

# INF1301 – Relatório de Testes Unitários

## Jogo – Mastermind

Bruno Abtibol Ramos

Daniel Stulberg Huf

João Pedro Khair Cunha

CT001 – Validar escolha de nível	
Objetivos do Teste:	Verificar se as configurações de nível são obtidas corretamente.
Pré-condições:	1. Nenhuma. Primeira ação do jogo.
Parâmetros de entrada:	1. Nível do jogo (escolha) - escolhido pelo jogador
Ações:	1. Verificar se o nível é válido: 1, 2 ou 3 (ver observação)
Retorno esperado:	1. Tupla contendo: <ul style="list-style-type: none"><li>a. Número de pedras da senha</li><li>b. Lista de cores disponíveis</li><li>c. Número máximo de tentativas</li></ul>
Observações:	A função funcionou conforme o esperado. A escolha do nível pelo jogador (por meio do comando input) e a verificação da validade do nível (por meio do comando while) serão substituídas por uma caixa de opções quando a parte gráfica do jogo for desenvolvida. Essa caixa de opções só oferecerá opções válidas para o jogador escolher.

CT002 – Validar definição de senha	
Objetivos do Teste:	Verificar se a função define_senha funciona corretamente.
Pré-condições:	<ol style="list-style-type: none"> <li>1. O jogador já selecionou o nível do jogo.</li> <li>2. A função escolhe_nivel() retornou corretamente os parâmetros de dificuldade do jogo.</li> <li>3. O pacote random foi corretamente importado.</li> </ol>
Parâmetros de entrada:	<ol style="list-style-type: none"> <li>1. Lista de cores disponíveis (cores)</li> <li>2. Número de pedras da senha (nPedras)</li> </ol>
Ações:	<ol style="list-style-type: none"> <li>1. Selecionar aleatoriamente, por meio do comando random.choice(), os elementos da lista cores que comporão a senha</li> </ol>
Retorno esperado:	Senha - lista com (nPedras) elementos. Cada elemento é uma cor da lista cores.
Observações:	A função retornou corretamente a lista senha.

CT003 – Validar inicialização do jogo	
Objetivos do Teste:	Verificar se a função novoJogo inicializa corretamente as variáveis e estruturas utilizadas nas regras do jogo.
Pré-condições:	<ol style="list-style-type: none"> <li>1. A função escolhe_nivel() funcionou corretamente.</li> <li>2. A função define_senha() funcionou corretamente.</li> </ol>
Parâmetros de entrada:	Nenhum.
Ações:	<ol style="list-style-type: none"> <li>1. Atribuir o valor 0 à variável rodada</li> <li>2. Inicializar as variáveis nPedras, cores e nJogadas por meio da função escolhe_nivel()</li> <li>3. Gerar uma senha aleatória por meio da função define_senha()</li> <li>4. Inicializar tentativa como uma lista de mesmo tamanho da senha, e cada elemento é uma string vazia.</li> <li>5. Inicializar compara como uma lista de 3 números (valendo, inicialmente, 0).</li> </ol>
Retorno esperado:	Nenhum.
Observações:	A função inicializou corretamente todas as variáveis.

CT004 – Validar comparação da tentativa e senha	
Objetivos do Teste:	Verificar se a função compara_senha compara corretamente a senha com a sequência de cores determinada pelo jogador
Pré-condições:	<ol style="list-style-type: none"> <li>1. O jogo foi iniciado (novoJogo())</li> <li>2. O jogador definiu a tentativa de senha na rodada.</li> </ol>
Parâmetros de entrada:	<ol style="list-style-type: none"> <li>1. Senha aleatoriamente determinada (senha)</li> <li>2. Sequência de cores definida pelo jogador (tentativa)</li> <li>3. Número de pedras (nPedras)</li> <li>3. Lista com o número de peças pretas, brancas e vazios (compara)</li> </ol>
Ações:	<ol style="list-style-type: none"> <li>1. Criar listas auxiliares de senha e tentativa por meio do comando .copy()</li> <li>2. Verificar a quantidade de peças pretas, ou seja, o número de peças da tentativa de mesma cor e posição da senha.</li> <li>3. Verificar a quantidade de peças brancas, ou seja, o número de peças da tentativa de mesma cor e posição diferente da senha.</li> <li>4. Determinar a quantidade de espaços vazios na resposta, ou seja, o número de peças cuja cor e posição estão incorretas.</li> <li>5.</li> </ol>
Retorno esperado:	<ol style="list-style-type: none"> <li>1. Lista compara, com 3 elementos inteiros: <ol style="list-style-type: none"> <li>a. Número de peças pretas</li> <li>b. Número de peças brancas</li> <li>c. Número de vazios</li> </ol> </li> </ol>
Observações:	Essa foi a função mais complexa da 1ª iteração, gerando alguns erros de comparação até obtermos a versão final. Esta foi capaz de comparar corretamente todos os exemplos testados, incluindo senhas e tentativas com uma ou mais repetições de cores em sua sequência.