



**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

DevBattle

ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

VERSÃO 1.2.0

**Augusto César Medeiros Franz
Ronaldo Albert Spranger
Wellington dos Santos Francisco**

**FLORIANÓPOLIS
2017**

Histórico de Revisões

Versão	Autor(es)	Data	Descrição
1.0.0	Augusto C.M. Franz Ronaldo A. Spranger Wellington dos S.F.	30/03/2017	Estabelecimento dos Requisitos
1.1.0	Augusto C.M. Franz Ronaldo A. Spranger Wellington dos S.F.	25/04/2017	Correção da versão 1.0.0 Diagrama de Interação Diagrama de Atividades
1.2.0	Augusto C.M. Franz Ronaldo A. Spranger Wellington dos S.F.	06/07/2017	Ajustes de requisitos e melhoria na clareza das regras do jogo

Sumario

- 1 - Introdução;
- 2 - Visão Geral;
- 3 - Requisitos de Software;
- 4 - Esboço da Interface Gráfica.

1 - Introdução

Objetivo:

O objetivo deste desenvolvimento tem como principal função a criação de um card game onde dois jogadores possam jogar. Agregando conhecimento a respeito de projetos e sistemas.

Definição de Palavras:

Baralho - Pilha de cartas viradas para baixo aonde poderão ser compradas as cartas aleatorias.

Descarte - Pilha de cartas descartadas pelos jogadores após cada rodada com a ultima carta virada para cima.

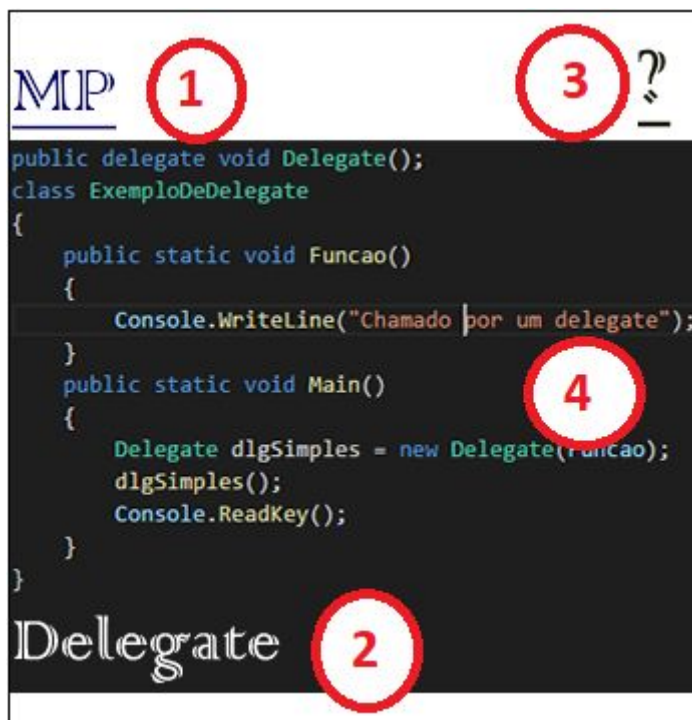
Turno - Intervalo de tempo que cada jogador faz sua jogada.

Bater - vencer a partida com a combinação correta de cartas e acertando os tipos de carta que ele usou para vencer.

Ajuda- Cada jogador tem direito a três ajudas, que quando acionadas revelam a linguagem e o tipo da carta selecionada em sua mão. As ajudas também contam como “vidas”, dado que quando um jogador tenta bater e não possuir as cartas necessárias e/ou não as identificou corretamente no ComboBox Linguagens o mesmo terá uma ajuda decrementada. Se o número de ajudas chegar a -1 o jogador perde o jogo.

Furar Batida - Isso acontece quando o jogador tenta **bater** sem a combinação correta de cartas na mão e/ou não acerta as linguagens nas listas de opções. O procedimento de batida será explicado na sessão **Batida** no fluxo de partida.

Carta - A carta é composta de 4 elementos que serão explicados ao lado do anexo:



1 - TIPOS : Paradigma/

Linguagens/ Engenharia - são 6 tipos no total nele são formados as sequências para vencer a:

1. **CP** - Ling. Compilada;
 2. **IT** - Ling. Interpretada;
 3. **MP** - Multiparadigma;
 4. **DB** - Banco de Dados;
 5. **LM/FE** - Linguagem
- Marcação/ Folha de Estilo;
6. **ENG** - Eng. Software

Mais detalhes na seção **Tipos de Cartas**.

