

# Projeto Detalhado de Software

## Universidade Federal de Goiás (UFG)

Kleudson Rodrigues  
Wallace Luis do Carmo

### Relatório - Roda a Roda

Este documento é responsável por registrar as atividades realizadas pelos responsáveis pelo projeto a fim de mensurar e detalhar o trabalho realizado pela equipe na construção do software Roda a Roda. O documento seguirá os requisitos do sistema estabelecidos a seguir pelo diretor do projeto.

#### Requisitos do sistema - Liberações

**L1:** Ter uma versão do jogo capaz de:

1. Escolher quantidade de jogadores: 1,2 ou 3;
2. Escolher a quantidade de palavras por etapa: 1, 2 ou 3;
3. Escolher a quantidade de etapas: 1, 2, 3, 4, 5, 6 ou 7;
4. Funcionar para um jogador, com uma palavra por etapa, com uma etapa somente. Para este caso o tema é animais e a palavra é cachorro.
5. Rodar a roleta. Não é necessário interface gráfica. A roleta tem 20 divisões, sendo:
  - a. duas divisões de perde tudo,
  - b. duas de passa a vez,
  - c. quatro divisões de 100 pontos,
  - d. quatro divisões de 200 pontos,
  - e. quatro divisões de 400 pontos,
  - f. duas divisões de 500 pontos,
  - g. duas divisões de 1000 pontos.
6. Informar se jogador é vencedor ou perdedor.

**L2:** Ter uma versão do jogo capaz de:

1. Funcionar para dois jogadores, com a quantidade de etapas selecionada e com a quantidade de palavras por etapa selecionada.
2. O tema disponível continua sendo animais, mas há pelos menos vinte e uma palavras (a sua escolha), que são lidas de um arquivo de texto. O jogo deve realizar um sorteio das palavras selecionadas por jogo. No futuro, deverá existir um arquivo para cada tema disponível.

**L3:** Ter uma versão do jogo capaz de:

1. Funcionar para três jogadores, com a quantidade de etapas selecionada e com a quantidade de palavras por etapa selecionada.

2. Funcionar com os seguintes temas: profissão, cidade e filme. Sendo um arquivo de texto para cada tema.
3. Usar diversas estratégias de sorteio da roleta. O compromisso inicial é fazer o sorteio da roleta usando Math.random, mas no futuro outras estratégias poderão surgir.

**L4:** A liberação 4 não é uma liberação de fato, mas os resultados produzidos até aqui são testados com o suporte do JUnit.

**L5:** O sorteio da roleta deve utilizar o padrão de projeto strategy. Duas estratégias são conhecidas: uma aleatória e outra viciada. Na viciada, sempre na segunda vez que o jogador girar, o resultado deve ser passe a vez. O padrão de projeto observer deve ser utilizado para tratar ações do jogo. Por exemplo, após escolher uma letra, observadores podem ser utilizados para avaliar se a letra é faz parte da palavra e se o jogo terminou.

	Data	Descrição	Informação Adicional
Kleudson Rodrigues Wallace Luis do Carmo	17 de agosto de 2015 - 23 de agosto de 2015	<ul style="list-style-type: none"> <li>• Inicialização do projeto;</li> <li>• Reunião referente aos requisitos;</li> </ul>	Foi estabelecido o primeiro contato com o problema a ser resolvido pela equipe
Kleudson Rodrigues Wallace Luis do Carmo	09 de novembro de 2015 - 15 de novembro de 2015	<ul style="list-style-type: none"> <li>• início do desenvolvimento em java, relacionado a primeira liberação;</li> </ul>	Liberação L1 foi apresentada a equipe
Kleudson Rodrigues Wallace Luis do Carmo	25 de Janeiro de 2016	<ul style="list-style-type: none"> <li>• Mudança na forma de construir o projeto (projeto construído por etapas);</li> </ul>	Inicialmente o projeto estava sendo desenvolvido com uma visão fechada a solução do problema como um todo. A solução é construir o software por etapas, assim como as liberações são distribuídas

