

UNIVERSIDADE FEDERAL DO PARANÁ

LARISSA YURI UEDA

**FERRAMENTA PARA CRIAÇÃO DE CARTÕES
CLASSE-RESPONSABILIDADE-COLABORAÇÃO**

CURITIBA

2016

LARISSA YURI UEDA

**FERRAMENTA PARA CRIAÇÃO DE CARTÕES
CLASSE-RESPONSABILIDADE-COLABORAÇÃO**

Trabalho de Graduação apresentado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal do Paraná.

Orientador: Prof. Dr. Andrey Ricardo Pimentel

CURITIBA

2016

“Não se pode escrever nada com indiferença.”

Simone de Beauvoir

RESUMO

O Cartão CRC (Classes, Responsabilidades e Colaborações) é uma ferramenta que foi elaborada para auxiliar na didática de programação orientada a objetos e posteriormente adotada como técnica para modelagem de software[2]. Vem sendo utilizada por equipes de desenvolvimento ágil como parte da documentação do software, a fim de proporcionar uma análise sistemática do problema e/ou entender as funcionalidades de um código já implementado[13]. Este trabalho tem como objetivo apresentar uma ferramenta criada com o intuito de agilizar a criação dos cartões, na qual é possível criar, arrastar e editar as fichas em ambiente web. Através de uma solução online, o projeto visa a elaboração de uma ferramenta intuitiva e auxiliadora no ensino ou sessões de modelagem CRC.

Palavras-chave: Cartões CRC.

ABSTRACT

The CRC(Class-Responsibility-Collaborator) Card is a tool that was designed to assist the object-oriented programming teaching and was later adopted as a technique for modeling software[2]. It has been used by agile development teams as part of software documentation in order to provide a systematic analysis of the problem and/or understand the features of a code already implemented[13]. This work aims to present a tool created in order to speed up the creation of the cards, which can create, drag and edit records in a web environment. Through an online solution, the project aims to develop an intuitive and support tool for classes or sessions of CRC modeling.

Keywords: CRC Card.

SUMÁRIO

1	INTRODUÇÃO	6
1.1	OBJETIVO GERAL E ESPECÍFICO	6
1.2	JUSTIFICATIVA	7
1.3	APRESENTAÇÃO DO DOCUMENTO	7
2	REVISÃO BIBLIOGRÁFICA	9
2.1	CARTÕES CRC	9
2.2	Modelagem de Software usando Cartões CRC	10
2.3	METODOLOGIA ÁGIL	13
2.4	CARTÕES CRC E A METODOLOGIA ÁGIL	14
2.5	TRABALHOS RELACIONADOS	15
3	A FERRAMENTA	18
3.1	TECNOLOGIA	18
3.2	OBJETIVOS	18
3.3	PRINCIPAIS FUNCIONALIDADES E ARQUITETURA	18
3.4	TELAS	19
3.5	USOS	19
4	EXEMPLO DE USO	21
5	CONCLUSÃO	24
5.1	TRABALHOS FUTUROS	24
	REFERÊNCIAS	26

CAPÍTULO 1

INTRODUÇÃO

A Engenharia de Software foi criada pela necessidade de estabelecer determinações para processos de execução de software. Esta é utilizada para a criação de métodos e princípios para a projeção e construção de programas com qualidade e agilidade [3]. Pressman [11] definiu como elementos fundamentais para a aplicação da Engenharia de Software no cenário contemporâneo os métodos, que detalham o projeto, as ferramentas, que automatizam o processo, e os procedimentos, que ligam o método à ferramenta.

A técnica Classe Responsabilidade Colaboração (CRC) foi criada originalmente para ensino de programação orientada a objetos para desenvolvedores do paradigma de programação estruturada [2]. Foi posteriormente incorporada como modelagem, pois mostra-se eficiente em descrever as necessidades e funcionalidades das funções do software. A técnica inclui uma sessão CRC, que produz cartões com informações de nome, responsabilidades e colaboradores das classes [1]. Portanto, cada cartão corresponde à uma classe, que deve ter um propósito que possa ser definido breve e claramente. O nome deve descrever adequadamente sua abstração, as responsabilidades podem descrever seus serviços para outras classes ou suas ações, e os colaboradores são outras classes necessárias para cumprir suas funções [4]. O objetivo dessa modelagem é prover uma análise sistêmica suficiente para o desenvolvimento de um software que atenda os requisitos do usuário.

1.1 OBJETIVO GERAL E ESPECÍFICO

O objetivo do projeto é oferecer uma alternativa digital para a produção de cartões CRC. O criador dinâmico de CRC foi pensado inicialmente com o intuito de auxiliar nas aulas de Engenharia de Requisitos do curso de Ciência da Computação da Universidade Federal do Paraná. O escopo inicial apresentava-o como um objeto educacional, entretanto, foi modificado para uma proposta mais genérica. Esta alteração ocorreu devido à

possibilidade de utilizá-lo também para documentação de software.

