

Documentação técnica

Classe BancoDAO

Seu principal papel é fazer a conexão com o banco de dados e executar queries.

Métodos:

__construct:

- Não possui parâmetros
- RETURN: a conexão com o banco. Utiliza das constantes da classe para fazer a conexão

executeQuery:

- Parâmetros:
 - \$sql: Espera uma sql para ser executada
- RETURN: se o resultado da query for apenas uma linha do banco, então retorna um dicionário simples. Se for um Array de dicionário, retorna esse array.

Classe Candidato

Seu principal papel é fazer a organização de cada candidato.

Métodos:

__construct:

- Parâmetros:
 - \$nome: nome do candidato (String)
 - \$cpf: cpf do candidato (String)
 - \$dataNascimento: Data de nascimento do candidato (String)
 - \$profissoes: Profissoes do candidato (String)
- Atribui os parâmetros para o objeto da classe

getCpf:

- Não possui parâmetros
- RETURN: Retorna o cpf, em string, que foi atribuído no construtor do objeto da classe

getNome:

- Não possui parâmetros

- RETURN: Retorna o nome, em string, que foi atribuído no construtor do objeto da classe

getProfissoesString:

- Não possui parâmetros
- RETURN: Retorna as profissões, em string, que foi atribuído no objeto da classe

getProfissoes:

- Não possui parâmetros
- RETURN: Retorna as profissões, em array, que foi atribuído no objeto da classe

Classe Concurso

Seu principal papel é fazer a organização de cada concurso.

Métodos:

__construct:

- Parâmetros:
 - \$orgao: Órgão do concurso (String)
 - \$edital: Edital do concurso (String)
 - \$codConcurso: Código do concurso (String)
 - \$listaDeVagas: Lista de Vagas do concurso (String)
- Atribui os parâmetros para o objeto da classe

getOrgao:

- Não possui parâmetros
- RETURN: Retorna o órgão, em string, que foi atribuído no objeto da classe

getEdital:

- Não possui parâmetros
- RETURN: Retorna o edital, em string, que foi atribuído no objeto da classe

getCodigo:

- Não possui parâmetros

- RETURN: Retorna o Código do Concurso, em string, que foi atribuído no objeto da classe

getListaDeVagasString:

- Não possui parâmetros
- RETURN: Retorna a Lista de Vagas, em string, que foi atribuído no objeto da classe

getListaDeVagas:

- Não possui parâmetros
- RETURN: Retorna as profissões, em array, que foi atribuído no objeto da classe

Classe Paginação

Seu principal papel é pegar os resultados buscados no banco de dados e exibir em forma de tabela e com limite de exibição por página.

Métodos:

__construct:

- Parâmetros:
 - \$candidatoResult: Órgão do concurso (Dicionário ou Array de Dicionário)
 - \$concursoResult: Edital do concurso (Dicionário ou Array de Dicionário)
 - \$entidade: Entidade que define a parte do código que será executada (String)
 - \$pagina: Pagina que o usuário está (Inteiro)
- Atribui os parâmetros para o objeto da classe e define o início e final da exibição dos resultados da busca. (O início e final são usados como índice para \$candidatoResult e \$concursoResult)

getIntervalo:

- Não possui parâmetros
- RETURN: Retorna o Intervalo definido da classe

setPagination:

- Não possui parâmetros

- Dependendo da entidade atribuída ao objeto da classe, chamará o método “setCandidatoPagination()” ou “setConcursoPagination()”

setCandidatoPagination:

- Não possui parâmetros
- Imprime dentro de uma tabela HTML os concursos, cujo, as vagas batem com as profissões do candidato que foi procurado

setConcursoPagination:

- Não possui parâmetros
- Imprime dentro de uma tabela HTML os candidatos, cujo, as profissões batem com as vagas do concurso que foi procurado

Arquivo control/function.php

Seu principal papel é guardar as funções que possam ser usadas nos outros arquivos de controle.

Funções:

redirectMsg:

- Parâmetros:
 - \$msg: Mensagem que aparecerá na tela
 - \$time: Tempo que demorará para atualizar a página
 - \$url: Url que será reencaminhado depois do tempo de espera
- Era imprimir na tela a mensagem e depois do tempo de espera passado por parâmetro, irá redirecionar para a url.

normalizarCPF:

- Parâmetros:
 - \$cpf: CPF de input.
- RETURN: Caso o usuário entre com o formato de cpf de apenas números, a função retornará o cpf em formato padrão (ex: 111.222.333-44)

Arquivo control/cdValidate.php

Seu papel é validar a entrada de dados dentro do input da view ‘concursos.php’.

O usuário só é permitido entrar com números.

Arquivo control/cpfValidate.php

Seu papel é validar a entrada de dados dentro do input da view 'candidatos.php'.

O usuário só é permitido entrar com números, caso a quantidade de caracteres seja 11 (ex: nnnnnnnnnnn. Seja 'n' um número qualquer).

Se a quantidade de caracteres seja 14, o usuário devera entrar com o formato nnn.nnn.nnn-nn. Seja 'n' um número qualquer.

Qualquer outro caso não é permitido e uma mensagem de erro será exibida.

Arquivo control/tableController.php

Seu papel é fazer o gerenciamento das tabelas, seja dos concursos ou dos candidatos.

Ele acha o candidato/concurso, exhibe na tela e cria o objeto da classe '*Paginacao*' para imprimir na tela os resultados achados dentro de uma paginação pré-definida.

Arquivo control/pageController.php

Seu papel é fazer o gerenciamento dos botões da paginação da tabela.

Esse arquivo herda alguns valores do arquivo '*tableController.php*'.

Esse arquivo faz com que seja exibido um número máximo de botões (números) da paginação. Também faz com que a página que o usuário está fique centrada no meio do menu da paginação.