

Social Game

Request for Proposals

1 Apresentação da empresa

A Graphs4Social, S.A. é uma empresa com sede no Porto (Portugal) criada em Novembro de 2012 cuja missão é fornecer aplicações de manipulação e visualização de grafos de redes sociais. A empresa decidiu recentemente expandir o seu portfolio de produtos entrando na área de jogos, mas mantendo o foco nos grafos de redes sociais. A empresa decidiu recorrer à subcontratação de serviços de desenvolvimento uma vez que não possui capacidade livre de momento.

2 Objeto pretendido

Pretende-se o desenvolvimento de um **protótipo** para um jogo baseado na visualização e manipulação de grafos sociais.

O protótipo inicial deve ser constituído pelos seguintes módulos:

- Jogo 3D com componentes de inteligência artificial
- Site web

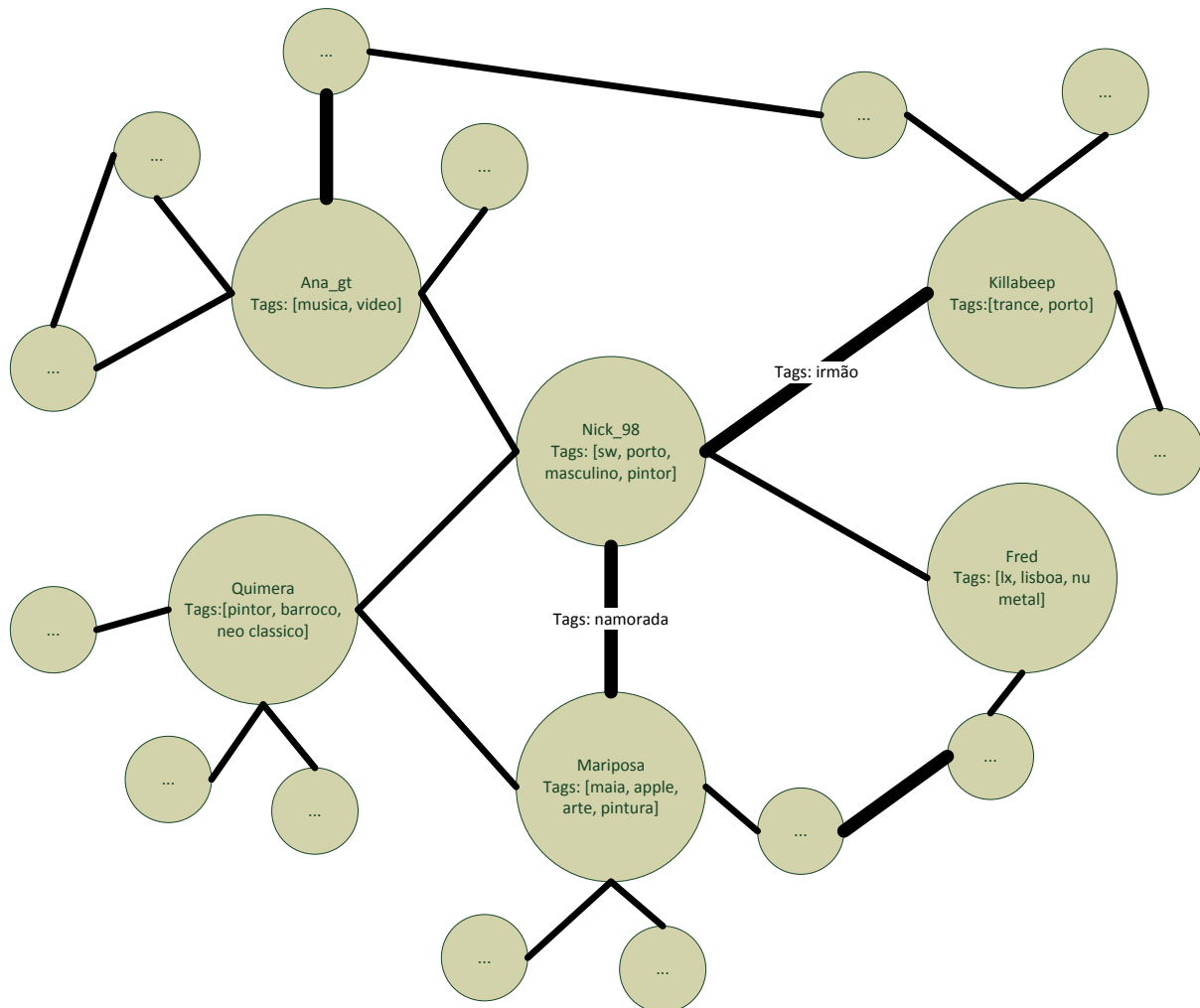
Tratando-se de um sistema protótipo, é aceitável que apenas algumas algumas funcionalidades estejam implementadas, devendo constar no relatório de proposta quais as funcionalidades implementadas.