1.2 JUSTIFICATIVA

A modelagem CRC estabelece um padrão de modelo para definir as operações, funcionalidades e restrições do sistema, porém distoa por ter uma característica marcante: são cartões físicos. O uso de ferramentas de apoio ganhou espaço na Engenharia de Software pois fornecem suporte, aumentam a eficiência e facilitam várias fases do processo de criação de software[5]. A modelagem baseada em cartões de papel diverge, portanto, do padrão atual de documentação, que é baseada em documentos digitais e/ou compartilháveis. Além disso, treinamentos de modelagem CRC podem produzir vários cartões que não tem utilidade e serão descartados. Este projeto foi elaborado com o intuito de oferecer uma ferramenta alternativa para o método da modelagem CRC, visando a expansão da aderência à esta metodologia no cenário atual de tendências de desenvolvimento e projeto de software.

1.3 APRESENTAÇÃO DO DOCUMENTO

Neste documento será apresentada a Revisão Bibliográfica no Capítulo 2, com três subcapítulos: Cartões CRC, Metodologia Ágil, Cartões CRC e a Metodologia Ágil e Trabalhos Relacionados. O primeiro descreve as principais características da modelagem e a técnica empregada em uma sessão CRC, com detalhes de como deve ser conduzida e pontos importantes. O segundo descreve um pouco da história da Metodologia Ágil, que vem sofrendo expansões e aderências, e também do que é necessário em uma equipe para implantá-la. O terceiro subcapítulo descreve como a metodologia do primeiro pode ser usada por equipes de desenvolvimento ágil, sendo utilizada para levantamento de requisitos de processos complexos ou integrada à documentação. O Capítulo 3 descreve a ferramenta criada, contendo subcapítulos intitulados Tecnologia, Objetivos, Telas e Usos. O capítulo também contém como o projeto foi desenvolvido, quais as razões dele ter sido realizado, o que foi incluído nele e como pode ser utilizado. O Capítulo 4 é o Exemplo,

ou seja, uma demonstração de como a ferramenta pode ser utilizada. O Capítulo 5 inclui as conclusões.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

2.1 CARTÕES CRC

A modelagem CRC é uma técnica com o intuito de determinar as necessidades do usuário. A técnica foi apresentada inicialmente para fins didáticos, pois facilita o entendimento do pensamento voltado a objetos. Ao invés de tentar fazer o ensino e o design de Orientação a Objeto ser o mais próximo possível do ensino de programação estruturada, o meio mais efetivo encontrado foi remover o material familiar e estruturar uma nova técnica [2]. Vem sendo utilizada também durante o processo de desenvolvimento por programadores do paradigma de orientação a objeto (OO) e destaca-se por ter abordagem simples e direta [1].

A metodologia utiliza fichas comuns divididas em três partes: nome da classe, responsabilidade e colaboradores. A classe é a estrutura que abstrai um conjunto de objetos similares, e seu nome deverá ser estar indicado no topo do cartão. As responsabilidades são obrigações ou conhecimento de uma classe e devem ser listadas no lado esquerdo do cartão. Os colaboradores correspondem a outros cartões necessários para que suas responsabilidades possam ser executadas quando não possui o conhecimento necessário, e localiza-se no lado direito do cartão. Um modelo de cartão é apresentado na Figura 2.1

Nome da Classe	
Responsabilidades	Colaboradores

Figura 2.1: Modelo de Cartão CRC

Fonte: A autora(2016)

2.2 Modelagem de Software usando Cartões CRC

Uma equipe eficiente é necessária para que os cartões sejam elaborados com êxito. Na reunião de definição dos cartões deve estar presente pessoas que tenham domínio do negócio, um bom intermediador e indivíduos que tenham facilidade em comunicação através da escrita. Os especialistas no negócio deverão ser os usuários do sistema, aqueles que realmente detêm o conhecimento das necessidades e do que é necessário no sistema, sendo desejável quatro ou cinco pessoas deste grupo. Este grupo deve ter conhecimento, bom raciocínio e serem comunicativos. O facilitador é o coordenador da sessão, responsável por manter a fluidez e corretude do processo e dos cartões, auxiliar no foco e promover a organização. Devem saber dar continuidade nas sessões, compreender CRC e o processo. Algumas ações dos facilitadores podem aumentar a chance de uma sessão de cartão CRC bem sucedida, como avisar aos participantes a programação e um breve resumo da sessão. Também é necessário que todos os envolvidos entendam com clareza a técnica e obter o apoio dos administradores é um facilitador. O redator é a pessoa responsável por documentar tudo aquilo que transcende o conteúdo do que pode ser anotado no cartão, devem ser bons ouvintes e compreender rapidamente o que é discutido. Como participantes não compulsórios existem os observadores, que podem ser, por exemplo, pessoas que estão sendo treinadas para serem futuros facilitadores.

