

# SempreLigados

## relatorio

### Estrutura de dados

Andre Pimenta Ribeiro 54745,

Daniel Miranda Santos 54825,

Joao Miguel Gomes 54738

Março de 2009/10

#### **Resumo**

Neste relatório pretendemos explicar as nossas estruturas de dados e qual a razão da sua escolha. Para as diferentes tarefas que são exigidas optamos por 3 tipos de estruturas de dados, com funcionalidades diferentes.

# Conteúdo

<b>1</b>	<b>Estrutura de dados</b>	<b>3</b>
1.1	Registos de utilizadores . . . . .	3
1.2	Relações entre utilizadores . . . . .	5
1.2.1	Registo de mensagens . . . . .	6

# 1 Estrutura de dados

## 1.1 Registos de utilizadores

Para o registo dos utilizadores decidimos usar tabelas de hash seguindo a opção chaining para o tratamento de colisões. Usando estes métodos conseguimos tempos de acesso aos perfis dos utilizadores perto de tempo constante. Para isso definimos duas funções de hash (para nome e para nif). As tabelas de hash contêm endereços de estruturas User-h que por sua vez terão um endereço de um perfil e um endereço de um novo elemento User-h para o qual a função de hash devolveu o mesmo endereço da tabela de hash. Iremos então usar duas tabelas de hash (hash-nome, hash-nif) para guardar os utilizadores. Decidimos usar este mecanismo para facilitar a pesquisa de utilizadores visto que seria muito difícil encontrar uma função de hash que servi-se para os dois casos. Esperamos assim ter um bom desempenho em termos de pesquisa/acesso a utilizadores quando precisarmos de obter informações sobre estes ou mesmo para os carregar para a base de dados do sistema.

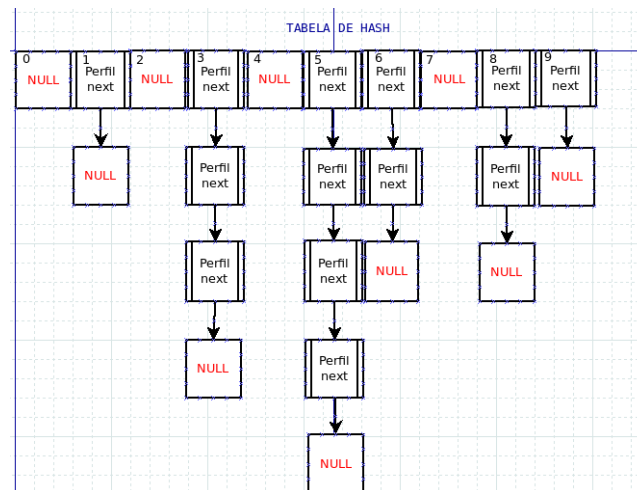


Figura 1: Tabela de Hash

Optamos por esta estrutura de dados pois esperamos ter um tempo de acesso aos perfis dos utilizadores bastante reduzido. O tempo desejado de acesso será um tempo constante. Nesta estrutura é possível observar que existe um campo id. Este campo servirá para sabermos em que posição dos grafos estará guardada a informação sobre os utilizadores que tem algum tipo de relação com um determinado utilizador. Assim se quisermos, por exemplo, ver quais são os utilizadores do utilizador com id 1 basta irmos à tabela das relações de amizade (grafo amigos que é uma tabela de adjacências) e acedermos à posição 1 onde encontraremos todos os amigos deste utilizador.

Quanto a apagar utilizadores, a nossa estrutura perfil possui um campo apagado que quando estiver activo (valor 1), indicará que o perfil foi apagado. Permitirá também que o utilizador recupere o seu perfil visto que os seus dados continuam gravados na base de dados.

## 1.2 Relações entre utilizadores

Para estruturarmos as relações entre utilizadores optamos pela utilização de um grafo para cada relação: Grafo amigos, Grafo família, Grafo profissional. Estes são constituídos por um inteiro que controla o numero de posições ocupadas que será usado no futuro para a alocação de mais espaço caso seja necessário, e por uma tabela de nodos adjacentes onde se guarda a informação relativa às relações.

Seguindo uma implementação destes por lista de adjacência, onde a cada índice do array nodos (corresponde ao id de registo do utilizador) corresponde a lista de amigos deste, contendo os dados : peso da relação, o nif e o id do utilizador e um apontador para a próxima pessoa que crie uma relação com este utilizador.