O Jogo simula uma rede social e o jogador tem por objetivo expandir a sua rede social, com o objetivo último de ter a maior e mais forte rede social possível. O jogo desenrola-se numa série de missões em que o jogador terá que avançar e aumentar a sua rede em competição com outros jogadores.

As missões consistem em tornar-se amigo de outro personagem separado por n grau de distância, em que n será tanto maior quanto maior a dificuldade do nível. Nos níveis mais fáceis, a escolha do utilizador deverá ser sugerida tendo por base as *tags* e conexões partilhadas. Ao longo do jogo, para avançar na sua missão o jogador poderá ter que defrontar mini-jogos (ex., enforcado, 4 em linha, torres de hanoi) por forma a ganhar a confiança de outro personagem.

No modo normal, a rede social do jogador é totalmente simulada pelo sistema. No modo avançado, a rede social utilizada pelos jogadores é baseada na verdadeira rede social do utilizador. Cada utilizador possui ligações com outros utilizadores, podendo catalogar essas ligações de acordo com etiquetas (*tags*) e “força de ligação”. Cada utilizador possui um perfil ao qual pode associar um conjunto de *tags* relativas a interesses seus, entre outra informação (ex., nome, data de nascimento, nº telefone, email, perfil LinkedIn, perfil Facebook). Cada utilizador possui também no seu perfil informação sobre o seu “estado de humor”, ex., alegre, triste, chateado, contente, etc.

Embora o sistema conheça toda a rede social, cada utilizador vê a rede a partir da sua perspectiva, ou seja, ele será o nó “central” do seu grafo. A rede social de um dado utilizador tem um “raio” máximo de 2: amigos dos amigos.



Em qualquer momento deverá ser possível ver uma representação 3D da rede até dois graus de separação a partir do nó atual, mas deverá ser possível navegar na totalidade da rede (de forma eficiente em termos de consumo de memória). A força das ligações deve ser evidenciada de forma visual (utilizadores fora da sua rede social deverão aparecer de forma visual diferenciada)

Para criar uma nova ligação, um jogador deverá “deslocar-se” (navegar pelo grafo seguindo as ligações) para um nó correspondente a alguém da sua rede social e que seja parte da rede social de quem se quer ser “amigo” e pedir uma “introdução” a esse amigo. No modo normal, o sistema terá uma função aleatória para decidir se faz a introdução ou não entre os dois jogadores. No modo avançado, a introdução terá que ser aprovada pelo utilizador real.

O sistema deve ter “leader board” com o Top 10 e a posição do utilizador atual para a seguinte informação:

- Dimensão da rede
- Fortaleza da rede
- *tag*

O sistema deverá dar dicas ao utilizador, nomeadamente,

- “Iluminar” o caminho mais forte e mais curto para determinado utilizador,
- Mostrar o tamanho da rede de um utilizador (até 3º nível – dois graus de separação)
- Mostrar o grafo de amigos comuns entre dois utilizadores

3 Sistema a desenvolver¹

3.1 Considerações gerais

Todas as aplicações devem ter em consideração as perspetivas de internacionalização da empresa e por isso devem suportar localização do software ou deve ser indicado uma estratégia para a sua localização posterior.

Adicionalmente para manter concordância com as práticas de desenvolvimento da empresa, todas as aplicações devem possuir uma organização em camadas separando os componentes de apresentação (interface pessoa-máquina) dos componentes de processamento e acesso a dados recorrendo às boas práticas referenciadas nos padrões de aplicações empresariais.

