

Seletiva WS Paraná
#09 TI Soluções de Software
Prova #06 – Web

Cenário

Sua empresa deseja criar um site para uso interno onde todos os funcionários poderão acessar e verificar padrões de implementação para diversos tipos de problema. Por exemplo, ao se fazer requisições http a empresa gosta de usar certas nomenclaturas e bibliotecas, assim, todo desenvolvedor pode checar a documentação interna para realizar a implementação.

Seu trabalho é desenvolver o banco de dados, as páginas html do sistema usando react, angular ou qualquer outra estrutura e o servidor em C# ou Java deste sistema. Você não precisa se preocupar com detalhes de implantação que serão realizados por outra equipe.

A estrutura do banco de dados fica a seu critério bem como detalhes de interface.

Login

Para garantir a segurança do sistema deve existir um sistema de login, porém, não deve haver possibilidade de cadastrar uma conta, isso será feito apenas pelos administradores. Assim, deve existir uma *procedure* no banco de dados chamada *register* que recebe nome de usuário, e-mail e senha criptografada do usuário. Contas novas serão criadas a partir desta função.

No sistema, o login será realizado com o e-mail ou nome de usuário mais a senha. O servidor deve disponibilizar, caso o login ocorra com sucesso, um *token* de segurança associado ao usuário. Todas as operações realizadas pelo servidor precisam receber apenas o *token* e outros parâmetros úteis a operação, mas nenhuma outra informação do usuário. Nenhuma operação deve funcionar se entregue um *token* inválido. Deve existir ainda uma *procedure* chamada *clear* que limpará todos os *token* do sistema. No processo de login a senha deve ser criptografada ao ser comparada no banco de dados. Desta forma para que a equipe de testes possa verificar se a criptografia funciona, exponha a *url* do servidor para realização do login e como usá-la em um arquivo chamado *endpoints.txt*.

Adicionar página

Dentro do sistema o usuário poderá clicar na opção “Adicionar Página” que permitirá que ele crie sua própria página de instruções. Assim ele terá duas opções: Escrever a página numa guia de edição ou carregar um arquivo do computador com a extensão .wiki.

Caso escreva o código na própria página ele também precisará incluir um nome para a página. No outro caso o nome do arquivo será o nome da página. Logo, só resta clicar no botão “salvar”. Em seguida a página adicionada é imediatamente aberta.

Apresentação de Páginas .Wiki

Páginas .wiki, adicionadas do computador ou escritas no próprio navegador tem uma estrutura específica que deve ser respeitada em sua visualização.

- Caso o usuário coloque algo entre asteriscos (*) o texto visualizado deve ficar em negrito.
- Caso o usuário coloque algo entre tios (~) o texto visualizado deve ficar em itálico.
- Caso o usuário digite *newline* deve-se pular para próxima linha. Caso contrário a quebra de linha é automática no tamanho da página e precisa ser responsivo.
- Caso o usuário digite uma *url* entre colchetes ([e]) deve-se colocar a imagem da *url* no documento.
- Caso o usuário digite algo entre parênteses seguido de uma *url* em colchetes, deve-se criar um hyperlink.

- Caso colocado uma exclamação dupla (!!), todo texto na mesma linha deve ser colocado com uma fonte maior, indicando título.
- Caso o usuário digite @ seguido de chaves ({ e }) deve-se abrir um bloco de código. Ele deve mostrar um contador de linhas e as regras acima não são aplicados, pois trata-se de um exemplo de código.

Abaixo você pode ver um exemplo de código de página e seu resultado:

```
!!Minha página de Wiki
*A seguir está detalhado como fazer
uma
página
de wiki*
newline
@{
  !!Minha página de Wiki
  _código em C abaixo_
  @{
    int i = 0;
    i = 8;
  }
}
(~pesquisar no google~)[www.google.com.br]
```



Minha página de Wiki

A seguir está detalhado como fazer uma página de wiki

```
1.!!Minha página de Wiki
2._código em C abaixo_
3. @{
4.   int i = 0;
5.   i = 8;
6. }
7.
```

[pesquisar no google](#)

Note que todas as regras foram ignoradas dentro de um bloco de código. Além disso, após um título a linha pulou automaticamente.

Deixe a *url* para fazer uma requisição pedindo uma página que espera o nome da página e o *token* exposta para permitir que seja testado pela equipe de testes em um arquivo chamado *endpoints.txt* e entregue junto com a aplicação.

Qualquer página adicionada ao sistema deverá poder ser carregada no site com uma *url* com seu nome incluso, os detalhes de como isso é construído fica a critério do programador.

Sistema de Pesquisa

Seu sistema deve possuir alguma forma de pesquisa, que lista todas as páginas que contenham a sequência digitada presente em seu nome ou código .wiki. A lista de elementos deve ser atualizada enquanto o código é digitado. Além disso, a *url* que pesquisa recebendo apenas o *token* de segurança deve ficar exposta e também documentada no *endpoints.txt* para realização de testes.

Página Pessoal

O usuário ainda poderá acessar seus dados pessoais clicando em uma página chamada perfil. A página pode ser editada clicando em *editar*, porém, deve apresentar uma mensagem de erro caso o usuário não seja o dono daquela conta, visto que qualquer página pessoal deve poder ser encontrada no sistema de pesquisa, considerando o nome do usuário como nome da página.

Ao criar uma conta/primeiro acesso a página pessoal, a página .wiki padrão deve ser mostrada como o padrão para páginas ainda não editadas:

Nome: *nome do usuário*

Email: *e-mail do usuário*

Diagrama de Banco de Dados

Para facilitar o trabalho das equipes de teste disponibilize MER ou DER para que o banco dados possa ser facilmente compreendido. Entregue como um arquivo *diagrama.bmp* ou qualquer outra extensão que consiga ser aberta em um software de visualização de imagens.

Entregas

Nesta entrega será esperado:

- Pasta *Frontend* com as páginas html/projeto react/angular mais um tutorial de como rodar o site chamado *howstart.txt*
- Pasta *Backend* com o servidor mais um tutorial de como rodar o site chamado *howstart.txt*
- Pasta *Database* com arquivo de *script* para gerar banco de dados e *procedures* pedidas.
- Arquivo *endpoints.txt*.
- Arquivo *diagrama.bmp*.