2 - NOME DA CARTA - Definições/ Frameworks/ Processos/Métodos

O nome da carta diz onde ela se encaixa nos tipos para e que linguagens implemento a função dela. Ex: Delegate é multiparadigma, pois linguagens compiladas e interpretadas como python, java, c#, javascript, ruby, PHP e javascript implementam ela nativamente.

3 - Dica - Ajuda Quando o jogador clica em ajuda, a carta se revela apresentando um texto sobre a função da carta e as linguagens que a implementam.

Ex. "Delegate é um tipo de referência que pode ser usado para encapsular um nomeada ou um método anônimo. Delegados são semelhantes aos ponteiros de função em C++. Entretanto, os delegados são fortemente tipado e seguro. Implementadas em Python, Java, C#, Javascript, Ruby, PHP e Javascript."

4 - Imagem - Imagem mostra um trecho de código ou diagrama da função da carta.

Ex. Na imagem da carta acima a imagem da carta é um trecho de código C#, mas isso não significa que o Delegate pertence somente ao C#, ela pode ser usada para bater usando outras linguagens.

Explicação do jogo:

Fluxo da Partida:

- **Pré-jogo**

Cada Jogador começa o jogo com 7 cartas e 3 ajudas, a ordem dos jogadores será escolhido pelo protocolo do NetGames. Existe uma pilha com as cartas embaralhadas e viradas para baixo e também uma segunda pilha para o descarte.

- **Início e Turnos**

No primeiro turno a primeira carta é virada automaticamente do baralho para o descarte.

Após o primeiro turno os demais jogadores podem escolher entre comprar a carta do monte de descarte ou do baralho, seguindo o procedimento de turno e podendo: pedir ajuda, descartar e passar, descartar e bater.

- **Partida**

O jogo consiste em montar um projeto usando as cartas que são condizentes nos projetos de sistemas respeitando seu tipos e quantidades. Quando o jogador acreditar ter a sequência correta para vencer a partida. Ele deve selecionar as linguagens correspondentes das cartas que ele tem em mão no menu de linguagens (representado no sistema como 6 combobox). Não é necessário preencher as 5 combobox para vencer, mas 6 é o número máximo de tipos que ele pode conseguir.

- **Batida**

O processo de batida tem 2 etapas:

- Na primeira etapa jogador necessita ter os tipos correspondentes de cartas para se montar um sistema. Restrito a usar somente **uma linguagem de cada grupo**:

Exemplo de jogada válida: O Jogador possui as cartas

1-Ponteiro(C/C++) - **2**-Orientação Objeto(C/C++,Java,...), **3**-AngularJS(Javascript*), **4**-HTML(Coringa), **5**-CSS(**), **6**-WorkBench(SGBD/SQL), **7**-Views(BD Relacional);

Este é um exemplo de jogo válido pois ele usa 1 carta do Tipo Ling.

Compilada(C/C++), 1 carta Multiparadigma(Orientação a Objetos(C/C++, PHP, ...)), 2 cartas de LM/FE(HTML,CSS) e 2 cartas DB(SGBD/SQL, DB não relacional) e apesar de ter uma carta do Tipo Ling. Interpretada(Angular) o javascript se torna um coringa quando o jogador tem a carta html na mão. Mas em contrapartida é obrigatório ter uma carta Html para bater com javascript.

Exemplo de jogada invalida: O Jogador possui as cartas **1**-Ponteiro(C/C++) -

2-Orientação Objeto(C/C++,Java,C#,...), **3**- Threads(C/C++,Java,C#,...) **4**-Linq(C#), **5**-CSS(**), **6**-AngularJS(Javascript*), **7**-Views(BD Relacional);

Neste exemplo são três situações que invalidam o jogo. Linq(C#) e Ponteiros(C/C++) apesar de pertencerem ao mesmo grupo de linguagens compiladas, são linguagens diferentes assim como CSS e Javascript não podem ser usada para bater sem uma carta de HTML presente.

-Na segunda etapa o jogador necessita preencher as 1 ou 6 combobox correspondentes às cartas que ele tem na mão.

No primeiro exemplo ele teria da etapa um ele teria que selecionar as seguintes informações: 1-C/C++, 2-Javascript, 3-HTML, 4-CSS, 5-SGBD e 6 em branco.