De notar que o objeto pretendido com este concurso é um **sistema integrado**, pelo que um dos critérios mais importantes de avaliação das propostas será a coerência dessa integração entre os vários módulos. Nesse sentido, cada proponente deve na descrição da arquitetura da sua proposta evidenciar como é que os vários módulos do sistema se encontram integrados (ex., interfaces e dependências entre módulos), como é que essa integração é realizada (ex., web services, base de dados, biblioteca de funções) e como é garantida a coerência da informação partilhada entre os vários módulos, por exemplo, entre o grafo utilizado pelo módulo de IA e o grafo utilizado pelo módulo visualizador.

3.2 Módulo de navegação e visualização 3D

Este módulo deverá permitir a visualização em 3D do grafo da rede social do utilizador. Deverá também apresentar, em sobreposição, um mini mapa de todo o grafo da rede social. O módulo deverá suportar áudio para os efeitos especiais (ex., chuva, trovões) e eventual “música de fundo”.

A colocação dos nós no espaço deverá obedecer a um algoritmo simples de distribuição radial dos nós tendo por base o nó atual em que o utilizador se encontra. A cota de cada nó deverá ser calculada com base no número de ligações de cada utilizador, ou seja, quanto maior for esse número, mais elevado se encontrará este. A dimensão de cada nó dependerá do número de *tags*. A

¹ Os requisitos claramente marcados como valorização apenas serão considerados caso as funcionalidades base (obrigatórias) estejam desenvolvidas.

força de ligação entre duas pessoas deverá traduzir-se na espessura do elemento de ligação que liga os respetivos nós.

O módulo visualizador deve mostrar de forma visual diferenciada o caminho de ligação entre o jogador/utilizador que está autenticado e um outro utilizador (escolhido com o rato ou através de uma pesquisa). Deverá ser possível controlar livremente a câmara para visualizar o grafo – pretende-se um efeito de navegação no espaço em que o grafo representa uma estrutura sólida e em que a câmara pode movimentar-se livremente à sua volta. Deve considerar os nós e as ligações como obstáculos através dos quais não é possível passar. É importante dar destaque aos nós e às ligações para onde “se está a olhar”, devendo configurar-se as fontes de iluminação mais adequadas para obter esse efeito final.

Quando for necessário ultrapassar um mini jogo, esse deve ser apresentado numa “janela sobreposta” ao restante jogo.

Será **valorizado** se ao passar o cursor por um nó for apresentada uma *tip* flutuante com informação acerca da pessoa associada a esse nó, bem como o avatar desse utilizador. Para tal poderão recorrer a modelos importados tais como, por exemplo, OBJ, 3DS ou MDL.

Deverá existir um modo de visualização em que o nó de cada utilizador tem uma indicação visual do “estado de humor” desse utilizador, por exemplo, chuva para os utilizadores com estado de humor “triste”. Pretende-se recorrer a *billboards* com a forma de uma nuvem negra, ou de um relâmpago, ou de um sol, ou de um coração vermelho, consoante o estado de espírito da pessoa associada a esse nó. Será **valorizado** a criação de efeitos atmosféricos localizados tais como nevoeiro e chuva ou fogo (utilizando motores de partículas) que reforcem o referido estado. Também será **valorizada** a inclusão de uma *skybox*, *skydome* ou *skyplane*.

Será **valorizada** a exportação para um ficheiro num formato vetorial (ex., WMF) ou *raster* (ex., jpeg).

3.3 Módulo de inteligência artificial

O sistema deve possuir um módulo de inteligência artificial desenvolvido em Prolog responsável pela lógica dos mini jogos e por alguma informação geral sobre o jogo. Este módulo deve estar integrado com o resto do projeto comunicando com os outros módulos sempre que necessário, podendo ser implementado como um componente local da aplicação ou como um serviço web.

Além da lógica de cada mini jogo, será neste módulo que devem ser implementadas as seguintes funcionalidades:

- Determinar o tamanho da rede de um utilizador (até 3º nível)
- Obter os amigos que tenham em comum X *tags* sendo X parametrizável. Deve ter em atenção que duas *tags* sintaticamente diferentes podem ter o mesmo significado semântico (e.g. C# e CSharp).
- Sugerir conexões com outros utilizadores tendo por base as *tags* e conexões partilhadas (até 3º nível)
- Determinar o caminho mais forte (segundo um peso dado à ligação que reflete a natureza da relação - família, amigo, conhecido, etc.) e o mais curto entre dois utilizadores.

