1. Construir um formulário para criação de usuário e senha:
2. Criar um formulário com os seguintes campos:
   * Usuário (user)
   * Email (email)
   * Senha (password)
   * Confirmar senha (confirm\_password)
   * Botão Ok (user\_create)
3. O campo password e confirm\_password devem “esconder” os caracteres digitados pelo usuário.
4. Criar uma função jquery para validar se os campos password e confirm\_password são iguais. Caso os campos sejam diferentes, apresentar uma mensagem de erro ao usuário informando que as senhas não conferem.
5. Caso os campos password e confirm\_password forem iguais, validar se o campo password tem no mínimo 8 caracteres. Caso seja menor que 8 caracteres, apresentar uma mensagem de erro ao usuário informando que a senha não tem o mínimo de 8 caracteres.
6. Criar uma função jquery para validar o campo email para que seja um email válido. Caso o email não seja válido, apresentar uma mensagem de erro ao usuário informando que o email é inválido.
7. Caso não tenha nenhum erro de validação, criar uma função em PHP para enviar os dados do formulário para uma tabela chamada **users** com os seguintes campos:
   * Id - integer
   * User – varchar(64)
   * Password – varchar(32)
   * Email – varchar (320)
   * Grupo – integer 🡪 foreign key do id da tabela **groups para id 2**
     1. Criar a tabela **groups** com
     2. Id - integer
     3. Nome - varchar (64)
     4. Descrição – varchar (320)
     5. Id = 1 Adminstrador
     6. Id = 2 Usuários
8. Antes de fazer o insert na base de dados, o campo password deve ser criptografado usando a função **md5**.
9. Antes de inserir na base de dados, crie uma função para validar se o campo user está cadastrado. Caso o usuário já estiver cadastrado, apresentar uma mensagem de erro ao usuário informando que o campo user escolhido já foi utilizado e que ele deve inserir um diferente. Para realizar isso você precisa executar um select na tabela **users**.
10. Caso o usuário não exista na tabela **users**, crie uma função para validar se o email está cadastrado. Caso o email já estiver cadastrado, apresentar uma mensagem de erro ao usuário informando que o campo email escolhido já foi utilizado e que ele deve inserir um diferente. Para realizar isso você precisa executar um select na tabela **users**.
11. Caso user e email sejam únicos na tabela **users**, crie uma função para inserir na tabela **users** os dados obtidos no formulário e retorne uma mensagem de sucesso ao usuário que a criação do usuário foi concluída com sucesso.
12. Construir um formulário para login:
13. Criar um formulário com os seguintes campos:
    * Usuário (user)
    * Senha (password)
    * Botão “Entrar” (user\_login)
    * Botão “Esqueci minha senha” (user\_forgot\_pass)
14. Criar uma função em jquery para validar se o password tem 8 caracteres. Caso seja menor que 8 caracteres, apresentar uma mensagem de erro ao usuário informando que a senha não tem o mínimo de 8 caracteres.
15. Caso o usuário tenha digitado uma senha maior que 8 caracteres, faça uma função em PHP para verificar se o usuário informado está relacionado com a senha conforme armazenados na tabela **users**. Lembre que antes de fazer o select na tabela, o campo password deve ser criptografado com a função **md5.**
16. Caso o usuário e senha não estejam relacionados, apresentar uma mensagem de erro ao usuário informando que “Usuário e/ou senha inválidos”.
17. Caso o usuário e senha estejam relacionados, apresentar uma mensagem de sucesso informando “Bem vindo!”.
18. Caso o usuário tenha pressionado no botão “Esqueci minha senha”, apresente um novo formulário ao cliente com um campo email para que ele digite o email.
19. Criar uma função jquery para validar o campo email para que seja um email válido. Caso o email não seja válido, apresentar uma mensagem de erro ao usuário informando que o email é inválido.
20. Enviar para o email informado um link com para a troca de senha.
21. O usuário ao clicar no link recebido, é direcionado a uma página para troca de senha. Será apresentado um formulário com dois campos:
    * Email (email)
    * Senha (password)
    * Confirmar senha (confirm\_password)
    * Botão Ok (user\_change\_pass)
22. Criar uma função jquery para validar se os campos password e confirm\_password são iguais. Caso os campos sejam diferentes, apresentar uma mensagem de erro ao usuário informando que as senhas não conferem.
23. Caso os campos password e confirm\_password forem iguais, validar se o campo password tem no mínimo 8 caracteres. Caso seja menor que 8 caracteres, apresentar uma mensagem de erro ao usuário informando que a senha não tem o mínimo de 8 caracteres.
24. Caso o usuário não exista na tabela **users**, crie uma função para validar se o email está cadastrado. Caso o email já estiver cadastrado, apresentar uma mensagem de erro ao usuário informando que o campo email escolhido já foi utilizado e que ele deve inserir um diferente. Para realizar isso você precisa executar um select na tabela **users**.
25. Caso user e email sejam únicos na tabela **users**, crie uma função para inserir na tabela **users** os dados obtidos no formulário e retorne uma mensagem de sucesso ao usuário que a criação do usuário foi concluída com sucesso.
26. Criar uma função em PHP para inserir na tabela **users** de dados após o password estar validado. Apresentar mensagem ao usuário informando que a senha foi alterada com sucesso.
27. Formulário de administração de usuários
28. Somente usuários que participem do grupo administrador podem entrar nesta tela. (group id = 2).
    * Ao fazer o login, o array $\_SESSION deve ter um campo para guardar o id, user e groupId do usuário.
    * Antes de entrar nesta tela, fazer a verificação de apresentação usando os campos armazenados na $\_SESSION
29. Criar uma tela com uma tabela que apresente todos os usuários cadastrados no na tabela **users**:

