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.a Dr.a Letícia Lopes Leite

Coorientador
Prof. Dr. José Ralha

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.a Dr.a Letícia Lopes Leite (Orientador)
CIC/UnB

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

Brasília, 20 de julho de 2017

Dedicatória

Dedicamos esse trabalho a todas as instituições e plataformas que buscam auxiliar e ensinar os programadores em com suas dúvidas e dificuldades no dia a dia de desenvolvimento.

Agradecimentos

Agradecemos aos nossos familiares que sempre nos apoiaram e estiveram ao nosso lado durante nossas vidas e principalmente durante essa fase da graduação.

Resumo

Q&A platforms, ou plataformas de perguntas e respostas são aplicações muito populares atualmente que permitem seus usuários compartilharem o próprio conhecimento através de perguntas e respostas. Essas plataformas servem como uma fonte online para que usuário encontrem de maneira rápida e eficiente respostas objetivas para suas dúvidas. Muitas aplicações desse tipo tem surgido com o seu conteúdo focado para a programação. Dúvidas respondidas e códigos de exemplo ajudam programadores a solucionarem problemas e obstáculos no dia a dia de desenvolvedor.

Design Patterns ou Padrões de Projetos são soluções gerais aplicadas na programação orientada a objeto que visam solucionar problemas recorrentes em determinado contexto no projeto de softwares. Um Design Pattern não é um código pronto para se aproveitar em uma aplicação e sim um modelo para resolver um problema sendo soluções que já foram utilizadas e testadas durante o tempo que nos dá confiança em sua eficácia. A utilização desses padrões trazem mutios benefícios para o projeto de software, como facilitar a manutenção, melhorar a documentação, tornar o software reutilizável entre outros. O grande problema para utilização de Design Patterns é que na maioria das vezes é necessário ser um desenvolvedor experiente para aplicação correta de determinado padrão.

Visando a popularidade e a eficiência das Plataformas Sociais de Perguntas e Respostas em auxiliar programadores com suas dúvidas e dificuldades na utilização de Desgin Patterns por programadores, propomos a criação de um aplicativo para smartphone, no modelo das Q&A Platforms, o e-DesignPatterns. Um aplicativo com objetivo de auxiliar e acelerar o processo de aprendizagem para utilização de Design Patterns utilizando as estratégias e características dessas plataformas para criar um aplicativo de sucesso.

Palavras-chave: Plataformas de perguntas e respostas, Designs Patterns, e-Design Patterns

Abstract

Q&A platforms, or question-and-answer platforms, are very popular applications today that allow your users to share Knowledge itself through questions and answers. These platforms serve as an online source for users to find Quick and efficient objective answers to your questions. Many such applications have come up with their content focused on programming.Answered questions and sample codes help programmers to solve problems and obstacles in the day to day developer.

Design Patterns are general solutions applied to object-oriented programming that address recurring problems In a given context in software design. A Design Pattern is not a code ready to take advantage of an application but a template To solve a problem being solutions that have already been used and tested during the time that gives us confidence in its effectiveness. The use of these Patterns bring many benefits to software design, such as facilitating maintenance, improving documentation, making software reusable among others. The big problem with using Design Patterns is that most of the time it is necessary to be an experienced developer for Application of a given standard.

Aiming at the popularity and efficiency of Social Question and Answer Platforms in assisting programmers with their doubts and difficulties in using Desgin Patterns by programmers, we propose the creation of a application for smartphone, in the model of Q& Platforms, The e-DesignPatterns. An application to help and accelerate the learning process to use Design Patterns using strategies and features of these platforms to create a successful application.

Keywords: Q&A Plataforms, Design Patterns, e-Design Patterns

Sumário

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 [**?**] e o GUJ [**?**]. 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 de mecanismos e estratégias como uma forma de ludificação dos fóruns tradicionais. Os usuários que possuem uma maior reputação recebem privilégios na plataforma, como permissões de editar e excluir posts.

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"[**?**]. 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 [**?**]. 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 e facilitar a manutenção e refatoração do sistema. Embora a utilização de Design Patterns ser uma boa prática e uma recomendação entre desenvolvedores, 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 o padrão adequado a se utilizar. Daí surge a motivação para a o projeto deste trabalho, o e-Design Patterns, um software educacional, como aplicativo para Smartphone nos moldes das plataformas de perguntas e respostas como o Stackoverflow e o GUJ, com seu conteúdo focado em Design Patterns.

