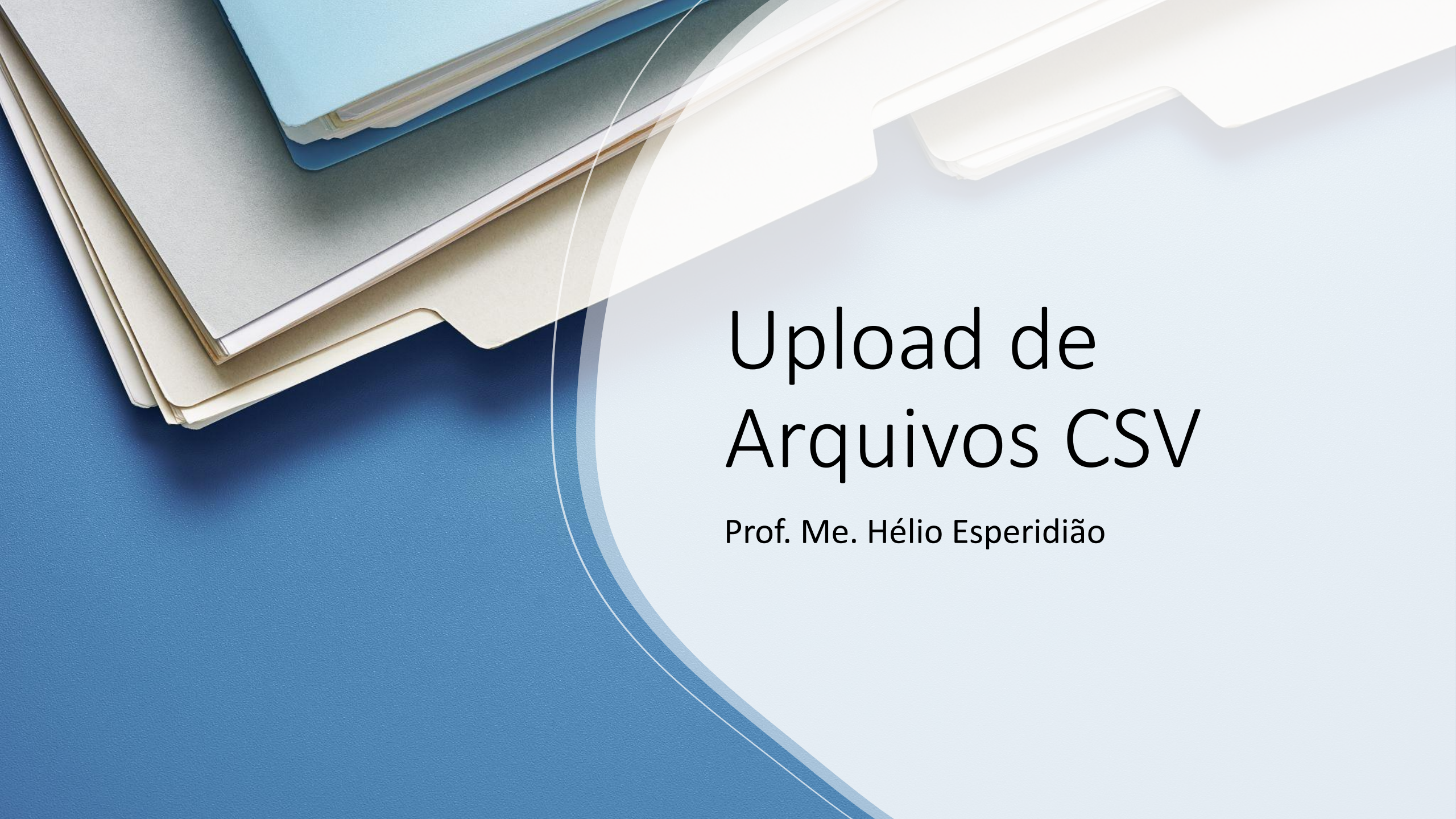




Upload de Arquivos CSV

Prof. Me. Hélio Esperidião

CSV - Character-separated values



Os arquivos CSV (do inglês "Character-separated values" ou "valores separados por um delimitador") servem para armazenar dados tabulares (números e texto) em texto simples.



O "texto simples" significa que o arquivo é uma sequência de caracteres puros, sem qualquer informação escondida que o computador tenha que processar.

Registros



- Um arquivo CSV abriga "registros", separados por quebras de linha (cada "registro" permanece numa linha do arquivo) e cada registro possui um ou mais "campos", separados por um delimitador, os mais comuns sendo a vírgula (","), o ponto e vírgula (";") e o caractere "invisível" que surge ao se pressionar a tecla "tab(\t)".