3.4 Módulo Site

Este módulo deve consistir numa aplicação web com uma interação pessoa-máquina enriquecida (por ex., através da utilização de AJAX, Themes, Skin, Web parts, Navigation e Profiles).

A principal função deste módulo será o registo e utilizadores, permitir ao utilizador construir a sua rede social associando-se a outros utilizadores (indicando *tags* e força dessa relação) e permitir ao utilizador fazer *download* do jogo.

O acesso a este módulo está disponível para utilizadores registados e cada utilizador tem apenas acesso aos seus dados e da sua rede social. Todos os utilizadores (estejam ou não autenticados) podem consultar a informação de carácter genérico tal como:

- Leader board
- Dimensão da rede (número de utilizadores),
- *Tag cloud* das *tags* de **todos** os utilizadores
- *Tag cloud* das *tags* de **todas** as relações

Para os utilizadores autenticados será ainda possível consultar:

- Sugestões de amigos
- Tamanho da sua rede social (até ao 3º nível)
- *Tag cloud* das **suas** *tags* de utilizador
- *Tag cloud* das **suas** *tags* das relações

Não sendo requisito do sistema protótipo, pretende-se que **no futuro** através do site também se possa ter acesso às funcionalidades de pesquisa no grafo. Também será considerado **valorização** importar ligações a partir de uma conta de Facebook, LinkedIn ou outra rede social.

Uma outra **valorização** deste módulo é a existência de um módulo de visualização do grafo do utilizador usando WebGL ou um toolkit JavaScript².

4 Infraestrutura computacional

As máquinas de instalação e demonstração do sistema devem ser máquinas de produção e não máquinas de desenvolvimento; isto é, **não devem ter instaladas as ferramentas de desenvolvimento nem devem executar as aplicações dentro dos ambientes de desenvolvimento**. Devem configurar devidamente as máquinas virtuais da infraestrutura do DEI³ para o efeito.

² Ver por exemplo, “Interactive Data Visualization for the Web” de Scott Murray, (2013) O’Reilly. Disponível em <http://chimera.labs.oreilly.com/books/1230000000345/index.html>

³ Devem ser reutilizadas as máquinas virtuais Linux que foram preparadas para o efeito durante as aulas PL de ASIST (uvmXXX.dei.isep.ipp.pt).

O servidor Windows que aloja o “site” do projeto deve ser monitorizado em diversos aspetos, para o efeito pretende-se que numa máquina virtual disponibilizada para o efeito (servidor Linux), seja configurado o serviço NAGIOS.

Estão previstas quatro formas de monitorização, devendo ser implementadas as mais adequadas em cada caso:

- *Direta* - os serviços de rede disponibilizados pelo servidor podem ser diretamente verificados usando os “plugins” apropriados
- *NSClient++* - o serviço pode ser instalado e configurado no servidor Windows. Permite ao NAGIOS através dos “plugins” “check_nrpe” ou “check_nt” efetuar diversas verificações do estado interno do servidor.
- *SNMP* – o agente SNMP pode ser ativado e configurado no servidor Windows permitindo então que o protocolo de gestão seja usado pelos “plugins” apropriados no acesso à MIB.
- *Via HTTP (Web Service)* – o site pode estar preparado para disponibilizar informações de estado em determinadas páginas, no NAGIOS será necessário criar um “plugin” ou configurar um já existente de modo a obter essas informações do “site”.

Devem monitorizar os seguintes elementos:

1. *Acesso http* - verificar que a página principal do site está acessível através do protocolo HTTP
2. *“Webservice”* - Verificar o funcionamento do “webservice” que considere mais importante no contexto do projeto.
3. *Espaço livre em disco* - Se o espaço livre for inferior a 10% do total deve ser emitido um alerta do tipo WARNING, se for inferior a 5% deve ser emitido um alerta do tipo CRITICAL.
4. *Taxa de utilização dos CPUs* - Se a taxa de utilização for superior a 95% deve ser emitido um alerta do tipo WARNING, se for superior a 97% deve ser emitido um alerta do tipo CRITICAL.
5. *Tráfego na interface de rede* - se o tráfego médio for superior a 10% da capacidade deve ser emitido um alerta do tipo WARNING, se for superior a 30% deve ser emitido um alerta do tipo CRITICAL.
6. *Número de utilizadores registados* - Sempre que ocorrer uma alteração no número de utilizadores registados, deve ser emitido um alerta do tipo WARNING.
7. *Taxa de “downloads”* - Sempre que o número de “download” da aplicação no espaço de uma hora ultrapassar o valor 5, deve ser emitido um alerta do tipo WARNING. Se ultrapassar o valor 100 deve ser emitido um alerta do tipo CRITICAL.

