

# GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO PROGRAMAÇÃO DE COMPUTADORES I

Trabalho Final Anual – TFA



## Introdução

O TFA tem por objetivo avaliar todo o conhecimento adquirido pelo aluno ao longo das disciplinas de Programação de Computadores (Linguagem de Programação) através do desenvolvimento de um game que contemple todos os conceitos apresentados durante o ano. O aluno deverá (terá a possibilidade) escolher uma proposta (jogo), e desenvolvê-la seguindo todas as especificações definidas pelo o professor e em conjunto com o ele, utilizando para tal, obrigatoriamente, a linguagem de programação C.

O jogo escolhido deve ser especificado e apresentado ao professor (pelo aluno) até o final da aula. Sendo a proposta aprovada, o aluno poderá então começar sua codificação.

## Funcionalidades Genéricas (Obrigatórias)

Todas as propostas devem possuir como padrão as seguintes funcionalidades:

- 1) *Cadastro do administrador do sistema*: a primeira vez que o programa for executado ele deve verificar se já existe um administrador cadastrado, se sim: o programa deve ser redirecionado para uma tela de autenticação, caso contrário ele deve ir para uma tela de cadastro de administrador que solicitará os seguintes dados: **nome**, **login** e **senha**; **(0.5 ponto)**
- 2) *Autenticação do usuário*: havendo um administrador cadastrado no sistema, o programa deverá apresentar uma tela de autenticação, que solicitará os dados **login** e **senha**. Após os dados terem sido digitados o aplicativo efetuará uma busca sobre as informações recebidas dentro do arquivo binário de cadastro de usuários, verificando se existe um usuário que possui **login** e **senha** iguais aos digitados. Caso não seja encontrado, uma mensagem de erro deve ser apresentada e tela de autenticação deve ser reapresentada, caso contrário, ou seja, seja constatado que existe um usuário cadastrado com os dados especificados o programa deve verificar se este usuário trata-se do administrador do aplicativo ou de um jogador, redirecionando o programa para respectiva tela. **(0.5 ponto)**
- 3) *Tela (Menu) do Administrador*: ao autenticar-se, o administrador receberá uma tela de boas-vindas com o texto: “Seja bem-vindo: Nome do Administrador”, e também com um menu apresentando as seguintes opções: **(0.5 ponto)**

- a. Cadastrar Jogador: permite ao administrador cadastrar novos jogadores, ela solicitará os dados: **nome**, **login** e **senha**, e os armazenará no mesmo arquivo binário onde foram armazenados os dados do administrador;
- b. Visualizar Jogadores: permite ao administrador visualizar as informações (nome, **login** e senha) de todos os usuários já cadastrados no programa, incluindo as suas informações, que devem ser apresentadas por primeiro indicando que se trata do administrador do sistema.
- c. Remover Jogador: permite ao administrador remover um jogador do programa, apagando do arquivo binário o registro do mesmo, o que impossibilitará sua autenticação futuramente.
- d. Efetuar “Logoff”: permite ao administrador “deslogar” do programa, após isso deve ser apresentada novamente a tela de autenticação do usuário.
- e. Sair: permite ao administrador sair do programa.

4) *Tela (Menu) do Jogador*: ao autenticar-se, o jogador receberá uma tela de boas-vindas com o texto: “Seja bem-vindo: Nome do Jogador”, e também com um menu apresentando as seguintes opções: **(1.0 ponto)**

- a. Iniciar o Jogo: permite ao jogador iniciar um novo jogo, de acordo com a proposta escolhida pelo aluno.
- b. Continuar partida: permite ao jogador reiniciar um jogo pausado **por ele** anteriormente.
- c. Modificar Dados de Autenticação: permite ao jogador alterar suas informações, **login** e **senha**, cadastradas e armazenadas anteriormente no arquivo binário de usuários.
- d. Configuração: permite ao jogador configurar algumas opções de jogo, de acordo com a proposta escolhida pelo aluno.
- e. Ranking: permite ao jogador ver o *ranking*, em ordem decrescente, dos 5 (cinco) melhores jogadores de acordo com as suas pontuações. Devem ser apresentados os dados: **posição** (no ranking), **nome** e **pontuação** de cada um dos cinco jogadores;
- f. Efetuar “Logoff”: permite ao jogador “deslogar” do programa, após isso deve ser apresentada novamente a tela de autenticação do usuário.
- g. Sair: permite ao jogador sair do programa.

5) *Ranking*: todos os aplicativos devem salvar, ao final de cada partida, os números finais do jogador, de acordo com a proposta escolhida pelo aluno, armazenando-os em um arquivo binário. O arquivo binário deve conter o nome do jogador e a sua pontuação. **(1.5 ponto)**

6) *Salvar/Sair do Jogo*: será permitido ao jogador salvar seu jogo a qualquer momento, podendo reiniciá-lo posteriormente, essa funcionalidade deve possibilitar ao jogador recomeçar o jogo salvo em um momento futuro, com as informações exatamente iguais aquelas apresentadas na hora em que a partida



|                             |  |
|-----------------------------|--|
| <i>Alterar Usuário:</i>     | <b>int</b> alterarUsuario( <b>char*</b> nome, <b>char*</b> login,<br><b>char*</b> senha);<br><i>0 – usuário não encontrado para esse nome</i><br><i>1 – dado do usuário alterado com sucesso</i> |
| <i>Visualizar Usuários:</i> | <b>void</b> verRanking();  |
| <i>Visualizar Ranking:</i>  | <b>void</b> verRanking();  |
| <i>Salvar Ranking:</i>      | <b>void</b> salvarRanking(ranking r);  |
| <i>Salvar Jogo:</i>         | <b>void</b> salvarGame(game g);  |

**Observações:** o uso de variáveis globais não é permitido inicialmente, podendo ser autorizado caso o aluno apresente, *durante a codificação do seu projeto*, uma justificativa aceita pelo professor.

### Funcionalidades Específicas

O aluno deve definir (e o professor aprovar) quais serão as funcionalidades existentes dentro jogo por ele proposto, definindo a pontuação que cada uma delas possuirá, lembrando que as funcionalidades específicas contribuem com um total de **3.0 pontos** para nota final do trabalho.

Os trabalhos (jogos) efetuados ao longo da disciplina não podem ser utilizados (escolhidos) para o TFA, assim como também não podem existir equipes com um mesmo tema, portanto quem apresentar sua proposta primeiro possuirá prioridade de escolha.

### Avaliação / Prazo de Entrega

*Integrantes da equipe:* o trabalho pode ser feito de forma individual ou em dupla, sendo a nota única para todos os componentes da equipe. O número de integrantes será diretamente proporcional a dificuldade da proposta apresentada pelos alunos, sendo definido pelo professor.

*Data de entrega:* 09/02/2015

*Recuperação Final:* 16/02/2015 (apresentação do trabalho valendo **8.0 ou prova**)

### *Critérios de avaliação:*

- Cumprimento de todas as funcionalidades especificadas;
- Conhecimento total sobre a codificação apresentada;
- Respostas corretas sobre os questionamentos feitos pelo professor;
- Uso dos recursos em C: funções, *structs*, arquivos; (como especificado)
- Originalidade;
- Criatividade;

### **Informações Finais**

Quaisquer outras dúvidas, necessidade de modificação ou inserção de funcionalidades não previstas, alteração dos pesos definidos para cada item, entre outras coisas, serão discutidas e efetuadas em sala de aula em comum acordo com todos os alunos presentes.