Para iniciar a sessão, é preciso arrumar a sala onde ela será realizada, para que todos

estejam confortáveis e o ambiente seja produtivo. Um modelo desse tópico é apresentado na Figura 2.2. O próximo passo é a troca de ideias, que pode ser um brainstorming, tendo como objetivo ambientar, discutir e sintonizar os membros, além de incentivar as ideias eficientes e criativas. A modelagem deve ser feita desenvolvendo os tópicos na seguinte ordem: 1-classes, 2-responsabilidades, 3-colaboradores, 4-cenários de uso, 5-distribuir cartões, 6-desenvolver protótipos.

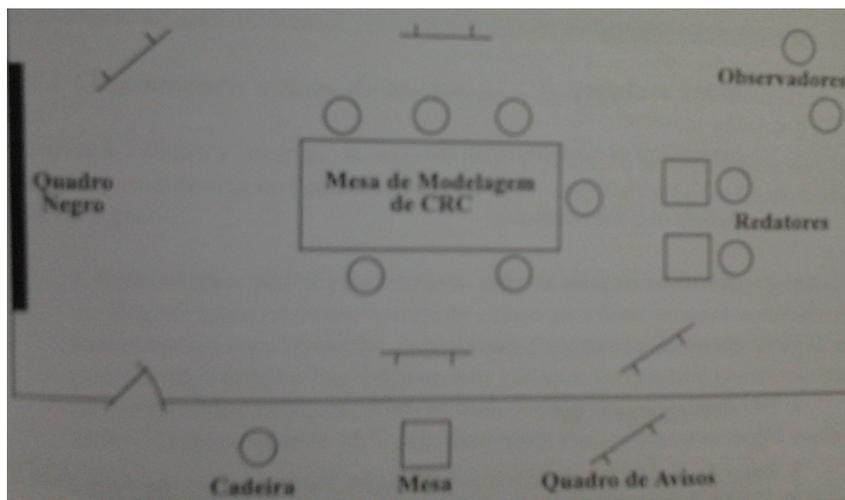


Figura 2.2: Modelo da Sala de Sessão CRC

Fonte: Ambler, 1997

As classes podem ser as que interagem com o sistema ou pertencentes a ele, consumidores, relatórios e telas, nomeadas com uma ou duas palavras, sendo cada uma em um cartão. Também podem ser identificadas ao se seguir o fluxo de dinheiro. As classes podem ser dos tipos: Atores, que representam cartões ou organizações que relacionam-se com o sistema; Negócios, indicando variando entre eventos lugares, conceitos e objetos; Interface, sendo as telas e menus, podendo relacionar funcionalidades complexas a esta classe; Relatórios, criadas para processamento e impressão de saída de dados, como listas. Um sistema normalmente possui 3-5 classes principais e é de grande ajuda identificá-las durante o processo.

A próxima definição são as responsabilidades, que são as ações da classe, suas funcionalidades e conhecimentos. Também pode-se identificar responsabilidades e depois atribuí-las à classe. A terminologia nessa fase deve ser compatível com a do usuário ao

invés do técnico, dando mais sentido aos cartões pois melhor representam as características de quem manuseia o sistema. Nessa fase é aceitável identificar responsabilidades que não serão implementadas no sistema.

Depois definem-se os colaboradores, que serão outras classes que esta necessita para completar suas responsabilidades, quando não possui o que é necessário. Também podem ser responsabilidades que incluem modificação de informações que pertencem à outra classe, e muitas vezes o colaborador é quem executa a maior parte da ação. Nesta etapa, responsabilidades podem ser adicionadas em outros cartões caso haja identificação da necessidade de colaborações que não foram identificadas na fase anterior.

Para os cenários de utilização, descrevem-se ambientes com fatores que o sistema deve conseguir resolver ou não, ou seja, os caminhos pelos quais o usuário pode interagir com o sistema. Neste passo, há a possibilidade de serem evidenciadas mais classes, responsabilidades e colaboradores. Os cenários podem ser descritos brevemente, representando apenas interações do ator com o sistema, entretanto, os redatores deverão anotar maior quantidade de detalhes. Os cenários serão utilizados posteriormente para validação dos cartões CRC [9], porém não fazem parte dos mesmos. Esses podem ser detalhados em pseudocódigos ou na própria linguagem [8], e podem ser desenvolvidos a partir de casos citados na fase anterior de troca de ideias.

Com os cartões prontos, eles devem ser distribuídos sobre uma mesa e sua proximidade é determinada através das similaridades e colaborações. É comum haver grande movimentação no início e sua redistribuição pode ser necessária várias vezes.

Os protótipos são esboços da interface e representam a aplicação. Devem ser elaborados simultaneamente à criação dos cartões, como na identificação de telas ou relatórios, e colados na parede. É preciso também determinar as responsabilidades da classe associada ao protótipo. Para avaliar o protótipo, componha o grupo com as pessoas que irão usar o sistema. O protótipo deve apenas conter funcionalidades que realmente farão parte do sistema final, podendo ser desenvolvido em alguma ferramenta com esse intuito.