É notório o impacto que o desenvolvimento da tecnologia vem imprimindo em nossas vidas de diferente 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 [**?**]. 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, auxilie o processo de ensino-aprendizagem e contribua para o enriquecimento intelectual[**?**].

1.1 Objetivos

Nesta seção serão descritos os objetivos gerais e os objetivos especícos 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 Perguntas e Respostas
- Prototipação das Telas

- Avaliar o protótipo desenvolvido com base na ficha de avaliação de software educacional.

1.2 Organização do Texto

O restante do trabalho está organizado da seguinte forma:

1. O capítulo 2 apresenta uma visão geral sobre Design Patterns, além de descrever alguns dos mais utilizados.
2. O capítulo 3 trata sobre as plataformas de Perguntas e Respostas, apresentando os conceitos básicos, características e estratégias comumente adotadas nesses ambientes virtuais que garantem seu funcionamento e sucesso.
3. O capítulo 4 descreve a nossa proposta para criação do aplicativo e-DesignPatterns e todo o processo de prototipação, além das ferramentas utilizadas.
4. O capítulo 5 apresenta a avaliação do aplicativo com base na ficha de avaliação de software educacional.
5. O capítulo 6 conclui o trabalho. Além da conclusão também é apresentado os trabalhos futuros para melhorias e continuidade do projeto

Capítulo 2

Design Patterns

Com o crescimento exponencial do desenvolvimento de softwares ao redor do mundo, uma crescente necessidade dentro da Engenharia de Software é a de Padrões de Projeto (Design Patterns). Podemos a definir brevemente como uma estrutura reusável de uso geral para se resolver problemas recorrentes que se enfrenta ao desenvolver um software [**?**]. Os padrões de projeto ajudam os desenvolvedores a estruturarem melhor suas idéias, e os dão um poder organizacional coerente e necessário, principalmente em projetos de maior proporção. Também são consideradas boas práticas de programação ao se desenvolver uma aplicação, por fornecerem um template da qual podemos usufruir para tornar um sistema escalável e manutenível.

Os padrões de projeto são de tamanha importância que essa abstração se encontra no intermédio do paradigma da linguagem de programação em questão, e do algoritmo de fato [**?**]. Ela é um template em cima da sua solução de problema, ou seja, do problema que queremos resolver algoritmicamente, e do modo que a faremos utilizando a linguagem de programação escolhida. Portanto, essa é uma área bastante ampla dentro da Engenharia de Software. Diferentes paradigmas de programação possuem diferentes padrões de projeto, e o modo como as pensamos e enxergamos pode mudar drásticamente de uma para outra. Tomemos como exemplo o paradigma de Orientação a Objeto, e vemos que os padrões já estabelecidos como boas práticas raciocinam pensando na relação e interação entre as classes / objetos e na mutabilidade que eles sofrem na aplicação, enquanto no paradigma funcional sabemos que qualquer padrão que envolva mutabilidade não caberia para seus projetos. O paradigma funcional adota conceitos avançados em relação a padrões de projeto, envolvendo composição de funções, e a técnica de Currying, que o permite transformar uma função de múltiplos argumentos para que ela possa ser chamada como uma sequência de execução de funções separadas com um único argumento. De qualquer forma, para todo projeto dentro da engenharia de software e para todo paradigma que se adote, existe essa abstração muito importante do padrão de projeto para

se pensar sobre.

Como foi uma área que cresceu rapidamente, ocorreu o surgimento de três categorias: os padrões criacionais, estruturais e comportamentais.

2.0.1 Criacional

Os padrões de projetos criacionais visam obter controle sobre os mecanismos de criação dos objetos. A forma básica de instanciação leva a problemas de design ou a uma complexidade desnecessária na arquitetura do projeto, e os padrões criacionais tentam resolver esse problema. Eles são rodeados entre duas grandes idéias: a de se encapsular o conhecimento sobre quais classes concretas o sistema usa, e a outra é desativando e ativando as instâncias com base em como elas são criadas, utilizadas e acopladas. Alguns dos padrões criacionais mais famosos são:

- Abstract Factory: disponibiliza a um cliente um conjunto de objetos relacionados ou dependentes. A família de objetos criados é determinada em tempo de execução.
- Builder: é usado para criar objetos complexos constituídos de partes que serão criadas em ordem ditada por um algoritmo.
- Factory Method: é usado para substituir construtores de classe, abstraindo o processo de geração de um tipo instanciado de tal maneira que pode ser determinado em tempo de execução.
- Prototype: é usado para instanciar um novo objeto copiando todas as propriedades de um objeto existente, criando um clone independente. É uma prática muito útil quando a construção de um novo objeto é ineficiente.
- Singleton: garante que apenas um objeto de uma classe é instanciado. Todas as futuras referencias desses objetos se referem apenas a instancia criada.