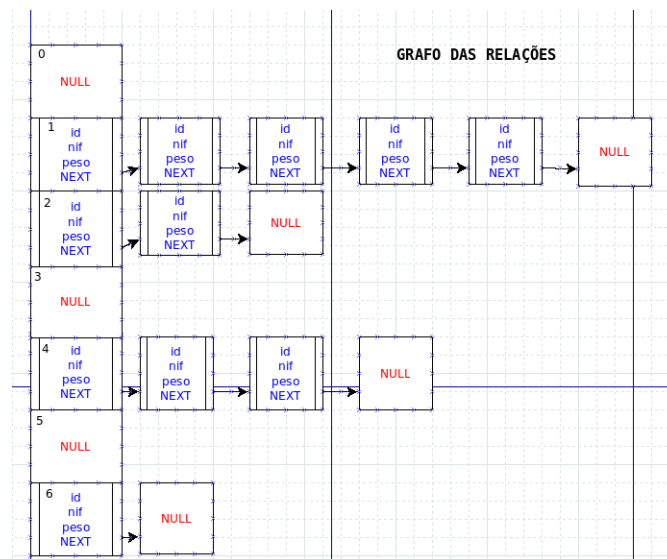


Figura 2: Grafo das Relações

Optamos por esta estrutura de dados pois através do índice do array conseguimos chegar imediatamente ao utilizador que se pretende efectuar operações nas relações. Esperamos ter um acesso rápido aos utilizadores relacionados com um o utilizador, cujo o id corresponde ao índice do array. No pior caso o tempo de acesso será de N utilizadores.

### 1.2.1 Registo de mensagens

Para o registo de mensagens optamos pela utilização de um array onde os índices deste corresponde ao id do utilizador que recebe a mensagem.

O array será então constituído pelo campo Remetente que contem o id e o nif de quem manda a mensagem o numero de mensagens desse mesmo utilizador(para o controlo do máximo de mensagens) e um apontador para as mensagens que corresponde a esse id e um outro apontador para um novo Remetente para podermos assim ter num único índice do array(que corresponde ao id do utilizador) todas as mensagens vindas de diferentes utilizadores.

Através desta estrutura será possível ter acesso de forma rápida a todas as mensagens dos diferentes utilizadores para um mesmo utilizador.

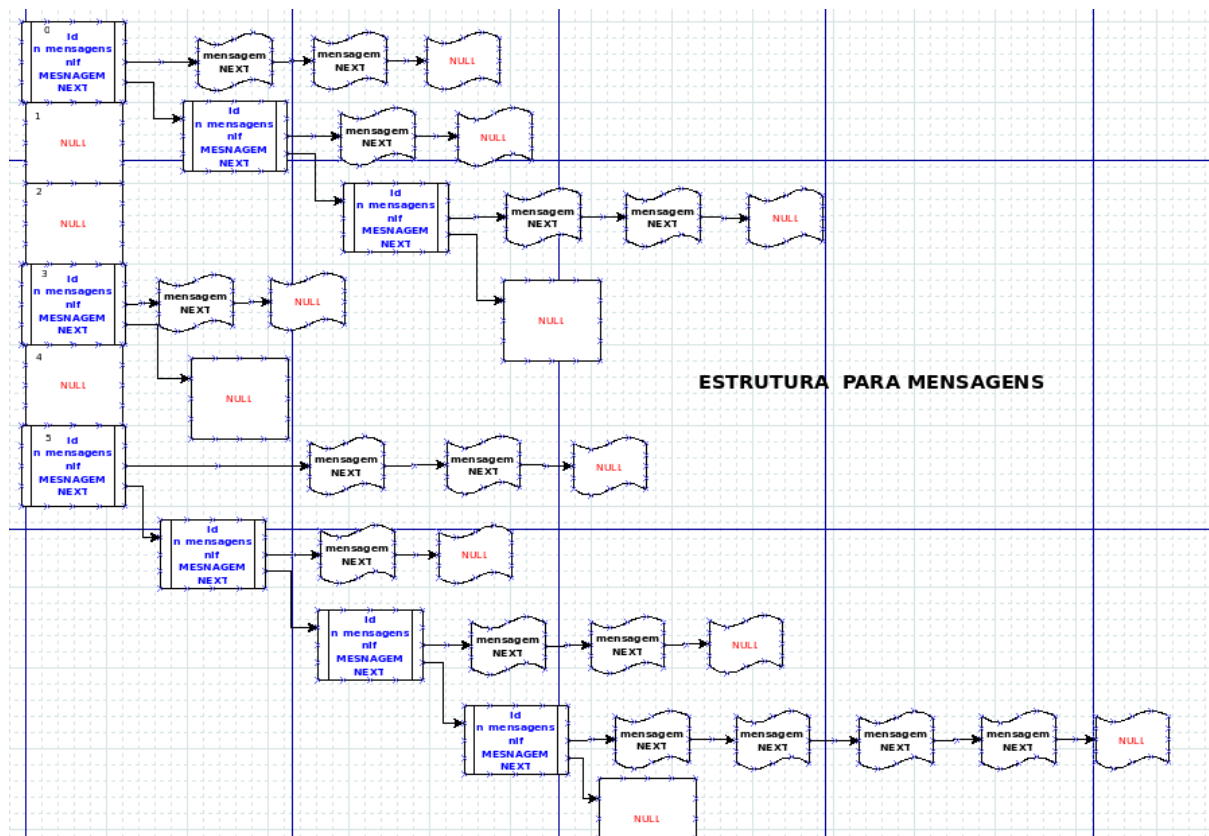


Figura 3: Estrutura das mensagens