Entre as vantagens da modelagem CRC está a qualidade da análise, pois são realizadas pelos usuários do sistema. Com a participação destes, a possibilidade do sistema ser mais

completo e atender melhor as expectativas é mais alta, pois são eles que entendem do que é necessário ou não para o cotidiano onde trabalham. Além disso, proporciona uma visão sistêmica do que será implementado e são muito coerentes com a orientação a objetos. Também há maior comunicação entre o setor de desenvolvimento e seu cliente, pois essa interação torna-se necessária para a modelagem. Entre as características físicas dos cartões, destacam-se o baixo custo, fácil manuseio e facilidade de transporte. Entretanto, o modelo apresenta limitações. É necessário reunir fisicamente todos os integrantes e nem todos os detalhes do sistema podem ser anotados nos cartões durante a reunião, além de ser necessária uma modelagem posterior, conectando as classes.

2.3 METODOLOGIA ÁGIL

A criação de software é, historicamente, uma atividade frequentemente caótica [7] e feita sem planejamento, no qual o código sofre as alterações conforme os erros aparecem ou as funcionalidades tornam-se necessárias. Este meio de implementação pode funcionar quando o sistema é pequeno, entretanto, torna-se um obstáculo para quando é maior ou cresce. As soluções criadas então foram as metodologias. Entretanto, o processo extremamente detalhado e burocrático criou tantos passos para os desenvolvedores seguirem que fez com que o ritmo de trabalho diminuísse. Essas metodologias ficaram conhecidas como "metodologias pesadas" [7].

Em resposta à essa metodologia, foi criado um tipo de metodologia que garante um número razoável de etapas, adaptáveis durante o desenvolvimento do projeto e orientadas às pessoas, e não ao processo. Fowler denominou essa metodologia de "Light Methods", podendo ser traduzida como "Métodos Leves". O termo popularizado, entretanto, foi "Metodologia Ágil" por um grupo de especialistas em processos de desenvolvimento de software nos Estados Unidos. Diversos métodos foram criados, porém todos estabeleciam os mesmos princípios principais, documentados no Manifesto Ágil, que foi estabelecido juntamente com a criação da Aliança Ágil [10].

O termo Desenvolvimento Ágil passou então a englobar os processos que valorizassem os indivíduos, as interações, o funcionamento e as respostas rápidas às mudanças acima

da documentação, das ferramentas, do planejamento e do processo [6].

Uma característica marcante da Metodologia Ágil é a adaptabilidade, substituindo o preditivo, um atributo pouco possível em desenvolvimento de software. Um erro muito grande é tentar seguir um planejamento que exige previsibilidade quando não há alguma, pois há grandes riscos do planejamento falhar e enormes consequências. A adaptabilidade vem como solução para, mesmo em algo imprevisível, instaurar controle através do desenvolvimento iterativo. Este fator é essencial na metodologia, pois garante a fluidez e a facilidade de mudança nas ferramentas durante o processo [7].

Executar um processo adaptativo, entretanto, é difícil. O foco na equipe, portanto, vem como um fator importante, pois é necessário que haja boa interação entre os participantes e que estes tenham boa capacitação. Logo, a comunicação constante entre os integrantes e com o cliente, bom trabalho em equipe e uma boa liderança tornam-se características cruciais [7].

2.4 CARTÕES CRC E A METODOLOGIA ÁGIL

A modelagem dos cartões CRC pode ter como escopo um caso de uso ou ser um caminho até o caso de uso. Muitas equipes de desenvolvimento ágil elaboram o cartão CRC quando modelam um caso singular de usuário ou interações importantes entre um grupo de objetos. Nem todos os casos de uso ou funcionalidades precisam ter uma sessão de modelagem, mas é importante, em funções novas ou complexas, refletir e documentar antes de começar a programar [13].

Na prática, manter os cartões e diagramas atualizados é uma tarefa difícil, mas é importante que haja algum controle e documentação, mesmo que não totalmente completos. Além disso, os cartões devem ser elaborados como resumos, com pequenas frases ou tópicos, sem grande detalhamento. O valor do cartão também se encontra na sumarização que ele provem pois, afinal, os detalhes estarão na análise, no código e nos testes.

Wirfs-Brock [13] reporta que também é possível utilizar os cartões com engenharia reversa, afirmando que a análise de código elaborado por outra pessoa pode ser transformada em cartões a fim de entender o que foi implementado. A autora também argumenta

que os cartões podem ser usados com a metodologia Test Driven Development (TDD), que tem como característica orientação a testes, tendo vários ciclos durante o desenvolvimento. Afirma que nada impede que a modelagem seja integrada a programação e aos testes, pois o cartão pode ser feito a partir de um esboço, que será refinado e complementado conforme a evolução do software.

Os cartões também podem ser usados como documentação mesmo depois da implementação estar completa, pois são de fácil entendimento e compreensão, provendo também muitas vezes uma introdução simples à uma modelagem complexa. É uma ferramenta que ajuda a raciocinar, explicar e explorar os essenciais das ideias.