Na demonstração do protótipo é obrigatório recorrer aos servidores virtuais criados para o efeito. Será **valorizado** ativar o serviço HTTPS no servidor Windows e monitorizar o mesmo através do NAGIOS.

5 Método de trabalho

Por uma questão de alinhamento com o método de trabalho da Graphs4Social deverão ser efetuadas entregas parciais ao longo do decorrer do projeto de acordo com o seguinte cronograma. Adicionalmente devem ser utilizadas boas práticas de controlo de versões e testes.

5.1 Controlo de versões

Para seguir as práticas definidas na empresa os concorrentes devem utilizar um sistema de controlo de versões baseado em **Git** em todo o seu desenvolvimento do protótipo. Cada empresa pode utilizar o sistema de *hosting* que entender (ex., bitbucket, github) e **terá que garantir acesso de leitura a dois utilizadores adicionais** indicados pela empresa durante todo o processo de desenvolvimento.

Apenas serão aceites propostas que evidenciem o uso de boas práticas de edição colaborativa de código entre os elementos da equipa, devendo ser apresentado o grafo/log de revisões de cada subprojeto.

Todas as tarefas da equipa devem ser devidamente planeadas e registadas numa ferramenta de *issue tracking*. A proposta será avaliada pela qualidade do processo de trabalho evidenciada nos registos da ferramenta de controlo de versões e *issues*.

De acordo com as práticas da empresa, **todos** os artefactos devem ser guardados no repositório de controlo de versões, nomeadamente (mas não limitado a): diagramas, documentos, código fonte, ficheiros de configuração, scripts de compilação, scripts de configuração do Nagios, imagens e outros recursos vídeo/áudio. Relativamente à documentação e diagramas, estes podem ser guardados na Wiki da ferramenta de controlo de versões se a mesma suportar controlo de versões da Wiki, não sendo assim necessário guardar esses artefactos no repositório.

5.2 Equipas multifuncionais

Um dos princípios fundamentais da Graphs4Social é o de equipas multifuncionais, pelo que, embora seja aceitável alguma especialização por parte dos elementos da equipa, nenhum elemento deve desempenhar apenas uma função, ex., programador web. Com este princípio a Graphs4Social reduz o risco inerente à falta (temporária ou permanente) de um dado elemento.

Cada proponente deve apresentar um quadro em que indique a divisão de tarefas efetuadas entre os elementos da equipa e que evidencie a aplicação deste princípio. Exemplo (cada equipa deve adequar o quadro à arquitetura/divisão de componentes da sua proposta de solução em particular):

		Elementos do grupo				
Módulo	Feature	António	Joana	Martim	Francisca	Sara
Visualizador	Modelo	x	X			X
	Minimapa		X			X
	Navegação câmara	X		X	X	
	Ligação a IA				X	
IA	BC grafo	X		X	x	



	Sugestão de amigos			x	X	
	maven	X	X			
	Cálculo de caminho		X			X
	Grau médio de separação					X
	Outros cálculos	x			X	
...

É assumido que a documentação é efetuada por todos os membros da equipa, nomeadamente segue-se a regra “quem fez, documenta”. É assumido que os testes são efetuados por todos os elementos da equipa, nomeadamente segue-se a regra “quem testa não é quem fez”. Essa evidência deve ser apoiada com os grafos/logs de revisões de cada subprojecto, relatórios de *pair programming* (se aplicável) ou outros meios. Casos detetados de elementos de equipa que apenas desempenharam uma função serão **excluídos** do concurso⁴.