No segundo exemplo caso o jogador fosse fazer um tentativa tira que preencher:

1-C/C++, 2-C#, 3-CSS, 4-Javascript, 5-BD Relacional e 6- Em branco.

-Passado as duas etapas o jogador seleciona uma carta para descarte não desconfigurando o seu jogo e usa a ação “Descartar e Bater”, caso tenha acertado ganha a partida. Caso erre, perderá um Ajuda, se não houver mais nenhum ajuda disponível, o jogador perde a partida.

Tipos de Cartas

Linguagens Compiladas

C/C++, Java e C#

1- Pode-se formar combo sozinhas

2 - Com JavaScript, HTML e/ou CSS

3 - Com XML

4 - Com os itens 2 e 3 mais BD e/ou Engenharia de Software

- **Linguagens Interpretadas**

PHP, Ruby, Python, Javascript*

- 1- Pode-se formar combo sozinhas
- 2 - Com JavaScript, HTML e/ou CSS
- 3 - Com os item 2 mais BD e/ou Engenharia de Software

- **Multiparadigma**

C\C++, Java e C#, PHP, Ruby, Python, Javascript*

- 1- Deve-se sempre respeitar a função correta da carta assim como sua linguagem.

- **Linguagem de Marcação/Folha Estilo**

HTML,XML*, CSS****

- **DB**

Relacional, Orientado a Documentos, Orientado a Grafos, SGBD/SQL

- 1- Necessitam de ao Menos uma linguagem de programação para funcionar

- **ENG Software**

Agile, UML, Cascata, Incremental

- 1- Necessitam de apenas uma carta de linguagem de programação ou HTML para funcionar

Número mínimo e máximo de cartas para vencer a partida:

Compiladas

0-7 C#, C/C++, Java

Interpretadas

0-7 Javascript, PHP, Python,Ruby

Multiparadigma

0-7 C#, C/C++, Java, Javascript, PHP, Python,Ruby

Linguagem de Marcação/ Folha de Estilo

0-7 HTML

0-4 CSS**, XML***

Banco de Dados

0-6 SGBD/SQL, Não Relacional, Relacional, Orientado a Grafos

Engenharia de Software

0-6 Eng. Software****

*combina com todos os grupos.

** Necessário HTML

*** Necessário Ling. Compilada, Interpretada ou do grupo de Multiparadigma.

**** Coringa, funciona com todos as cartas.

2 - Visão Geral

Arquitetura do Programa:

Arquitetura cliente-servidor , distribuído.

Premissas do Desenvolvimento:

- O programa deve ser implementado em Java, devendo executar em qualquer plataforma que disponha de máquina virtual Java;
- O programa deve apresentar interface gráfica bidimensional ;
- O programa deve executar distribuído ao NetGamesNRT.

3 - Requisitos de Software

Requisitos Funcionais:**Conectar**

- O programa deve apresentar o botão conectar em seu menu para estabelecer uma conexão com o seu servidor.

Desconectar

- O programa deve apresentar o botão desconectar em seu menu para finalizar a conexão com o servidor e assim terminar partida.

Iniciar

- O programa deverá iniciar uma nova partida somente se a conexão já estiver estabelecida e irá distribuir 7 cartas aleatórias do baralho para cada jogador da partida, virar a primeira carta do monte para o descarte e definirá o jogador que irá iniciar.

Efetuar jogada

- O programa deve apresentar qual jogador irá jogar no turno, permitindo somente que o mesmo compre do baralho ou a última da pilha de descarte, após a compra o jogador deverá selecionar uma carta de sua mão para enviá-la para a pilha de descarte podendo escolher se tentará bater ou passar sua vez .

Receber Jogada

- O programa deve identificar o término do turno do jogador adversário, atualizando então a interface do jogador atual: última carta do descarte e o status do jogo.

Ajuda

- O sistema irá possuir um botão ajuda que ao ser clicado irá revelar a linguagem da carta selecionada.

ComboBox Linguagens

- O sistema irá possuir um combo-box com todas as possíveis combinações de naipes que terá de ser selecionado de acordo com as cartas de sua mão para poder bater, será uma extensão de efetuar jogada.

Requisitos não Funcionais:

- **Requisito não funcional 1 - Especificações**
 - O sistema deverá possuir especificações baseado na segunda versão da UML.
- **Requisito não funcional 2 - Ajuda**
 - A funcionalidade de “ajuda” poderá ser usada apenas 3 vezes por partida.

4 - Esboço da Interface