2.5 TRABALHOS RELACIONADOS

The EasyCRC Tool [12] é uma ferramenta desktop que cria cartões CRC e tem a possibilidade de salvá-los no formato XML. Criada por Assaf Raman, a ferramenta é leve (apenas 438kb) e foi feita com o intuito de ser um objeto de ensino para orientação a objetos. Um exemplo de tela é mostrado na Figura 2.3. É disponibilizada gratuitamente¹ e foi desenvolvida usando a tecnologia .NET da Microsoft.

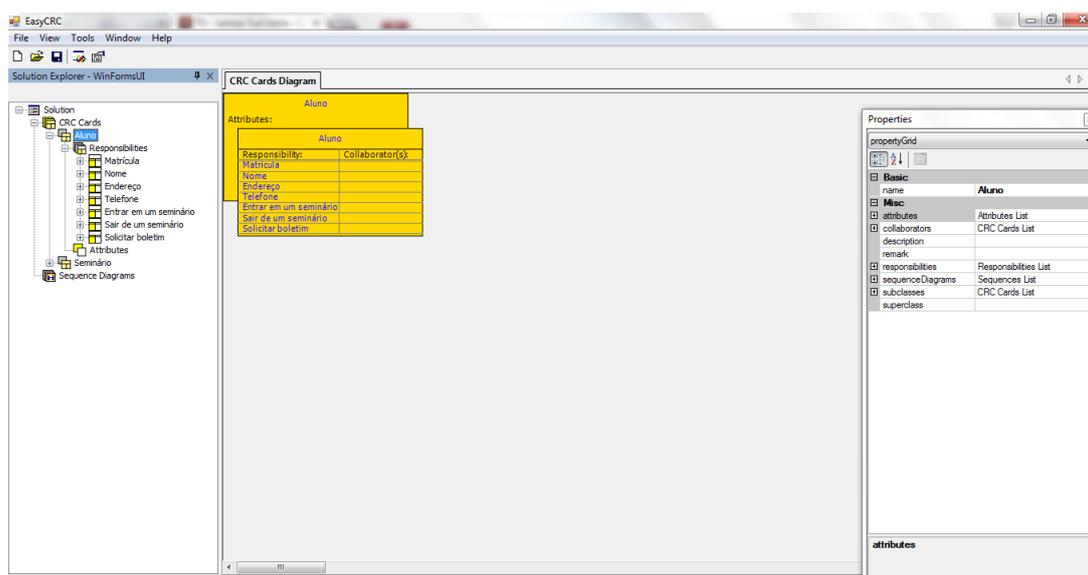


Figura 2.3: Exemplo de uso do Software EasyCRC

Fonte: A autora(2016)

¹Disponível em <https://sites.google.com/site/easycrc/>

O CRC Card Constructor ² é uma ferramenta online que permite criar um único cartão, sem possibilidade de exportação. A interface é simples e possui poucas funcionalidades, como mostrado nas Figuras 2.4 e 2.5

CRC CARD CONSTRUCTOR		
Class Name:	Aluno	
Class Type:	External Entities	
Characteristics:	Tangibility <input type="radio"/> Tangible <input type="radio"/> Abstract Inclusiveness <input type="radio"/> Atomic <input type="radio"/> Aggregate Sequentiality <input type="radio"/> Concurrent <input type="radio"/> Sequential Persistence <input type="radio"/> Transient <input type="radio"/> Temporary <input type="radio"/> Permanent Integrity <input type="radio"/> Corruptible <input type="radio"/> Guarded	
	Responsibilities: 1: Matricula do Aluno 2: Nome 3: Endereço 4: Telefone 5: Entrar em um seminário 6: Sair de um seminário 7: Solicitar boletim 8: 9: 10:	
	Collaborators: 1: Seminário 2: 3: 4: 5: 6: 7: 8: 9: 10:	
	Click below for assistance... <input type="button" value="Help"/>	
	<input type="button" value="Submit"/> <input type="button" value="Clear Form"/>	

Figura 2.4: Exemplo de cadastro no CRC Card Constructor

Fonte: A autora(2016)

Class Name: Aluno	
Class Type: External Entities	
Class Characteristics: Tangible, Atomic, Concurrent, Transient, Corruptible	
Responsibilities:	Collaborators:
Matricula do Aluno	Seminário
Nome	
Endereço	
Telefone	
Entrar em um seminário	
Sair de um seminário	
Solicitar boletim	

Figura 2.5: Exemplo de cartão criado no CRC Card Constructor

Fonte: A autora(2016)

O software Visual Paradigm é uma ferramenta que auxilia na construção de Diagramas

²Disponível em <http://groups.engin.umd.umich.edu/CIS/tinytools/cis375/f00/crc/crc.htm>