2.0.2 Estrutural

O padrão estrutural foca em diferentes formas de enxergar relacionamentos entre as entidades do seu sistema. As diferentes abordagens na construção desses relacionamentos possuem vantagens e desvantagens, e portanto, podemos selecionar a melhor solução para diferentes contextos de aplicação.

- Adapter: Faz um link entre dois tipos incompatíveis envelopando uma adaptação com a classe que suporta uma interface necessária para um cliente.

- Bridge: Separa elementos abstratos de uma classe dos detalhes da implementação, disponibilizando meio de trocar detalhes da implementação sem modificar a abstração.
- Composite: é usado para criar hierarquia, árvore de objetos relacionados com estruturas recursivas onde qualquer elemento da estrutura pode ser acessado e utilizado de alguma maneira.
- Decorator: é usado para estender ou alterar funcionalidade de objetos em tempo de execução envelopando-os em um objeto de uma classe decorator.
- Facade: é usado para definir uma interface simplificada para um subsistema mais complexo.
- Flyweight: é usado para reduzir memória e utilização de recurso para modelos complexos contendo milhares de objetos similares.

2.0.3 Comportamental

O padrão comportamental busca identificar padrões comuns na comunicação entre objetos e entidades. Esses necessitam de comunicação entre si dentro de um sistema para compartilhar informações, atualizar estados da aplicação e cuidarem de suas relações. Com isso, é possível padronizar como podemos organizar um projeto para tornar essa comunicação visível e clara.

- Chain of Responsibility: é usado para processar variadas requisições, cada uma pode ser tratada por um diferente handler.
- Command: é usado para expressar uma requisição, incluindo a chamada a ser feita e todos os seus parâmetros necessários e um objeto ordenado. Pode ser sincronizado ou assíncrono.
- Interpreter: é usado para definir a gramáticas para instruções que são parte de uma linguagem ou notação, enquanto facilita que a linguagem seja estendida.
- Iterator: é usado para disponibilizar uma interface padrão para atravessar uma coleção de itens em um objeto agregador sem a necessidade de entender a estrutura por trás.
- Mediator: é usado para reduzir o acoplamento entre classes que se comunicam umas com as outras. Ao invés de classes se comunicarem diretamente, sendo necessário que uma saiba sobre a implementação da outra, as classes enviam mensagens umas para as outras através do objeto mediador.

- Memento: é usado para capturar o estado atual de um objeto e amazena-lo de tal maneira que pode ser restaurado posteriormente sem quebrar nenhuma regra de encapsualmento.
- Observer: é usado para permitir que objetos publiquem alterações de estados. Outros objetos sub-escrevem para serem imediatamente notificados de qualquer mudanças.
- State: é usado para alterar o comportamento de um objeto e suas mudanças de estados internas. State permite que a classe para um objeto altere em tempo de execução.
- Strategy: é usado para criar uma familia de algoritmo compartilháveis onde os processos necessários são escolhidos em tempo de execução
- Template Method: é usado para definir passos básicos para um algoritmo permitir a implementação de passos individuais a serem alterados.

Capítulo 3

Q&A Platforms

3.1 Breve Histórico

O termo web 2.0 emergiu em 2004 e foi popularizado pela empresa americana O'Reilly Media para designar uma nova geração de aplicações e serviços para internet que surgiam para formar a "web participativa", exemplos desses serviços e aplicações são blogs, wikis, redes sociais etc citar [**?**]. Nesse sentido, o desenvolvimento da web criou uma revolução na internet onde seus usuários passaram de consumidores passivos para produtores e compartilhadores de conteúdo. Cada vez mais usuários utilizam fontes online para encontrar as informações que desejam. Motores de pesquisas, como google, yahoo, ask frequentemente são as portas para o encontro dessas informações. Apesar desses mecanismos dominarem as buscas por informações na internet outras fontes de pesquisa e compartilhamento de informações surgiram e ganharam força, são os site de perguntas e respostas ou as plataformas sociais de perguntas e respostas [**?**].

