Aurora

****

Sumário

[O objetivo 3](#_Toc515453247)

[A ideia 3](#_Toc515453248)

[O funcionamento 3](#_Toc515453249)

[Os testes 3](#_Toc515453250)

[Casos de testes 4](#_Toc515453251)

[Validação dos campos 4](#_Toc515453252)

[Teste 1 – Verificar se aceita campos vazios 4](#_Toc515453253)

[Teste 2 – Verificar se aceita apenas números 5](#_Toc515453254)

[Teste 3 – Verificar se aceita números inválidos 5](#_Toc515453255)

[Validação das Categorias 6](#_Toc515453256)

[Validação com usuário 6](#_Toc515453257)

# O objetivo

Desenvolver um software que auxilie o jogador de “Aurora” a selecionar a melhor opção de categoria, de maneira a receber a melhor pontuação.

# A ideia

Após a leitura de todas as especificações passadas no projeto, decidiu-se desenvolver em “Windows Form application”, pois é simples e fácil de organizar o código e desenvolver as telas.

Como se trata de um software para auxiliar no jogo, a única função realizada é mostrar a melhor opção de categoria, de forma a pontuar o máximo possível.

Este software não faz o calculo dos pontos que o jogador conquistou, nem realiza partidas, apenas mostra a melhor opção para pontuar.

# O funcionamento

Ao abrir o software, você encontra a tela inicial, não há nada na tela além do menuStrip, onde você pode selecionar o tipo de jogo. Você pode utilizar o “Jogo com dado”, onde você digita os valores dos dados físicos que jogou, e é informado da melhor opção de categoria que pode selecionar, ou “Jogo com máquina” onde não é necessário ter um dado físico, o jogo gera o valor dos dados de maneira aleatória, e informa a melhor opção de categoria.

Como muitas vezes algumas categorias pontuam o mesmo valor, existe uma ordem de prioridade, que segue: Aurora, Full House, Sequência maior, Sequência menor, Quadra, Trio, Dois Pares, Par, Seis, Cinco, Quatro, Três, Dois, Uns.

# Os testes

Durante a programação do software, sempre foi realizado depurações em cada parte do código para ter certeza de que tudo estava funcionando bem, o que auxiliou na procura por erros em diversos momentos.

Além da depuração parte por parte, também foram realizados diversos testes no software, que serão listados no tópico “Casos de Teste”.

# Casos de testes

## Validação dos campos

Mensagens de erro:

ValidarSeVazio = “O campo deve ser preenchido” – Verifica se o campo foi preenchido;

ValidarSeNumero = “O campo só aceita números” – Verifica se foi digitado valor numérico.

ValidarSeExisteDado = “Digite um valor válido” – Verifica se o valor é valido para um dado (1 a 6).

### Teste 1 – Verificar se aceita campos vazios

1. Abrir o software “Aurora”.
2. No menu Strip, selecionar a opção “Jogo/ Novo/ Jogo com dado”
3. Deixar as caixas de texto vazias, e apertar play (Erro: ValidarSeVazio).
4. Deixar uma das caixas vazias, e as outras preenchidas com números de 1 a 6 (Erro: ValidarSeVazio).
5. Deixar duas das caixas vazias, e as outras preenchidas com números de 1 a 6 (Erro: ValidarSeVazio).
6. Deixar três das caixas vazias, e as outras preenchidas com números de 1 a 6 (Erro: ValidarSeVazio).
7. Deixar quatro das caixas vazias, e as outras preenchidas com números de 1 a 6 (Erro: ValidarSeVazio).

### Teste 2 – Verificar se aceita apenas números

1. Abrir o software “Aurora”.
2. No menu Strip, selecionar a opção “Jogo/ Novo/ Jogo com dado”
3. Preencher todas as caixas com letras (Erro: ValidarSeNumero)
4. Preencher uma das caixas com letras, e as outras com números de 1 a 6 (Erro: ValidarSeNumero).
5. Preencher duas das caixas com letras, e as outras com números de 1 a 6 (Erro: ValidarSeNumero).
6. Preencher três das caixas com letras, e as outras com números de 1 a 6 (Erro: ValidarSeNumero).
7. Preencher quatro das caixas com letras, e as outras com números de 1 a 6 (Erro: ValidarSeNumero).

### Teste 3 – Verificar se aceita números inválidos

1. Abrir o software “Aurora”.
2. No menu Strip, selecionar a opção “Jogo/ Novo/ Jogo com dado”
3. Preencher todas as caixas com números fora do intervalo (Erro: ValidarSeExisteDado).
4. Preencher uma das caixas com números fora do intervalo (Erro: ValidarSeExisteDado).
5. Preencher duas das caixas com números fora do intervalo (Erro: ValidarSeExisteDado).
6. Preencher três das caixas com números fora do intervalo (Erro: ValidarSeExisteDado).
7. Preencher quatro das caixas com números fora do intervalo (Erro: ValidarSeExisteDado).
8. Preencher cinco das caixas com números fora do intervalo (Erro: ValidarSeExisteDado).

## Validação das Categorias

Para verificar se todas as categorias foram utilizadas e escolhidas corretamente pelo programa, utilizou-se o “Jogo com máquina” pra uma maior abrangência de números aleatórios.

## Validação com usuário

Após desenvolvido e testado, o software foi apresentado a um usuário para utilização e validação final do produto.