UML, Modelo Entidade Relacionamento, Cartões e outros. É uma ferramenta completa, com um custo de 39 dólares por mês (dado de Junho/2016) e com uma opção Trial que dura 30 dias. Bastante eficiente porém um pouco menos intuitiva de modelar, devido as suas várias opções e funcionalidade. Um demonstrativo de criação no software é apresentado na Figura 2.6

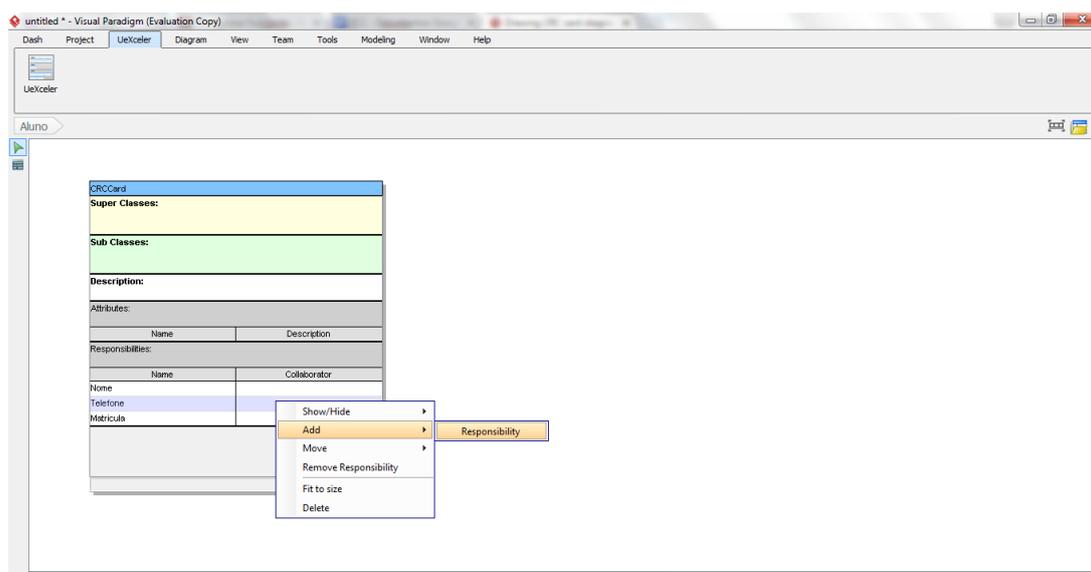


Figura 2.6: Exemplo de cartão criado no Visual Paradigm

Fonte: A autora(2016)

Como opções pagas sem opção de Trial temos o "CRC Cards for Software Design"³ e o "QuickCRC Windows"⁴

³ <<http://www.excelsoftware.com/crccards>>

⁴ <<http://www.excelsoftware.com/quickcrewin.html>>

CAPÍTULO 3

A FERRAMENTA

3.1 TECNOLOGIA

O projeto foi elaborado como página WEB, podendo ser compactado facilmente e hospedado em vários lugares. O layout foi desenvolvido em HTML e CSS, elaborado com o auxílio do framework da Google, o Materialize. O projeto foi codificado utilizando as linguagens JavaScript e sua biblioteca JQuery. Cada cartão criado é uma nova divisão do documento(div) que pode ser editada e arrastada.

3.2 OBJETIVOS

O objetivo da ferramenta é oferecer uma solução prática e online para a criação de cartões CRC. Foi elaborada visando uma interface simples e funcionalidades intuitivas. Além disso, o software não terá custos ao usuário, diferente da maioria das ferramentas de CRC atualmente disponíveis. Com essas características, espera-se que haja contribuição ao ensino dessa modelagem e maior facilidade no aprendizado e treinamento dela.

3.3 PRINCIPAIS FUNCIONALIDADES E ARQUITETURA

O principal fluxo da utilização ferramenta inclui as funções de criação, edição e movimentação. Para criar um novo cartão é necessário clicar no botão "Criar Novo". Este botão chama uma função implementada no arquivo javascript, criando uma nova divisão que conterà três partes distintas: Nome, Responsabilidades e Colaboradores. Se preenchidas, o cartão conterà as informações do formulário, mas nenhum dos campos é obrigatório, devido ao roteiro da sessão de cartão CRC que não impede a criação de todos os cartões primeiro para posterior preenchimento.

O cartão é editável em todas as partes, propriedade atribuída pelo HTML5 à tabela,

devido à possibilidade de adição de responsabilidades em um cartão quando estabelecidas outras colaborações, adequação de nomes das classes após a atribuição das características, entre outros acontecimentos comuns na elaboração dos cartões.

Por fim, a movimentação dos cartões é feita durante e após a criação dos cartões, visando o agrupamento por características comuns. A propriedade do jQuery que habilita arrastar uma divisão impede a edição e o contrário também. Portanto, é necessário que o cartão não esteja em modo de edição para movê-lo e esteja selecionado para editá-lo.

