

Amigo Oculto 6 - Convites



[Amigo Oculto](#) > 6. Convites

Convidando os amigos



Vamos, agora, criar mais um relacionamento 1:N. Desta vez, será de convites para participação em grupos com o próprio grupo. Cada grupo poderá ter N convites abertos, mas cada convite será para apenas 1 único grupo.

CONVITES



A entidade Convite

O convite terá uma estrutura bem simples pois precisamos apenas registrar os emails dos usuários que convidaremos. Assim, nossa entidade Convite deverá ter os seguintes atributos.

- `int idConvite` - Este atributo identificará cada convite de forma exclusiva. Esse identificador específico é necessário, para que, mais tarde, possamos remover o convite aceito ou recusado da lista.
- `int idGrupo` - Aqui, incluiremos o ID do grupo para o qual este convite foi emitido. Este atributo é que estabelecerá o relacionamento 1:N entre as entidades Convite e Grupo.
- `String email` - Agora, registraremos o email do usuário convidado. Lembre-se que, neste projeto, o usuário só deve usar um único email. Se ele receber o convite por um email e tentar se cadastrar por outro, o sistema não reconhecerá o convite.
- `long momentoConvite` - Apenas para conhecimento dos usuários administrador e convidado para o grupo, este atributo registrará a data e hora em que o convite foi emitido.

- `byte estado` - Finalmente, teremos um atributo que diz qual é o estado desse convite, podendo ser: 0 (pendente), 1 (aceito), 2 (recusado) ou 3 (cancelado). Todo novo convite, portanto, deve ser criado com o estado 0.

Implemente essa entidade, incluindo os métodos da interface `Registro`, incluindo os métodos `getID()`, `setID()`, `chaveSecundaria()`, `toByteArray()` e `fromByteArray()`. O seu método de chave secundária deve retornar a concatenação do ID do grupo com o email do usuário convidado para esse grupo, de tal forma que o usuário possa ser convidado para grupos diferentes, mas não possa receber dois convites para o mesmo grupo.

Listas invertidas

Neste ponto, nós precisaremos de algum mecanismo para resolver dois problemas: o primeiro é não deixar que um convite seja emitido para alguém que já esteja convidado ou que já esteja participando do grupo. Resolveremos isso por meio da busca pela chave secundária do CRUD, exatamente como fizemos na criação das sugestões e dos grupos - apenas uma concatenação entre o ID do grupo e o email do usuário convidado. Então, não há nada a ser implementado, a não ser o teste para ver se esse valor já existe no CRUD.

O segundo problema é conseguirmos visualizar para quais grupos um usuário está com convites pendentes. Essa busca não será implementada neste módulo, mas já precisamos deixar a estrutura de dados que permitirá esse tipo de busca pronta. Ela deve nos permitir buscar, para um determinado email, quais são os IDs dos convites emitidos. Como os convites podem ser emitidos para usuários que ainda não estão cadastrados no sistema, não podemos criar uma simples árvore B+ que associasse o ID do usuário ao ID do grupo. Precisamos, realmente, trabalhar com o email neste momento.

Usaremos, assim, um atributo da entidade (email) para recuperarmos o ID dessa própria identidade. E eu acho que você já é capaz de imaginar qual é a estrutura adequada para isso, não é? Sim, isso mesmo. uma lista invertida. Por questão de simplicidade, usaremos uma árvore B+ para implementar a nossa lista invertida de convites pendentes. Mas você pode se sentir a vontade para usar outra estrutura. Essa será uma árvore B+ que armazena pares de valores dos tipos [string, int]:

[lista.invertida.zip](https://pucminas.instructure.com/courses/9019/files/864843/download?wrap=1) (<https://pucminas.instructure.com/courses/9019/files/864843/download?wrap=1>) (<https://pucminas.instructure.com/courses/9019/files/864843/download?wrap=1>)

Emissão e cancelamento de convites

Como você deve se lembrar, cuidaremos agora da segunda opção do menu de gerenciamento dos grupos: a emissão e o cancelamento de convites.