Kleudson Rodrigues	25 de Janeiro de 2016	<ul style="list-style-type: none"> <li>• Criação do repositório no GitHub;</li> </ul>	Gerenciamento de projeto
Kleudson Rodrigues Wallace Luis do Carmo	05 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• Comit de implementação;</li> <li>• Comit do primeiro documento de diagrama de classe</li> </ul>	
Wallace Luis do Carmo	06 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• Criação e atualização do Relatório de Atividades</li> </ul>	
Kleudson Rodrigues Wallace Luis do Carmo	07 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• <b>L1</b></li> <li>• <b>L2</b></li> </ul>	Conclusão das liberações 1 e 2.
Kleudson Rodrigues Wallace Luis do Carmo	22 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• <b>L3</b></li> </ul>	Jogo jogável com até 3 jogadores.
Kleudson Rodrigues Wallace Luis do Carmo	23 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• <b>L4</b></li> </ul>	Criação do pacote e das Classes de Teste
Kleudson Rodrigues Wallace Luis do Carmo	24 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• <b>L5</b></li> </ul>	Implementação do padrão Strategy
Kleudson Rodrigues Wallace Luis do Carmo	25 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• <b>L5</b></li> </ul>	Implementação do padrão Observer
Kleudson Rodrigues	25 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• Criação da documentação Javadoc</li> </ul>	
Kleudson Rodrigues Wallace Luis do Carmo	25 de Fevereiro de 2016	<ul style="list-style-type: none"> <li>• <b>L5</b></li> </ul>	Implementação do padrão Strategy

## Relatório de Construção

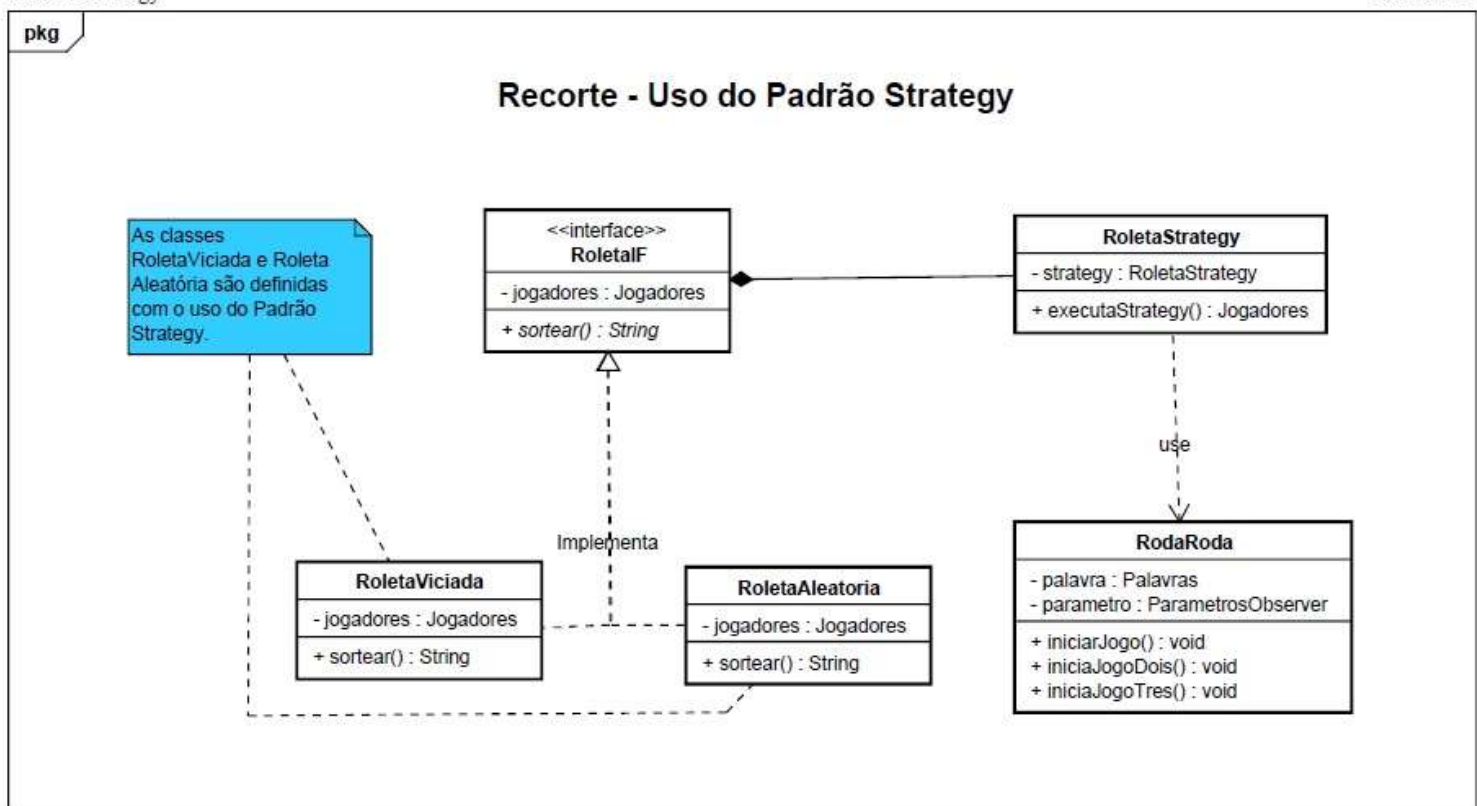
Aqui estarão descritas todas as estratégias utilizadas para a construção do software Roda a Roda, tendo como base o documento UML e o próprio código fonte.