Q&A (Questions and Answers) platforms, ou plataformas de perguntas e respostas são ambientes virtuais em que usuários voluntariamente postam perguntas e outros usuarios respondem. A plataforma armazena em seus bancos de dados esse rico conteúdo composto pelas perguntas e respostas e disponibiliza para que os usuários possam pesquisá-lo. Esse processo torna esses ambientes virtuais em sistemas auto-sustentáveis, ou seja, são os próprios usuários que geram o conteúdo disponível. Nesse formato, usuários quando possuem alguma dúvida, ao invés de postarem uma pergunta e terem que aguardar uma respostas, podem primeiramente pesquisar na plataforma se sua dúvida já foi postada por outro usuário e inclusive respondida, economizando dessa forma seu tempo.

As plataformas de perguntas e respostas vão alem de um local onde perguntas são respondidas, os usuários além de compartilharem seu próprio conhecimento, compartilham experiências e opiniões. De maneira geral, é responsabilidade dos própiros questionadores avaliarem as respostas dos outros usuários e filtrarem as informações que melhor satisfa-

zem suas necessidades [**?**]. A melhor resposta, avaliada pelo autor da pergunta, fica em destaque e marca o tópico da pergunta como solucionado.

3.2 Estratégias e características

Essas plataformas possuem uma estratégia de pesquisa diferente e são mais específicas do que os motores de buscas tradicionais, enquanto o google, por exemplo, busca referências de todos o tipos e em toda a web, essas plataformas disponibilizam respostas personalizadas para perguntas indiviuais pesquisadas ou postadas pelos próprios usuários. Em motores de buscas como Yahoo!, Google e Bing os usuários recebem como resposta uma lista de sites e precisam garimpar entre eles as informações que consideram relevantes. Já nas plataforemas de perguntas e respostas os usuários apenas precisam escolher, em sua concepção, a melhor resposta para a pergunta que postou ou pesquisou.

Como característica da web 2.0, esses sites ganharam um carater social, ou seja, essa plataformas que no início eram utilizadas apenas como ponto para compartilhamento de informações (fóruns tradicionais) com usuários postando perguntas e respostas, decidiram introduzir páginas para cadastramento de um perfil completo dos usuários, onde eles podem registrar diversas informações pessoais, como nome completo, idade, redes sociais, preferências de estudo, entre outras informações. Assim os usuários podem conhecer um pouco mais uns aos outros. Alem disso, como uma forma de aumentar a comunicação e interação entre os usuários, essas plataformas adicionaram chats como outra funcionalidade. Os chats permitem uma maior interação entre os usuários, quando o contato através do topicos de perguntas e respostas não é suficiente.

Existem vários sites que adotam esse tipo de plataforma, alguns deles são Yahoo! Respostas, Amazon's Askville, Quora e Mind the Book. A propria Google, que é a lider em busca de conteúdo online, também possui um serviço de Q&A, o Google Answres, que falhou e foi descontinuado. Muitas razões foram levantadas para justificar esse insucesso. Uma dos principais fatores que influenciam para o sucesso de uma plataforma de perguntas e respostas é a participação do usuários [**?**]. Constatando a importância desse fator, essas plataformas começaram adotar estratégias que motivassem e tornassem cada vez mais agradável a particiapação dos usuários. Desta forma, com a participação constante dos usuários, o sistema, como citado no inicio do capítulo, se torna auto sustentável e sua chance de sucesso é maior. Segue algumas estratégias de gamificação adotadas para motivar a particiapção dos usuários:

- Reputação através de pontuação: Usuários recebem pontos por sua participação e pela qualidade dessa participação. Quanto maior sua sua pontuação, maior sua reputação no site.

- Privilégios: Usuários recebem privilégios de acordo com alguns fatores, que podem variar, mas são adiquiridos principalmente com base na reputação. Exemplos desses privilégios são: Permissões para editar e excluir perguntas e respostas.
- Badges (Medalhas): Além da pontuação recebida, usuários também recebem medalhas como reconhecimento a sua participação no site.