A partir do [menu de gerenciamento de grupos](#), o usuário deve poder acessar o menu de convites com as seguintes opções:

```
INÍCIO > GRUPOS > GERENCIAMENTO DE GRUPOS > CONVITES
```

- 1) Listagem dos convites
- 2) Emissão de convites
- 3) Cancelamento de convites

- 0) Retornar ao menu anterior

Opção: _

O que precisamos implementar aqui é bem simples e parecido com o código do CRUD de sugestões.

Listagem

A listagem deverá gerar uma lista dos convites deste grupos, independentemente do seu estado. A listagem, como vimos antes, não apresentará os IDs dos grupo ou do convite, mas será uma listagem sequencial dos convites. Todo o processo, porém, deve ser iniciado com a seleção de qual grupo se deseja ver a lista de convites. Assim, precisamos usar o método de listagem de grupos, como vimos na página anterior:

```
ESCOLHA O GRUPO:
```

- 1. Ciência da Computação PUC Minas
- 2. Família Silveira
- 3. Turma do Truco

Grupo: _

Depois que o usuário, escolher o grupo, será possível apresentar a lista de convites:

```
CONVITES DO GRUPO "TURMA DO TRUCO"
```

- 1. fulano@gmail.com (16/02/2020 19:00 - pendente)
- 2. beltrano@hotmail.com (16/02/2020 19:10 - cancelado)
- 3. beltrano@gmail.com (17/02/2020 08:15 - aceito)
- 4. sicrano@outlook.com (18/02/2020 14:00 - aceito)

Pressione qualquer tecla para continuar...

Os passos para a listagem são:

1. Obter a lista de IDs de grupos na árvore B+ de grupos usando o ID do usuário;
2. Para cada ID nessa lista,
 1. Obter os dados do grupo usando o método *read(ID)* do CRUD;
 2. Se o grupo estiver ativo, apresentar o nome (e outros dados desejados) do sugestão na tela;
3. Solicitar do usuário o número do grupo do qual deseja visualizar os convites;
 1. Se o usuário digitar 0, retornar ao menu de gerenciamento de convites;

4. Usando o ID do grupo escolhido, recuperar os dados do grupo usando o método *read(ID)* do CRUD;
5. Apresentar o nome do grupo na tela;
6. Obter a lista de IDs de convites desse grupo na árvore B+ de convites, usando o ID do grupo escolhido;
7. Para cada ID nessa lista,
 1. Obter os dados do convite usando o método *read(ID)* do CRUD;
 2. Apresentar o email e o estado do convite na tela.

Você deve ter percebido que também precisaremos de uma árvore B+ para gerenciar o relacionamento de ID de grupo, com ID de convite, certo? Como já discutimos esse uso nos CRUDs anteriores, não entraremos em detalhes sobre isso aqui.

Emissão de convites

A emissão de convites, nada mais que é a inclusão de novas entidades no arquivo, por meio do CRUD. No entanto, precisamos assegurar que o convite para um email ainda não existe antes de incluí-lo.

Tudo começa com a listagem dos grupos, para que o usuário escolha para qual grupo deseja emitir convites. É importante que o grupo esteja ativo e que o sorteio ainda não tenha sido realizado para ele (Não permitiremos a emissão de convites para grupos cujo sorteio já tenha sido feito). Em seguida, basta ele informar os emails para os quais deseja enviar convites. O sistema deverá, automaticamente, informar se o email estiver repetido.

