

# Amigo Oculto 3 - Acesso



[Amigo Oculto](#)> 3. Acesso

## Acesso ao sistema



Cada usuário do sistema deverá se cadastrar por conta própria, isto é, não há um *administrador do sistema* que fará o cadastro de usuários. Assim, precisamos implementar rotinas para o acesso ao sistema, incluindo o primeiro acesso, por meio do qual o usuário se cadastrará. Apenas após cadastrado no sistema é que o usuário poderá participar de amigos ocultos.

### ACESSO



## As formas de acesso

Para participar da brincadeira de amigo oculto, cada usuário precisará estar cadastrado no sistema. Não haverá uma forma de participação no amigo oculto que não seja por meio do acesso pessoal de cada um dos usuários participantes.

O controle de acesso será feito por meio do **email** do usuário e da sua **senha**. Se o usuário já estiver cadastrado, bastará informar esses dados para ter acesso. Caso contrário, ele deverá, então, se cadastrar no sistema. Como ainda não trabalharemos com interfaces gráficas, você pode implementar um menu tão simples quanto este:

```
AMIGO OCULTO 1.0
=====
```

## ACESSO

```
1) Acesso ao sistema
2) Novo usuário (primeiro acesso)

0) Sair

Opção: _
```

É uma interface muito simples, mas suficiente para as nossas necessidades. Nesta disciplina, estaremos mais focados na implementação das estruturas de dados em arquivos do que em interfaces gráficas. Para a construção de uma interface simples assim, seus conhecimentos de AED1 e AED2 são suficientes. Se você quiser fazer algo diferente, considere esse seguinte desafio:



**DESAFIO 3:** Crie uma terceira opção no menu de acesso que permita ao usuário cadastrar uma sua senha, caso ele tenha esquecido a atual. Para isso, você deveria simular uma operação de envio de email com uma nova senha temporária.

## O primeiro acesso

Os usuários só serão cadastrados no sistema por meio da rotina de primeiro acesso, pois cada usuário deverá se cadastrar pessoalmente. Talvez algum usuário não chegue a se cadastrar até a data do sorteio do amigo oculto. Assim, a operação de sorteio deverá excluir da brincadeira todos aqueles que não tiverem se cadastrado. É uma pena, mas precisamos de algumas regras. Se ele nem se der ao trabalho de entrar no sistema, quem garante que ele vai se dar ao trabalho de comprar um presente, não é?

Enfim, o novo usuário precisará apenas informar um email válido, seu nome e uma senha para futuros acessos. Aqui é importante considerar que **o email é um identificador exclusivo de cada usuário**, isto é, não podemos ter dois usuários com o mesmo email e nem um usuário com dois emails diferentes. Nesse último caso, o sistema considerará um usuário cadastrado com 2 emails como 2 pessoas diferentes.

Assim, a primeira ação do usuário é informar o email que usará no sistema. Sempre que alguém for convidá-lo para um grupo de amigo oculto, deve usar exatamente esse email, pois o convite não será associado a uma pessoa, a um nome ou a um CPF, mas será associado a um email. Se o usuário se cadastrar com um email (por exemplo, *fulano@gmail.com*) e for convidado por meio de outro email (por exemplo, *fulano@hotmail.com*), o sistema não terá como saber que se trata da mesma pessoa. Quem sabe uma versão 2 do nosso sistema não permita o cadastro de emails alternativos ou até mesmo o cadastro por meio do WhatsApp?

Você pode implementar uma tela como a abaixo para a leitura do email:

NOVO USUÁRIO

Email: \_

Após o usuário informar o seu email, você deve checar se esse email já existe. Se existir, você deve informar uma mensagem de erro e solicitar um novo email. Se o email não existir, então você pode solicitar os demais dados: nome e senha. Ao final, redirecione o usuário para o menu de acesso para que ele faça o primeiro acesso informando o email e a senha cadastrados.

No fim das contas, a sua rotina de novo usuário deve ser mais ou menos assim:

1. Solicitar o email do novo usuário;
  1. Se o email informado estiver em branco (nenhum valor preenchido), retornar ao menu de acesso;
2. Buscar se um usuário com esse email já existe, por meio do método *read(email)* do CRUD;
3. Se o email já existir para algum usuário,
  1. Apresentar mensagem de email já cadastrado;
  2. Voltar ao passo 1 do laço principal;
4. Se o email não existir,
  1. Solicitar o nome do usuário;
  2. Solicitar a senha do usuário;
  3. Solicitar a confirmação da inclusão do novo usuário com esses dados;
    1. Se o usuário não confirmar a inclusão, voltar ao menu de acesso;
  4. Incluir o usuário no arquivo, por meio do método *create()* do CRUD;
  5. Apresentar mensagem de confirmação da inclusão;
  6. Voltar ao menu de acesso.

Observe que essa rotina não tem acesso direto ao arquivo de usuários. Quaisquer acessos a ele precisam ser feitos por meio do CRUD de usuários.

## O acesso de usuários já cadastrados

Os usuários já cadastrados poderão ir à outra opção do menu de acesso, que deverá validar o nome e a senha do usuário. A tela inicial dessa operação deve ser algo como essa:

ACESSO AO SISTEMA

Email: \_

A operação aqui é bastante simples, bastando implementar os seguintes passos:

1. Solicitar o email do usuário;
2. Buscar os dados do usuário por meio do método *read(email)* do CRUD;
3. Se não existir um usuário com esse email,
  1. Mostrar mensagem de erro;

2. Retornar ao passo 1 do laço principal;
4. Solicitar a senha do usuário;
5. Comparar essa senha informada pelo usuário com a senha recuperada pelo CRUD no passo 2;
6. Se as senhas forem diferentes,
  1. Mostrar mensagem de erro;
  2. Retornar ao passo 1 do laço principal;
7. Se as senhas forem iguais,
  1. Redirecionar o usuário para a tela principal (mas que ainda não existe).

Como você vê, as operações de primeiro acesso e de acesso são bem simples. Considere melhorar a sua tela e construa boas mensagens de erro. Lembre que toda mensagem apresentada deve dar uma chance do usuário ler o que foi escrito, antes de ele tomar alguma providência. Eventualmente, coloque algum comando do tipo "pressione qualquer tecla para continuar...".

Nas próximas páginas, trabalharemos na construção do menu principal e das suas operações.