Como facilitador para utilização do site, essas plataformas contam com serviço de busca eficiente. Além de filtros para realizar pesquisas persoalizadas, adotam também sistemas de TAG's. Todas perguntas são marcadas com suas respectivas áreas. TAG'S são marcadores utilizados pelos usuários em suas perguntas. Uma pergunta sobre como implementar um web-service rest, por exemplo, pode ser marcada com TAG's como: java, web-service e JBoss. Essas TAG's são temas que tem haver com a implementação de um Web-service REST. A marcação com TAG's facilita para os usuários a procura por conteúdos específicos, ou seja, clicando em determinada TAG topicos marcados com a mesma TAG serão apresentados.

3.3 StackExchange

No início, essas plataformas abordavam todos os tipos de conteúdos, ou seja, os usuários compartilhavam informações sobre qualquer assunto. Após algum tempo, viu-se a necessidade de criar plataformas no mesmo seguimento, com as mesmas funcionalidades, porém mais específicas. Novos Q&A sites surgiram, focando em conteúdos específicos, como matemática, esportes, programação etc. Um exemplo muito interessante, de extremo sucesso e que serviu como base para o desenvolvimento desse trabalho é o StackExchange.

O StackExchange é uma plataforma voltada para unir pessoas com o mesmo interesse. Ela é subdividida em comunidades, cada um com o seu foco específico em determinado assunto, onde cada comunidade dessa possui seu próprio domínio e plataforma de perguntas e respostas. Cada comundiade dessa é autosuficiente, no sentido em que cada uma delas possui sua própria base usuários, administradores e moderadores, regras, e base de dados baseado no assunto tratado. Esses dados não estão necessariamente atrelados ao StackExchange, que atua como gerenciador maior dessas plataformas, dando suporte a correção de bugs e melhorias na plataforma, usada pelas comunidades.

3.3.1 StackOverflow

O StackOverflow é uma comunidade de Perguntas e Respostas do StackExchange, que tem seu conteúdo voltado exclusivamente para a programação e faz parte de uma rede de sites desse tipo chamado Stack Exchange. O StackOverFlow é lider na web quando

se trata de sites Q&A voltados para programação. Ele conta com uma base solida de usuários que participam ativamente do site, imersos pelas estratégias de gamificação e que mantém o site atualizado. Além disso, diferentes das outra plataformas, o site possui um ambiente focado em perguntas e repostas objetivas, conteúdos considerados não objetivos, são editados e podem ser até excluídos. Esse é um dos principais fatores para o sucesso da plataforma, pois condiciona o site a ter em seu banco de dados um conteúdo de perguntas e respostas totamente objetivo o que facilita a busca por informações.

3.3.2 API

A plataforma possui sua própria API (Application Provider Interface), que possibilita integrações de outros sistemas com o StackExchange, onde se tem acesso a informações muito interessantes. No escopo do nosso trabalho, pensamos em validar as respostas dos nossos usuários conforme sua reputação no StackOverflow, adquirindo uma credibilidade inicial para montarmos nossa própria base de dados e de usuários.

Capítulo 4

e-Design Patterns

Conforme abordado no Capítulo 4, o uso de design Patterns por programadores em determinado contexto no projeto de softwares não é tarefa trivial e, na maioria das vezes, requer um certo nível de experiência. Além de conhecer os padrões disponíveis, os desenvolvedores devem saber identificar em qual contexto aquela solução se aplica. O aplicativo e-Design Patterns foi idealizado justamente para minimizar esses requisitos de experiência, buscando facilitar o entendimento sobre o conceito de DesignPatterns, os padrões disponíveis e e como aplica-los em seus projetos. O e-Design Patterns é um protótipo de aplicativo para smartphone, tablets e browser no molde das plataformas sociais de Perguntas e Respostas.

4.1 Motivação

Desenvolvedores enfrentam uma problemática prática em muitos de seus dias de trabalho. Seja por falta de experiência, ou por estarem enfrentando desafios novos, muitas vezes não é claro qual padrão de projeto é mais adequado à solução do problema em vigor.

Em nossa pesquisa, não encontramos um aplicativo ou sistema parecido com o que propomos. Existem aplicativos que atuam como referência ao conteúdo de padrões de projeto, porém não fornecem nenhuma interatividade com a pessoa interessada, e o conteúdo muitas vezes se mostra denso e nada prático para aplicar na rotina corrida de um desenvolvedor.