Todas as características dos cartões e do site, estão, portanto, implementadas em arquivos contidos na pasta do projeto. Nenhuma implementação foi feita considerando comunicação com servidor ou banco de dados, visando uma ferramenta concisa que possa ser facilmente incorporada em outros lugares. Como todas as alterações são feitas localmente, não há persistência no banco de dados e a atualizar a página limpa todos os dados gerados.

3.4 TELAS

A ferramenta possui três telas: Sobre, Cartão CRC e Criar. A primeira tela é uma breve introdução ao site. A segunda é uma explicação sucinta sobre os cartões, contando brevemente sua história e suas características. A terceira tela é a que cria os cartões, contém um formulário para inserir o nome da classes, suas responsabilidades e colaboradores. Também possui três botões: Criar Novo, que cria o cartão com as informações inseridas, Salvar, que permite salvar os cartões na máquina e Abrir, que abre um arquivo com cartões já salvos. Os últimos dois ainda não foram implementados.

3.5 USOS

O criador de cartões pode ser usado em simulações de sessões CRC, como em aulas ou treinamentos, onde a produção de cartões físicos seria desnecessária para documentação posterior. Portanto, também foi criada com o intuito de diminuir o desperdício de cartões que não serão necessários a longo prazo. Posteriormente, com a implementação dos

botões "Salvar" e "Abrir", será possível a utilização da ferramenta para sessões CRC que necessitem da documentação para posterior consulta.

CAPÍTULO 4

EXEMPLO DE USO

Para a demonstração de como criar cartões na ferramenta, teremos como exemplo uma classe Seminário e uma Aluno, sendo que a segunda depende da primeira para executar suas responsabilidades com êxito. Primeiramente, pode ser criada o cartão da classe Seminário, que deve saber o nome de cada Seminário registrado. Portanto, no nome da classe teremos "Seminário" e "Nome" como Responsabilidade.

Criador de Cartão CRC

The screenshot shows a web interface for creating a CRC card. At the top, there is a navigation bar with three tabs: 'SOBRE', 'CARTÃO CRC', and 'CRIAR'. The 'CARTÃO CRC' tab is active. Below the navigation bar, the form is titled 'Sobre'. It has a 'Nome da Classe' field with the text 'Seminário' entered. Below this, there are two sections: 'Responsabilidades' and 'Colaboradores'. The 'Responsabilidades' section has a text input field with 'Nome' entered. At the bottom of the form, there are three buttons: 'CRIAR NOVO', 'SALVAR', and 'ABRIR'.

Figura 4.1: Passo 1: Preenchendo o cartão Seminário

Fonte: A autora(2016)

Clicando então no botão "Criar Novo", teremos o cartão criado e seu nome é adicionado à lista de possíveis colaboradores.

Criador de Cartão CRC

SOBRE CARTÃO CRC CRIAR

Nome da Classe

Responsabilidades

Colaboradores

Seminário

CRIAR NOVO

SALVAR ABRIR

Seminário	
Nome	

Figura 4.2: Passo 2: Criando o cartão Seminário

Fonte: A autora(2016)

Então, deve-se preencher as suas informações do próximo cartão assim como no passo 1. O nome da classe no exemplo será Aluno, contendo informações como Nome e Telefone e também o campo Entrar em um Seminário. Para cumprir essa responsabilidade, deverá consultar o nome do seminário no cartão Seminário, portanto, deve adicioná-lo como Colaborador.

Criador de Cartão CRC

SOBRE CARTÃO CRC CRIAR

Nome da Classe

Aluno

Responsabilidades

Colaboradores

Seminário

CRIAR NOVO

SALVAR ABRIR

Seminário	
Nome	

Figura 4.3: Passo 3: Preenchendo o cartão Aluno

Fonte: A autora(2016)

Após clicar no botão Criar Novo, o cartão de Aluno será criado e pode-se editá-lo conforme a necessidade. A edição é acionada com um clique duplo no cartão, podendo ser percebida por uma borda leve de tom esverdeado em torno, e desabilitada clicando-se

fora dele.



Figura 4.4: Passo 4: Criando e editando o cartão Aluno

Fonte: A autora(2016)

Para arrastar um cartão é necessário que ele não esteja em modo de edição. Se ele está editável, portanto, é necessário clicar fora do cartão. Depois, é necessário clicar nele com o botão esquerdo e mantê-lo pressionado até o local em que deseja deixá-lo.

