



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

e-Design Patterns

Marcelly Luise

Pedro Henrique Vidal Pinho

Thiago Figueiredo

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador

Prof. Dr. Letício Lopes Leite

Brasília
2017



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

e-Design Patterns

Marcelly Luise

Pedro Henrique Vidal Pinho

Thiago Figueiredo

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof. Dr. Letício Lopes Leite (Orientador)
CIC/UnB

Prof.a Dr.a Ada Lovelace
Coordenadora do Curso de Computação — Licenciatura

Brasília, 20 de julho de 2017

Dedicatória

Na *dedicatória* o autor presta homenagem a alguma pessoa (ou grupo de pessoas) que têm significado especial na vida pessoal ou profissional. Por exemplo (e citando o poeta):
Eu dedico essa música a primeira garota que tá sentada ali na fila. Brigado!

Agradecimentos

Nos *agradecimentos*, o autor se dirige a pessoas ou instituições que contribuíram para elaboração do trabalho apresentado. Por exemplo: *Agradeço aos gigantes cujos ombros me permitiram enxergar mais longe. E a Google e Wikipédia.*

Resumo

O *resumo* é um texto inaugural para quem quer conhecer o trabalho, deve conter uma breve descrição de todo o trabalho (apenas um parágrafo). Portanto, só deve ser escrito após o texto estar pronto. Não é uma coletânea de frases recortadas do trabalho, mas uma apresentação concisa dos pontos relevantes, de modo que o leitor tenha uma ideia completa do que lhe espera. Uma sugestão é que seja composto por quatro pontos: 1) o que está sendo proposto, 2) qual o mérito da proposta, 3) como a proposta foi avaliada/validada, 4) quais as possibilidades para trabalhos futuros. É seguido de (geralmente) três palavras-chave que devem indicar claramente a que se refere o seu trabalho. Por exemplo: *Este trabalho apresenta informações úteis a produção de trabalhos científicos para descrever e exemplificar como utilizar a classe L^AT_EX do Departamento de Ciência da Computação da Universidade de Brasília para gerar documentos. A classe UnB-CIC define um padrão de formato para textos do CIC, facilitando a geração de textos e permitindo que os autores foquem apenas no conteúdo. O formato foi aprovado pelos professores do Departamento e utilizado para gerar este documento. Melhorias futuras incluem manutenção contínua da classe e aprimoramento do texto explicativo.*

Palavras-chave: LaTeX, metodologia científica, trabalho de conclusão de curso

Abstract

O *abstract* é o resumo feito na língua Inglesa. Embora o conteúdo apresentado deva ser o mesmo, este texto não deve ser a tradução literal de cada palavra ou frase do resumo, muito menos feito em um tradutor automático. É uma língua diferente e o texto deveria ser escrito de acordo com suas nuances (aproveite para ler [http://dx.doi.org/10.6061/2Fclinics%2F2014\(03\)01](http://dx.doi.org/10.6061/2Fclinics%2F2014(03)01)). Por exemplo: *This work presents useful information on how to create a scientific text to describe and provide examples of how to use the Computer Science Department's L^AT_EX class. The UnB-CIC class defines a standard format for texts, simplifying the process of generating CIC documents and enabling authors to focus only on content. The standard was approved by the Department's professors and used to create this document. Future work includes continued support for the class and improvements on the explanatory text.*

Keywords: LaTeX, scientific method, thesis

Sumário

1	Introdução	1
1.1	Objetivos	2
1.1.1	Objetivo Geral	2
1.1.2	Objetivos Específicos	2
1.2	Organização do Texto	3
2	Design Patterns	4
3	Plataformas Q&A	5
4	e-Design Patterns	6
5	Avaliação	7
6	Conclusão	8
6.1	Conclusão	8
6.1.1	Trabalhos Futuros	8
	Referências	9

Capítulo 1

Introdução

Programar é uma tarefa complexa considerando a quantidade de dúvidas e problemas que surgem no processo de desenvolvimento. Quando uma erro precisa ser corrigido, vários desafios podem surgir: uma nova funcionalidade precisa ser adicionada, um código precisa ser refeito, um design precisa ser melhorado, um sistema legado precisa ser migrado entre outros e, por isso os programadores estão em uma busca constante de respostas e códigos de exemplos que possam auxiliar nessas tarefas.