Buscamos fornecer com o nosso aplicativo um ambiente onde o usuário possa consultar sucintamente o conteúdo que desejar e interagir com o sistema de forma inteligente e perguntar a usuários mais experientes e tirar dúvidas específicas.

4.2 Descrição do Aplicativo

O projeto do nosso aplicativo visa resolver essa carência descrita acima, e mudar o jeito como programadores e estudantes de computação interagem com essa temática dos design patterns. Desejamos que seja fácil reconhecer e aplicar os padrões para o seu caso de uso, ajudando a comunidade a encontrar as melhores soluções pros seus contextos específicos. Além de criar uma plataforma para a interação de usuários envolvidos em padrões de projeto, nos atentamos a tornar o nosso software um software dito educacional, conceitualmente. Por isso, pensamos nas funcionalidades necessárias para atender os nossos objetivos.

A plataforma reuniria programadores e entusiastas em padrões de projeto em um lugar feito especialmente com esse propósito, de se discutir e consultar conteúdos nessa temática. Teríamos conteúdos atualizados sobre os diversos ramos dos padrões de projeto; perguntas classificadas por tags para facilitar sua busca e indexação; interatividade entre os usuários; classificação de respostas e reputação dos usuários; conteúdo alimentado pelos próprios usuários...

As possibilidades são muitas.

- User Story #1:

  Eu, programador de uma empresa de software, estou tomando as decisões iniciais de como começar nosso novo projeto. Irei lidar com várias instâncias de classes que gerenciarão o comportamento do sistema, e sei que não posso de maneira alguma ter mais de uma instância viva de cada classe dentro do meu programa. Estamos programando em Java, e não sei muito bem como organizar a arquitetura do nosso projeto para este fim.

4.2.1 Core Features

Definiremos aqui as principais funcionalidades para a plataforma eDesignPatterns funcionar do jeito que foi concebida. Realizamos, com orientação da Profa. Dra. Letícia, que deveríamos nos atentar como a plataforma se tornaria um ecosistema sustentável, com garantia da qualidade do conteúdo alimentado pelos usuários, e que motivasse a volta dos mesmos.

Com essas funcionalidades, concebidas durante a concepção do nosso projeto, teríamos uma base funcional para a plataforma, que nos proveria interação entre os usuários, organização dos conteúdos por assuntos e interesses do usuário, e uma espécie de gamificação através da reputação dos usuários.

Platafoma Q&A

Como vimos no Capítulo 3, uma grande tendência da web participativa são as plataformas de pergunta e resposta. São uma poderosa forma de organizar pessoas e organizar o conteúdo produzido por elas. Gera uma grande interatividade entre os usuários, que podem continuar uma discussão no post da questão discorrendo sobre várias ideias, até que a melhor resposta ou solução venha a tona e ganhe destaque no post.

Tags

O sistema de tags possibilitaria ao sistema a organização de conteúdo por setores independentes, e ao de classificar uma pergunta/resposta com todas as tags ali englobadas.

Isso classificaria as questões da aplicação de forma eficiente e abrangente, pois muitas vezes uma mesma pergunta engloba tópicos de diversas áreas diferentes. Como uma pergunta pode ter várias tags, não ficamos restrito a unicidade de classificação de uma pergunta.

Reputação e Rating

Para termos confiabilidade de que a resposta dos colaboradores esteja correta e com conteúdo de qualidade, necessitamos de algum mecanismo que classifique os usuários e as respostas, respectivamente, de acordo com suas contribuições.

A reputação é a funcionalidade que validaria a trajetória dos nossos usuários. Com base nas contribuições e na colaboração provida por alguém, os outros usuários podem avalia-lo, aumentando ou diminuindo a reputação em questão.

O rating diz respeito a melhor resposta pra tal questão. Como é comum nas plataformas Q&A, principalmente em programação, ocorre de uma questão ter várias respostas igualmente certas de diferentes abordagens. Com essa feature, permitiríamos que os próprios usuários classificassem a resposta que melhor atendeu as suas necessidades.

Conteúdo de Design Patterns

A fim de caracterizarmo-nos como um aplicativo educacional, disponibilizaríamos um conteúdo pragmático, atualizado e completo sobre design patterns, disponível aos usuários para consultas rápidas e eficientes. Além de explicarmos o conteúdo, teríamos exemplificações em diferentes linguagens para consolidar melhor o conteúdo na cabeça do usuário.