O processo ocorre de acordo com a seguinte sequência de passos:

1. Obter a lista de IDs de grupos na árvore B+ de grupos usando o ID do usuário;
2. Para cada ID nessa lista,
 1. Obter os dados do grupo usando o método *read(ID)* do CRUD;
 2. Se o grupo estiver ativo e o sorteio ainda não tiver sido realizado, apresentar o nome (e outros dados desejados) do grupo na tela;
3. Solicitar do usuário o número do grupo do qual deseja visualizar os convites;
 1. Se o usuário digitar 0, retornar ao menu de gerenciamento de convites;
4. Usando o ID do grupo escolhido, recuperar os dados do grupo usando o método *read(ID)* do CRUD;
5. Apresentar o nome do grupo na tela;
6. Iniciar laço de criação de convites;
 1. Solicitar o email do usuário a ser convidado;
 1. Se o usuário deixar o email em branco, retornar ao menu de gerenciamento de convites;
 2. Usando o método *read(chave secundária)* do CRUD, testar se a combinação ID grupo + email já existe;
 1. Se existir,
 1. Se o estado for de pendente ou aceito, apresentar mensagem de que convite já

- toí emitido para esse email e avançar para a próxima repetição do laço;
- 2. Se o estado for de recusado ou cancelado, apresentar mensagem correspondente e solicitar confirmação do usuário para reemissão do convite;
 - 1. Se o usuário não confirmar a reemissão, avançar para a próxima repetição do laço;
- 3. Se não existir, incluir o convite no arquivo, usando o método *create()* do CRUD (o convite deve ser criado com o ID do grupo selecionado, data e hora atuais e o estado de pendente).
- 4. Inserir o par email e ID do novo convite na lista invertida de convites pendentes;
- 5. Inserir o par ID do grupo e ID do novo convite na árvore B+ desse relacionamento (no caso de reemissão, esse par já estará na árvore).

Observe que não criaremos uma rotina real de envio de emails. Neste projeto, estamos mais preocupados com o gerenciamento dos arquivos. Em um projeto real, talvez fosse até mais interessante trabalhar com números de WhatsApp do que emails e os convites deveriam realmente ser enviados aos usuários finais.

Cancelamento de convites

O cancelamento de convites é uma alteração em que apenas o atributo estado é alterado (passando do valor 0 para o valor 3). A aceitação ou recusa de convites é feita por cada usuário que foi convidado. Assim, as únicas operações que o administrador do grupo pode fazer é criar novos convites ou cancelar convites já emitidos (inclusive no caso de emails errados, pois, teoricamente, uma mensagem já teria sido enviada para o destinatário).

Como você deve estar imaginando, a sequência de passos começa com os mesmos procedimentos da listagem (apenas para convites pendentes) e apenas pergunta o número do convite que o administrador deseja cancelar:

- 1. Obter a lista de IDs de grupos na árvore B+ de grupos usando o ID do usuário;
- 2. Para cada ID nessa lista,
 - 1. Obter os dados do grupo usando o método *read(ID)* do CRUD;
 - 2. Se o grupo estiver ativo e o sorteio não tiver sido realizado, apresentar o nome (e outros dados desejados) do sugestão na tela;
- 3. Solicitar do usuário o número do grupo do qual deseja visualizar os convites;
 - 1. Se o usuário digitar 0, retornar ao menu de gerenciamento de convites;
- 4. Usando o ID do grupo escolhido, recuperar os dados do grupo usando o método *read(ID)* do CRUD;
- 5. Apresentar o nome do grupo na tela;
- 6. Obter a lista de IDs de convites desse grupo na árvore B+ de convites, usando o ID do grupo escolhido;
- 7. Para cada ID nessa lista,
 - 1. Obter os dados do convite usando o método *read(ID)* do CRUD;
 - 2. Se o convite estive pendente, apresentar o email usando para no convite.
- 8. Solicitar do usuário o número do convite que deseja cancelar;

1. Se o usuário digitar 0, retornar ao menu de gerenciamento de convites;
9. Solicitar a confirmação de cancelamento do convite (mostrando novamente o email usado nesse convite);
 1. Se o usuário não confirmar o cancelamento, voltar ao menu de gerenciamento de convites;
10. Alterar os dados do convite por meio do método *update()* do CRUD. O único atributo que será alterado é o atributo `estado` cujo valor deve ser 3;
11. Remover o par email e ID do convite da lista invertida de convites pendentes;
12. Apresentar mensagem de confirmação de cancelamento;
13. Voltar ao menu de gerenciamento de convites.

Na próxima página, veremos como um usuário aceita ou recusa um convite.