5.3 Testes

Todos os módulos devem ter a descrição de um plano de testes descrevendo os vários testes a efetuar, por exemplo:

Módulo:	Inteligência artificial	
Objetivo:	Testar cálculo do número de ligações de um utilizador	
Método de teste:	Manual	
Cenário	Teste	Resultado esperado
Base de conhecimento	nr_lig('nick98', N)	1

⁴ Em termos de avaliação de LAPR5 corresponde a ter atribuído SM na pauta de avaliação. Tenham especial cuidado para não cometer o erro de alguém apenas fazer a componente de infraestrutura ou de documentação do projeto não participando **ativamente** nas restantes atividades. Notem que no caso de disciplinas que usem o projeto de LAPR5 para atribuir a sua nota de frequência é da responsabilidade do regente dessa disciplina decidir como proceder nestas situações, sendo aceitável que no âmbito dessa disciplina os alunos que não trabalharam nesse componente não tenham nota de frequência.

"um_utilizador_soltario.pl"	nr_lig('ana_gt', N)	<i>Erro: Utilizador inexistente</i>
Base de conhecimento "turma_TP2.pl"	nr_lig('nick98', N)	20

Sempre que adequado devem ser desenvolvidos testes unitários sobre o código. Devem ser preparados "bases de dados" e cenários de teste que permitam verificar e reproduzir os testes indicando os resultados esperados.

6 Processo de avaliação de propostas

6.1 Esclarecimento de dúvidas funcionais com o cliente

A Graphs4Social organizará 1 reunião de arranque e esclarecimento de dúvidas sobre o presente RFP no dia 9 de Dezembro das 18h00 às 20h00. Nessa reunião devem estar presentes pelo menos dois representantes de cada concorrente.

No restante período as dúvidas funcionais serão esclarecidas através do fórum eletrónico a criar para o efeito, ao qual terão acesso todos os concorrentes ao RFP. O cliente responderá às dúvidas levantadas no prazo médio de 24 horas. Cada concorrente deverá primeiro verificar se a sua dúvida já se encontra respondida no fórum antes de colocar nova entrada.

6.2 Demos intermédias

A decorrer na semana de 16 a 20 de dezembro de 2013 e 6 a 10 de janeiro de 2014 no local de trabalho de cada equipa concorrente⁵. Cada proponente fará uma pequena demonstração informal (aproximadamente 10 minutos) do estado do projeto a um júri segundo atribuição a definir posteriormente.

Devem apresentar:

- Arquitetura do sistema (principais módulos e suas interligações)
- Principais estruturas de dados utilizadas, ex., grafo do módulo visualização; factos do módulo IA
- Modelo de dados (se aplicável)
- Divisão de tarefas entre membros da equipa
- Repositório de controlo de versões
- Servidores virtuais utilizados
- Estado funcional do sistema (focando na **integração** entre os vários módulos, preferencialmente instalados e a executar dos servidores virtuais)

⁵ Esta apresentação informal decorrerá nas aulas OT/PL e será avaliada por um docente a atribuir posteriormente.



6.3 Demo final

A decorrer de **15 a 17 de janeiro de 2014**. Cada grupo fará uma demonstração formal do projeto a um painel de avaliação segundo calendário a apresentar posteriormente. Nessa apresentação cada proponente terá **20 minutos** (máximo) para descrever e demonstrar a sua proposta, seguido de um período de até 40 minutos para esclarecimentos ao júri.

Devem apresentar:

- Documentação
 - Arquitetura do sistema (principais módulos e suas interligações)
 - Principais estruturas de dados utilizadas
 - Modelo de dados (se aplicável)
 - Plano de testes
 - Divisão de tarefas entre membros da equipa e breve descrição do processo de desenvolvimento utilizado pela equipa
 - Grafo de revisões de cada subprojeto
- Servidores virtuais utilizados e respetiva monitorização de servidores e serviços
- Software (instalado e a executar dos servidores virtuais)
 - Funcionalidades implementadas
 - Integração funcional **coerente** entre todas as aplicações
- Formulário de avaliação da equipa.

6.4 Critérios de avaliação de propostas

As propostas serão avaliadas por um júri e classificadas segundo os seguintes critérios:

- Qualidade da apresentação e demonstração (ex., preparação, tempo limite, dados de demonstração)
- Apreciação global da solução
- Coerência de integração entre módulos
- Grau de realização dos objetivos específicos de cada módulo
- Qualidade da apresentação da proposta
- Boas práticas evidenciadas no método de trabalho

Os requisitos claramente marcados como valorização apenas serão considerados caso as funcionalidades base (obrigatórias) estejam desenvolvidas.