Nessa perspectiva, surgiram websites que servem como plataformas em que usuários podem realizar perguntas e também responde-las, com o intuito de auxiliá-los a encontrarem soluções para seus problemas. Inicialmente surgiram os tradicionais fóruns sem muitas funcionalidades, basicamente os usuários podiam perguntar e responder as perguntas. Atualmente, essas plataformas se desenvolveram e encontramos verdadeiras comunidade virtuais com diversas funcionalidades: sistemas de TAGS, ranking de reputação, sistemas de votação para perguntas e respostas e filtros para consultas detalhadas como é o caso do StackOverflow [3] e o G.U.J. [4]. Essas aplicações além de facilitar a busca e disseminação de informações acerca de desenvolvimento de software estimulam os usuários a se manterem ativos na plataforma através desses rankings de reputação e sistema de votação como uma forma de ludificação dos fóruns tradicionais. Os usuários que possuem uma maior pontuação recebem privilégios na plataforma, como permissão de editar e excluir respostas.

É comum usuários de todo o mundo buscar na internet soluções para esses problemas que surgem no ciclo de vida do software. As dúvidas, muitas vezes, se repetem e muito dos problemas apresentados revelam características fundamentalmente semelhantes. Diante disso, na engenharia de software surgiram os padrões de projeto (Design Patterns). Segundo Christopher Alexander “Cada padrão descreve um problema que ocorre uma e outra vez em nosso ambiente e, em seguida, descreve o núcleo da solução para esse problema, de tal forma que você pode usar essa solução um milhão de vezes Sem jamais

fazê-lo da mesma maneira duas vezes”[1]. Embora essa definição do autor fazer referência a construção de prédios e cidades ela se aplica adequadamente ao conceito de Design Patterns em linguagens de programação orientada a objeto [2]. Em suma, o Design Patterns na programação descreve uma solução geral para um problema que ocorre com frequência em determinado contexto.

Sua utilização pode melhorar o desempenho do software, tornar seu código mais limpo e elegante, facilitar a manutenção e refatoração. Apesar da utilização de Design Patterns ser uma boa prática entre os programadores e uma recomendação, sua aplicação não é trivial, exigindo além do conhecimento técnico uma experiência para identificar se é pertinente a sua utilização em determinado problema e qual padrão de projeto utilizar. Daí surge a motivação para a criação do projeto deste trabalho, o e-Design Patterns, um software educacional, como aplicativo para Smartphone nos moldes das comunidades virtuais como o stackoverflow e o G.U.J., com seu conteúdo focado em Design Patterns.

É notório o impacto que o desenvolvimento da tecnologia vem imprimindo em nossas vidas de diferentes formas. A crescente disponibilidade dessas tecnologias consequência da redução no custo dos hardwares aliado ao surgimento de softwares cada vez mais aplicados a problemas reais, contribui diretamente para o aumento da demanda na utilização da informática na educação [3]. Não existe uma definição exata de software educacional, pois ainda é uma questão em aberto, mas em uma definição ampla, pode-se entender como aplicativos que reforcem conteúdos educacionais de forma interativa, auxiliem o processo de ensino-aprendizagem e contribuam para o enriquecimento intelectual[4].

1.1 Objetivos

Nesta seção serão descritos os objetivos gerais e os objetivos específicos do trabalho.

1.1.1 Objetivo Geral

Desenvolver um aplicativo para Smartphone no modelo de comunidades virtuais de perguntas e respostas com seu conteúdo focado em Design Patterns tendo como intuito auxiliar e acelerar o processo de aprendizagem dos usuários.

1.1.2 Objetivos Específicos

- Identificar e estudar as funcionalidades, estratégias e elementos das plataformas de Q&A
- Estudar a API Stack Exchange para integração com o protótipo

- Prototipação das Telas
- Avaliar o aplicativo desenvolvido com base na ficha de avaliação de software educacional.

1.2 Organização do Texto

O restante do trabalho está organizado da seguinte forma:

Capítulo 2

Design Patterns

Desenvolvimento Design Patterns

Capítulo 3

Plataformas Q&A

Desenvolvimento Plataformas Q&E

Capítulo 4

e-Design Patterns

4.1 Processo de Desenvolvimento

4.1.1 Ferramentas utilizadas

Descrição do Projeto, funcionalidades e Ferramentas utilizadas

Capítulo 5

Avaliação

Descrição da avaliaçãoodfsfgsdfhgsdhfsdhfsdfhsdf

Capítulo 6

Conclusão

Descrição da conclusão de trabalhos futuros

6.1 Conclusão

6.1.1 Trabalhos Futuros

Referências

- [1] Gamma, Erich: *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995. 2
- [2] Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Joaquim Romaguera i Ramió, Max Jacobson e Ingrid Fiksdahl-King: *A pattern language*. Gustavo Gili, 1977. 2
- [3] Brandão, Edemilson Jorge Ramos: *Repensando modelos de avaliação de software educacional*. 3º Simpósio de Investigação e desenvolvimento de software educativo, 1998. 2
- [4] Oliveira, Noé de *et al.*: *Uma proposta para a avaliação de software educacional*. 2001. 2