* Id (id)
* Usuário (user)
* Senha (password)
* Email (email)
* Grupo (group)
* Ativo (active)

1. A tela tem um botão **+** para adicionar um novo usuário
   * A tela é similar ao formulário do requisito 1
2. A tela deve ter um botão Pesquisar para pesquisar pelo usuário
3. Para cada usuário existe um botão Editar e outro Desativar na mesma linha.
   * Ao editar um usuário, podem ser alterados apenas:
     1. Senha
     2. Email
     3. Grupo
     4. Ativar ou desativar
   * Botão apresenta o texto Ativar se estiver desativado e Desativar se estiver ativo
4. Disparo de email para todos os usuários
5. Somente usuários que participem do grupo administrador podem entrar nesta tela. (group id = 2). Mesma lógica do requisito 3.
6. Criar um botão Enviar Mensagem
   1. Oferecer a opção de escolher usuários individualmente que podem receber email
      1. Apresentar todo os usuários, como no requisito 3, menos os botões Editar, Ativar/Desativas e +.
   2. Oferecer a opção de escolher os grupos que vão receber os emails
   3. Oferecer um campo para escrever o título do email
   4. Oferecer um campo para escrever o corpo do email
7. Botão de envio da mensagem 🡪 class PHPMailer
8. Cadastro de produtos
9. Criar um formulário para manutenção de cadastro de produtos.
10. Criar uma tabela na base de dados chamada **products** para armazenar os produtos com os seguintes campos
    * Id (id)
    * Código (code)
    * Nome (shortname)
    * Descrição (description)
    * Categoria (categoryId)
    * Foto (photo)
    * Preço (price)
    * Quantidade em estoque (stock)
    * Ativo (active)
11. A tela deve ter um botão **+** para adicionar um novo produto
12. A tela tem um botão Pesquisar para pesquisar um produto pelo nome
13. Para cada produto existe um botão Editar e outro Ativar/Desativar na mesma linha
    * Ao editar um produto é possível alterar:
      1. Nome
      2. Descrição
      3. Categoria
      4. Foto (adicionar ou remover fotos)
      5. Preço
      6. Quantidade em estoque
    * Botão apresenta o texto Ativar se estiver desativado e Desativar se estiver ativo
14. Cadastro de categorias
15. Criar um formulário para manutenção de cadastro de categorias.
16. Criar uma tabela na base de dados chamada **categories** para armazenar as categorias com os seguintes campos
    * Id (id)
    * Código (code)
    * Nome (shortname)
    * Descrição (description)
    * Quantidade de Fotos (photoQuant)
    * Estoque Mínimo (minimumStockLevel)
    * Ativo (active)
17. A tela deve ter um botão **+** para adicionar uma nova categoria
18. A tela tem um botão Pesquisar para pesquisar uma categoria pelo nome
19. Para cada usuário existe um botão Editar e outro Desativar na mesma linha do produto.
    * Uma categoria que é desativada irá desativar todos os produtos relacionados a ela
    * Ao editar um produto é possível alterar:
      1. Nome
      2. Descrição
      3. Quantidade de Fotos
      4. Estoque mínimo
    * Botão apresenta o texto Ativar se estiver a categoria estiver desativada e Desativar se estiver ativa

- MD5

- regex para PHP

- regex para jquery

- função de banco para execução de queries