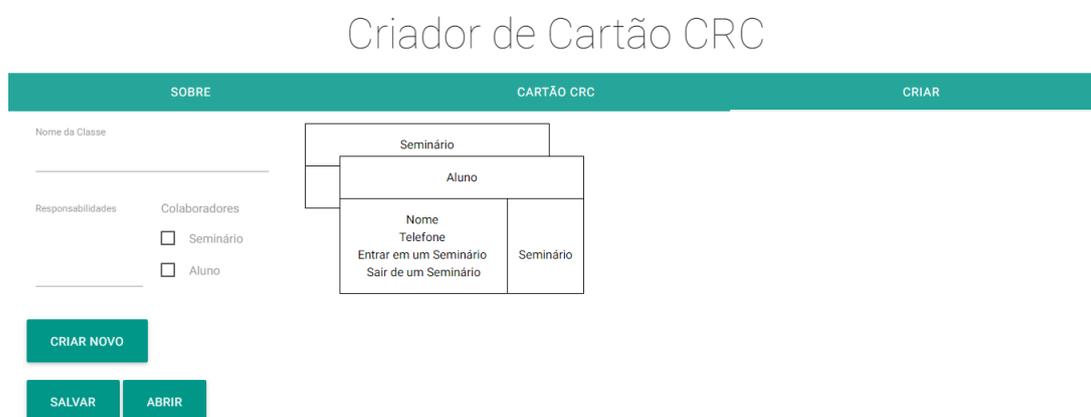


Figura 4.5: Passo 5: Movendo o cartão Aluno

Fonte: A autora(2016)

CAPÍTULO 5

CONCLUSÃO

As modelagens de software são importantes para uma melhor elaboração de projeto e também como aprendizado e esclarecimento para desenvolvedores. Os cartões CRC podem ser utilizados para fins didáticos, reforçando e ensinando características de programação orientada a objetos e relacionamento de classes, assim como para modelar e documentar funções de um novo produto ou mapear fluxos de um já finalizado. A necessidade de ferramentas para auxílio no desenvolvimento e documentação de software já é uma realidade, que vem sendo enfatizada com a evolução das metodologias e modelagens na área.

O criador de Cartões CRC online, portanto, foi elaborado com o intuito de auxiliar uma modelagem já existente, adequando-a à nova realidade que se vivencia em Engenharia de Software. A maioria das ferramentas da área incluem várias funcionalidades de outras modelagens, o que as torna completas porém complicadas e demandam investimento financeiro. Esta ferramenta tem um visual conciso e funcionalidades específicas, pois almeja um uso dinâmico e simples, que realmente facilite ao usuário atingir seu objetivo.

O trabalho também se mostrou importante no aprendizado de modelagens e metodologias, pois além de adquirir conhecimento sobre fatos históricos, também há muito conteúdo sobre novos estudos e tendências nos artigos utilizados. O aprofundamento no tema foi importantíssimo para um melhor entendimento da necessidade de entender corretamente o problema proposto pelo usuário e formas de abstraí-lo para documentação.

5.1 TRABALHOS FUTUROS

Algumas funções do software foram mapeadas, porém não foram implementadas, como o botão "Salvar" e o botão "Abrir", que posteriormente serão codificadas para gerar e ler arquivos na extensão XML. O modelo, então, poderia ser salvo e posteriormente visuali-

zado. Também haverá a possibilidade da criação e visualização de um grafo de colaboração para consultar o relacionamento entre os cartões. Através das responsabilidades e colaboradores de cada classe será possível determinar as dependências, facilitando a leitura do fluxo e funcionalidades do software. Com essas funcionalidades implementadas, o uso educacional da ferramenta será facilitado, pois será um incentivo e um meio mais prático de ensinar aos alunos sobre a modelagem, gerando os cartões com maior agilidade e demonstrando as relações com maior clareza.

REFERÊNCIAS

- [1] Scott W Ambler. *Análise e projeto orientados a objeto*. Infobook, 1997.
- [2] Kent Beck e Ward Cunningham. A laboratory for teaching object oriented thinking. *ACM Sigplan Notices*, volume 24. ACM, 1989.
- [3] Barry W Boehm. Software engineering-as it is. *Proceedings of the 4th international conference on Software engineering*. IEEE Press, 1979.
- [4] Jürgen Börstler e Carsten Schulte. Teaching object oriented modelling with crc cards and roleplaying games. *Proceedings WCCE*, volume 5, 2005.
- [5] Adalberto Cajueiro de Farias. Ferramentas case: suporte, adoção e integração. 1999.
- [6] Michel dos Santos Soares. Comparação entre metodologias ágeis e tradicionais para o desenvolvimento de software. *INFOCOMP Journal of Computer Science*, 3(2):8–13, 2004.
- [7] Martin Fowler. The new methodology. *Wuhan University Journal of Natural Sciences*, 6(1-2):12–24, 2001.
- [8] Ivar Jacobson. Object-oriented software engineering. 1992.
- [9] Morten Kyng e Joan Greenbaum. Design and work. *Hillsdale, NJ.: Lawrence Erlbaum.*, 1991.
- [10] Paula LO Libardi e Vladimir Barbosa. Métodos ágeis, 2010.
- [11] Roger S PRESSMAN. Engenharia de software., 2002.
- [12] Assaf Raman e Shmuel Tyszberowicz. The easycrc tool. *International Conference on Software Engineering Advances (ICSEA 2007)*. IEEE, 2007.
- [13] Rebecca Wirfs-Brock. *CRC Cards: An Agile Thinking Tool*. 2009.