O padrão de projeto Iterator está sendo utilizado para imprimir os dados da lista de jogadores. Este padrão também está sendo usado na classe Principal (Main) para auxiliar na criação dos Objetos Jogadores.

A roleta do jogo possui vinte possibilidades de jogadas, onde 4 delas são de azar e dezesseis de sorte, sendo assim os jogadores terão as mesmas oportunidades de jogadas. No entanto, o requisito aponta para a possibilidade de haver uma ou mais maneiras de giro, ou seja, a roleta poderá girar de maneiras diferentes. Para que o sistema possa trabalhar com vários tipos de giros, o Strategy é o padrão de projeto escolhido para lidar com tal situação, devido a facilidade pela qual permiti implementar vários tipos de giros, sem que uma interfira na outra, através do uso de uma interface. Dessa forma foi criado a interface chamada <<RoletaIF>> e duas formas de giros, sendo elas, **“RoletaViciada”** e **“RoletaAleatoria”** conforme ilustra o documento de apoio logo abaixo. Assim sendo, fica claro a possibilidade de criar novos tipos de roletas com diferentes tipos de giros, sem que uma interfira no funcionamento das outras.

Recorte Strategy

2016/03/03

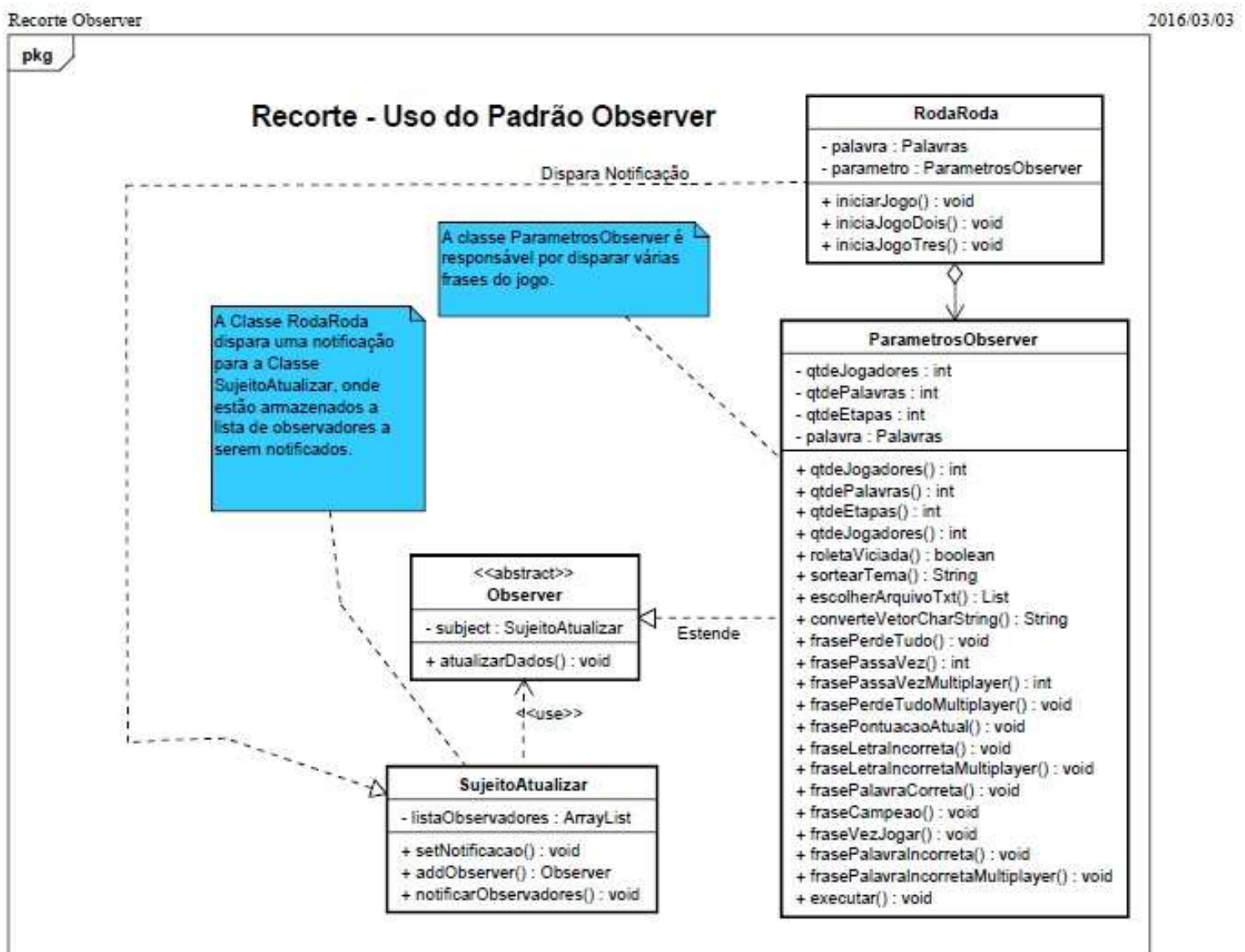


A roleta funciona da seguinte forma, assim que o jogo é iniciado é solicitado que o usuário insira a quantidade de jogadores, quantidade de palavras e

quantidade de rodadas. Em seguida, o programa irá perguntar com qual tipo de roleta o jogo deverá ser iniciado. Caso a 'Roleta Viciada' seja selecionada, a cada segunda jogada de cada jogador, a roleta sempre cairá no valor de azar "Passe a Vez", passando a vez para o próximo jogador. Caso a 'Roleta Aleatória' seja escolhida o jogo será jogado com as 20 posições randômicas para cada jogada.

As chamadas "ações do jogo", foram implementadas usando o padrão de projeto Observer, de modo que, a classe "SujeitoAtualizar" sempre estará responsável por disparar notificações (atualizações), informando o usuário com diversas frases de acordo com as ações do jogo.

Seguindo o padrão Observer, temos o seguinte diagrama.



A classe "SujeitoAtualizar" é responsável por disparar as notificações para as classes observadoras e também por adicionar os observadores na lista de observadores. Neste projeto a classe observadora é a classe ParametrosObserver

que estende a classe abstrata “Observer” que contém o método atualizarDados. Este método na classe ParametrosObserver dispara as mensagens do jogo de acordo com os eventos de cada classe.