4.2.2 Features desejáveis

Com as core features implementadas e devidamente testadas, poderíamos começar a expandir nossa ideia inicial do aplicativo, adicionando novas features para melhorar o aprendizado do aluno, assim como engajando-o de diferentes maneiras.

Submissão de conteúdo por usuários comuns

Uma forma que pensamos para aumentar a interatividade do usuário com a plataforma seria termos uma seção de conteúdo alimentado pelos próprios usuários. Design patterns é uma área muito mutável e sujeito a constantes inovações, criando-se diferentes formas de organizar projetos a todo tempo.

Exercícios interativos sobre o conteúdo

4.3 Processo de Software

Em um projeto de software, é necessário seguir um conjunto de passos, que servem como um roteiro para direcionar o rumo do desenvolvimento e para que o resultado tenha alta qualidade. Segundo Pressman [**?**], o processo de software é uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade. Existem alguns desse modelos e a escolha de um deve ser feita de acordo com o software a ser desenvolvido e os requisitos levantados, mas é certo que todos o modelos oferecem ao projeto de software em diferentes níveis, estabilidade, controle e organização.

4.3.1 Modelo Evolucionário

O modelo de processo evolucionário, é uma abordagem interativa, onde é possível desenvolver versões de software cada vez mais completas com base em versões iniciais. Essa abordagem é muito utilizada, pois permite o desenvolvimento do software mesmo com todas as mudanças que podem ocorrer nos requisitos, regras de negócio, funcionalidades etc. Essas mudanças durante todo o processo impedem que o desenvolvimento siga um planejamento linear, por isso o modelo evolucionário permite a criação de versões do software com base no que já foi estabelecido e, essas versões servem como ponto de partida para a continuação do software até o produto final.Existem dois modelos comum em processos evolucionários, a Prototipação, que foi o modelo escolhido para a construção do e-DesignPatterns e, o modelo espiral [**?**].

- Prototipação: Considerando o aumento nos custos para o desenvolvimento de softwares e o crescente número de sistemas construídos que falharam em satisfazer as

necessidades dos clientes, as organizaçõe estão utilizando cada vez mais a prototipação. A prototipação é uma implementação parcial de um sistema que serve para que clientes e desenvolvedores aprendam mais sobre o produto final, ou seja, possibilita uma melhor compreensão das necessidades que devem ser atendidas. Um protótipo permite que clientes avaliem o que já foi desenvolvido até o momento e forneçam um retorno (feedback) que servirá para aprimorar os requisitos até que seja alcançado o software final [**?**].

4.4 Prototipação do e-DesignPatterns

O processo de prototipação do e-DesignPatterns passou por 4 etapas:

1. Brainstorming: Nessa primeira etapa o grupo de desenvolvimento se reuniu e fez o leventamento de todas as idéias para o software. Além do levantamento de requisitos, foram tratadas questões de design, estratégias de implementação, delagação de tarefas, roteiro de desenvovimentos, metas, público alvo e ferramentas a serem utilizadas.

2. Produção de Wireframes: Na segunda etapa, todas a idéias levantadas no brainstorming foram utilizadas na produção dos wireframes, ou seja, esboços das principais telas do aplicativo que visam estabelecer as melhores estratégias para que a experiência do usuário com o software (UX) sejam as melhores possíveis: usabilidade, acessibilidade e prazer proporcionado na interação entre o usuário e o produto. O grupo optou em realizar o desenvolvimento dos wireframes em papel como estratégia de acelerar o processo e não ter que utilizar mais um software de apoio. Não foram criados wireframes para todas as telas, somente para as principais. Como estratégia de poupar tempo, os wireframes foram desenvovidos somente para as telas que ofereciam um maior desafio.

3. Design das telas: Nesta etapa, com base nos wireframes desenvolvidos e nas estratégias estabelecidas, utilizamos a ferramenta Figma para design final de todas a telas, inclusive aquelas que não tiveram wireframes relacionados.

4. Interação entre as telas: Na quarta e última etapa, utilizamos uma ferramenta de design como o figma, mas que também permite criar interações entre as telas, o Marvel. As telas desenvovidas no Figma, na 3 terceira etapa, foram exportadas para arquivos no formato .png e em seguida foram importadas para o Marvel, que foi utilizado para criar as devidas interações entre as telas.

