necessidade de criar as *tabs* em cada página do sistema.

O *APEX* tem alguns aspectos negativos. A ferramenta disponibiliza apenas *tabs* como estrutura de navegação, existem estruturas que oferecem um suporte melhor para encontrar informações. Além disso, o *APEX* não é bom para implementar a navegação contextual do sistema. A navegação contextual deve mostrar a tela que o usuário estava antes. O *APEX* não permite que a navegação contextual de uma tela dependa do caminho utilizado para chegar até essa tela. Por exemplo, utilizando o sistema é possível acessar a tela com informações de um álbum a partir da tela que mostra os álbuns do usuário ou a partir da tela que mostra os álbuns que são resultados de uma pesquisa. A navegação contextual da tela com informações de um álbum é sempre a mesma, independente do caminho utilizado. Apresentar a mesma navegação contextual para diferentes caminhos pode gerar perda de contexto e confusão. Esses problemas foram resolvidos, porém, utilizando *APEX* foi dispendioso resolvê-los. Algumas páginas e elementos (regiões, itens etc.) foram criados mais de uma vez. Isso foi feito para que navegação contextual mostrasse corretamente as páginas acessadas pelo usuário durante a sessão ativa.

Em suma, o *APEX* é uma tecnologia que tem os elementos necessários para implementar rapidamente o que foi especificado nos modelos *UWE*. Não foi preciso criar longos e complexos procedimentos *PL/SQL* para implementar *tabs*, relatórios, botões, links, imagens e outros elementos nas telas do sistema. O *APEX* ofereceu recursos que facilitaram a implementação desses elementos. A ferramenta permite criar navegações que mostram

aos usuários onde estão e onde podem ir para encontrar as informações. No entanto, usando o *APEX* não é possível implementar estruturas de navegação que não sejam *tabs*. Além disso, é trabalhoso implementar navegações contextuais.

Capítulo 8

Conclusão

Este trabalho apresentou o contexto histórico da arquitetura da informação, seus princípios, atividades e artefatos para atender os requisitos estabelecidos pelas aplicações *web*. O trabalho buscou apresentar uma visão dos aspectos teóricos e práticos da arquitetura da informação no desenvolvimento de aplicações *web*, apresentando suas características e elementos essenciais para o desenvolvimento nessa área, com ênfase nos modelos para representar o conteúdo, a navegação e a interface com o usuário. Conceitos acerca da arquitetura da informação foram descritos, o método *UWE* para construção dos componentes de arquitetura da informação foi apresentado. O método *UWE*, conhecimentos de Arquitetura da Informação e a ferramenta *APEX* foram utilizados no desenvolvimento de uma aplicação *Web*. A metodologia utilizada e os resultados obtidos foram avaliados.

8.1 Considerações Finais

Foi observado que os *wireframes* do modelo *UWE* de apresentação não mostram detalhadamente a interface com o usuário. Não especificam o tipo das estruturas de navegação, a posição das *tabs*, tamanho, cor e estilo de botões e textos. Esses elementos foram definidos na implementação da aplicação. O desejável é serem visualizados na etapa de modelagem. A visualização desses elementos na etapa de modelagem poderia evitar custos na manutenção da aplicação. Para que os usuário soubessem encontrar uma informação ou funcionalidade no sistema, conhecimentos de AI foram usados para escolher os elementos certos da navegação global e navegações locais. Ao realizar a abordagem prática, foi observado o que foi descrito em outros trabalhos da área. Foram observados problemas como a perda de contexto ao utilizar esquemas organizacionais ambíguos e foi observada a facilidade para acessar o conteúdo ao usar navegações locais e globais. As navegações podem ser usadas para facilitar o acesso em aplicações onde o usuário precisa passar por muitas páginas até chegar a página desejada. Permitir caminhos alternativos para acessar

o conteúdo tem aspectos positivos e negativos. Os usuários tem facilidade em encontrar o conteúdo, porém, a navegação contextual não mostra corretamente as páginas que o usuário passou (perda de contexto). A ferramenta *APEX* não oferece facilidades para resolver esse problema. Além disso, o *APEX* não disponibiliza, além de *tabs*, estruturas de navegação que ofereçam um suporte melhor para encontrar informações. No entanto, a ferramenta oferece facilidade para implementar a navegação e a rotulagem de aplicações. Não há dificuldade para criar e mudar textos, botões, *links* e *tabs* de navegação.

Materiais bibliográficos publicados relacionados a Arquitetura da Informação geralmente não descrevem detalhadamente como usar tecnologias para implementar o que está descrito em modelos. Este trabalho apresentou uma forma para obter modelos de arquitetura da informação usando o método *UWE*, além disso, apresentou as vantagens e desvantagens em utilizar a ferramenta *APEX* na implementação dos componentes especificados por esses modelos.