4.5 Ferramentas Utilizadas

As seguintes ferramentas foram utilizadas no projeto do e-DesignPatterns, tanto para a prototipação do aplicativo como para comunicação e organização.

4.5.1 Slack

Slack [**?**] é um ferramenta colaborativa que permite a comunicação entre usuários de um grupo (time) de maneira fácil e interessante. Para utilizar a plataforma basta criar um time e convidar outros participantes para compo-lo. Alem da comunicação por mensagens em tempo real a ferramenta também permite a criação de canais que podem ser utilizados para que os usuários do time tratem de assuntos diferentes. Diversas outras funcionalidades são oferecidas pela plataforma, uma muito interessante é a possibilidade da integração com outras ferramentas, como o gitHub e o Marvel. A ferramenta também oferece notificações instantâneas sobre atividades realizadas no grupo, permite mensagens privadas, compartilhamento de arquivos entre outros funcionalidades.

4.5.2 Git e GitHub

Git [**?**] é um sistema de controle de versões distribuídas e gerenciamento de código, com ênfase em velocidade. É a ferramenta lider no mercado, pois é extremamente eficiente e robusta. Para utilizá-lo é preciso criar um repositório local ou remoto onde os arquivos serão armazenados e versionados de acordo com cada utilização. Para este trabalho, escolhemos utilizar um repositório remoto para que todos os integrantes do grupo tivessem acesso. O repositório escolhido foi o GitHub, também muito eficiente e de larga utilização A opção por um repositório remoto foi fetia para que todos os componentes do grupo tivessem acesso. Nesse repositório armazenamos os arquivos latex do trabalho escrito.

4.5.3 Figma

Figma [**?**] é uma ferramenta online de design de interface e que permite colaboração em tempo real. A ferramenta disponibiliza a criação de times para o desenvolvimento colaborativo de interfaces, oferece controle de versões e designs responsivos entre várias outras funcionalidades.

4.5.4 Marvel

Marvel [**?**] é um ferramenta colaborativa de design e prototipação e está disponível para web e smartphones. Oferece ferramentas para criação de telas mas também permite

importar imagens do Sketch ou Photoshop, além de permitir a sincronização com cloud storages. Sua simples interface de edição permite lincar todos os designs juntos e adicionar gesture e transiçoes para tornar o seu prototipo fiel com um aplicativo real ou um website.

4.6 Telas e suas funcionalidades

Capítulo 5

Avaliação

5.1 Ficha de Avaliação de Software Educacional

É consenso entre vários autores que a qualidade é uma meta no desenvolvimento de software a ser atingida. Uma das principais medidas é satisfazer as necessidades dos usuários além de atender determinadas propriedades e características da aplicação de acordo com seu contexto, ou seja, existem diferentes dimensões de qualidade e a importancia dos atributos ou características do software é variável segundo o domínio da aplicação [**?**].

A análise da qualidade de um software, além de abordar a base pedagógica, ou seja, a concepção teórica que descreve como o sujeito aprende, como ele se apropria e constrói seu conhecimento deve levar em conta o ponto de vista técnico uma vez que estes aspectos orientam para uma adequada utilização. Nesse sentido fichas de avaliação de softwares educacionais são propostas para auxiliarem no processo de avaliação do software [**?**]. Dessa forma propomos a seguinte ficha para avaliação do e-DesignPatterns:

Capítulo 6

Conclusão

Para que uma plataforma social de perguntas e respostas obtenha sucesso, é ncessário que além de implementar suas funcionalidades adequadamente utilize de recursos e mecanismos que motivem os usuários a se manterem participativos na plataformas, como a gamificação e outra formas de ludificação, tornando assim o ambiente virtual muito mais atrativo e agradável. Com isso a plataforma se torna auto sustentável com seus usuários gerando e gerenciando o contúdo.

O protótipo do eDesignPatterns segue um modelo que já vem obtendo sucesso no mundo web tanto em funcionalidade como implementando esses recursos e mecanismos de ludificação e motivação dos usuários.

6.0.1 Trabalhos Futuros

- Implementação da aplicação
- Integração com a API do StackExchange
- Ampliação para uma aplicação web