8.2 Trabalhos Futuros

As sugestões de trabalhos são pesquisar sobre a utilização de metadados e vocabulários controlados na busca de informações. Usar vocabulários controlados como anéis de sinônimos e taxonômias hierárquicas na busca do sistema e avaliar o uso dessas estruturas. Verificar se os vocabulários controlados contribuem para trazer resultados de busca relevantes.

Além de implementar vocabulários controlados, outra atividade é usar o sistema em experimentos semelhantes aos de outros trabalhos. Nesses experimentos, grupos de pessoas usam o sistema para encontrar informações e realizar operações. O número de tentativas até obter sucesso (encontrar informação) é registrado para cada pessoa. O objetivo dessas atividades é verificar a qualidade da navegação, rotulagem e busca do sistema e a facilidade dos usuários para encontrar informações.

Referências

- [1] Nations D. Web applications. http://webtrends.about.com/od/webapplications/a/web_application.htm, 2014. Acessado em : 2015-04-11. 4
- [2] A. Dillon. Information architecture in jasist : Just where did we come from? *Journal of the American Society for Information Science and Technology*, 53(10):821–823, 2002. 5
- [3] D. Dillon, A.;Turnbull. *Encyclopedia of Library and Information Science*, chapter Information Architecture. Marcel-Dekker, School of Information, University of Texas, Austin, Texas, U.S.A, 2006. x, 1, 5, 6, 7
- [4] Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang. A human-centered semantic service platform for the digital ecosystems environment. *World Wide Web*, 13(1-2):75–103, 2010. 15
- [5] E Evernden, R.;Evernden. *Information First: Integrating Knowledge and Information Architecture for Business Advantage*. Butterworth-Heinemann, 2003. 4
- [6] Hedden H. Taxonomies and controlled vocabularies best practices for metadata. *Journal of Digital Asset Management*, 6(5):279–284, 2010. 14, 15
- [7] T.; Prabhu P. V. Helander, M.; Landauer. *Hand- book of Human Computer Interaction*. North- Holland: Amsterdam, 1997. 6, 7
- [8] Cardello J. The difference between information architecture (ia) and navigation. <http://www.nngroup.com/articles/ia-vs-navigation/>, 2014. Acessado em : 2015-04-11. x, 1, 11, 12
- [9] Nielsen J. Mega menus work well for site navigation. <http://www.nngroup.com/articles/mega-menus-work-well/>, 2009. Acessado em : 2015-04-11. x, 13
- [10] M. Lancaster. *Oracle Application Express 4.0 with Ext JS*. Packt Publishing, 2011. 34
- [11] Brown M. *Communicating Design: Developing Web Site Documentation for Design and Planning*. New Riders Press, 2011. 6
- [12] R. Mattamal and A. Nielsen. *Expert Oracle Application Express Plugins: Building Reusable Components*. Apress, 2011. 34

- [13] L. Morville, P.; Rosenfeld. *Information architecture for the world wide web*. O'Reilly Media Inc, 2006. x, xii, 1, 5, 6, 8, 9, 10, 11, 12, 13, 16
- [14] Cimolini P. *Agile Oracle Application Express*. Apress, 2011. 35
- [15] Guenther R. *Understanding Metadata*. NISO press, 2004. 14
- [16] G. A. Reis. Centrando a arquitetura de informação no usuário. http://www.guilhermo.com/mestrado/Guilhermo_Reis-Centrando_a_Arquitetura_de_Informacao_no_usuario.pdf, 2007. Acessado em : 2014-06-14. 6, 8, 10
- [17] L. Resmini, A.; Rosati. A brief history of information architecture. *Journal of Information Architecture*, 3(2):33–46, 2012. 4, 6
- [18] J. M. Rojas, L. A. Macías. Bridging the gap between information architecture analysis and software engineering in interactive web application development. *Science of Computer Programming*, 78(11):2282–2291, 2013. 5
- [19] N. Schwinger, W.; Koch. *Web engineering - the discipline of systematic development of web applications*. Kappel Gerti, 2006. x, 1, 16, 17, 18, 19, 20, 21
- [20] Russel T. *Designing the Search Experience: The Information Architecture of Discovery*. Morgan Kauffmann, 2013. 13
- [21] Zimmermann T. Information architecture. <http://wwwmayr.informatik.tumuenchen.de/konferenzen/Jass05/courses/6/Papers/03.pdf>, 2005. Acessado em : 2014-10-09. x, 9, 10, 14
- [22] L. M. Weitzman. *The architecture of information : interpretation and presentation of information in dynamic environments*. PhD in media arts and science, Massachusetts Institute of Technology. Dept. of Architecture, 1995. 5
- [23] R. S. Wurman. *Information Architects*. Graphis Inc, 1997. 6