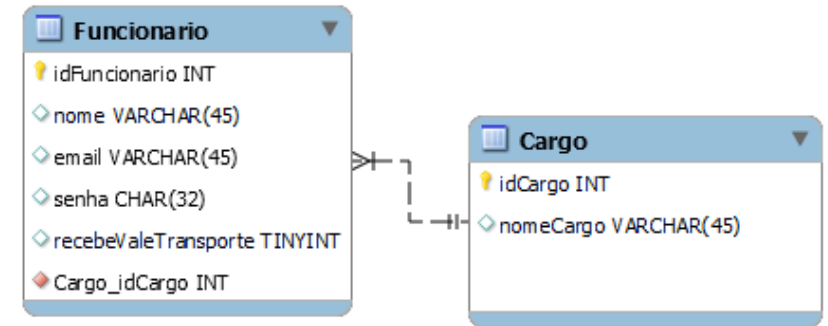
Exemplo: Estados Brasileiros

- IBGE;Estado;UF;Região;Qtd Mun;Sintaxe
- 11;Rondônia;RO;Região Norte;52;11'RO'
- 12;Acre;AC;Região Norte;22;12'AC'
- 13;Amazonas;AM;Região Norte;62;13'AM'

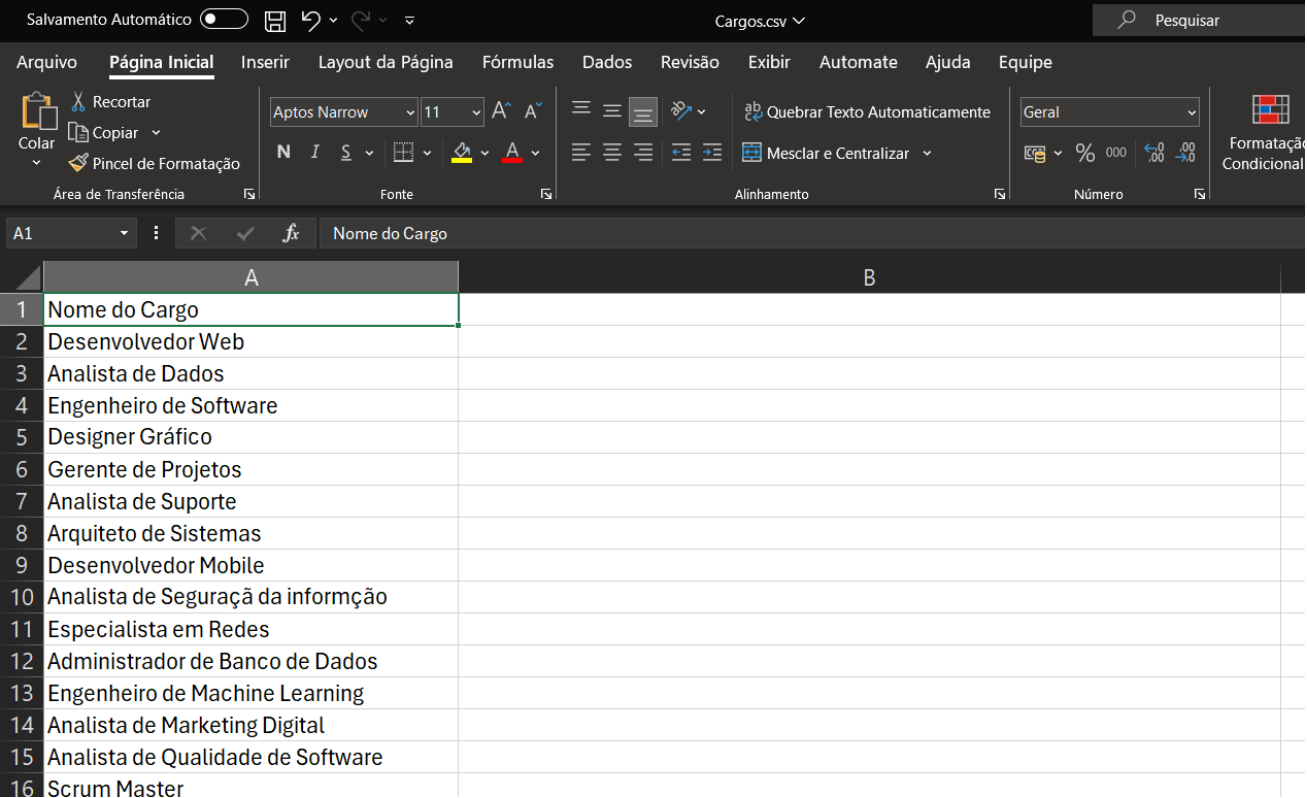
	A	B	C	D	E	F
	IBGE	Estado	UF	Região	Qtd Mun	Sintaxe
	11	Rondônia	RO	Região Norte	52	11'RO'
	12	Acre	AC	Região Norte	22	12'AC'
	13	Amazonas	AM	Região Norte	62	13'AM'
	14	Roraima	RR	Região Norte	15	14'RR'
	15	Pará	PA	Região Norte	144	15'PA'
	16	Amapá	AP	Região Norte	16	16'AP'
	17	Tocantins	TO	Região Norte	139	17'TO'
	21	Maranhão	MA	Região Nordeste	217	21'MA'
	22	Pernambuco	PE	Região Nordeste	224	22'PE'

Exemplo completo CSV

- Os dados serão gravados nas tabelas
- A rota /cargos/csv
 - Recebe um arquivo csv contendo apenas os cargos.
- A rota /funcionarios/csv
 - Recebe um arquivo com os dados dos funcionários incluindo os cargos.



Cargos.csv

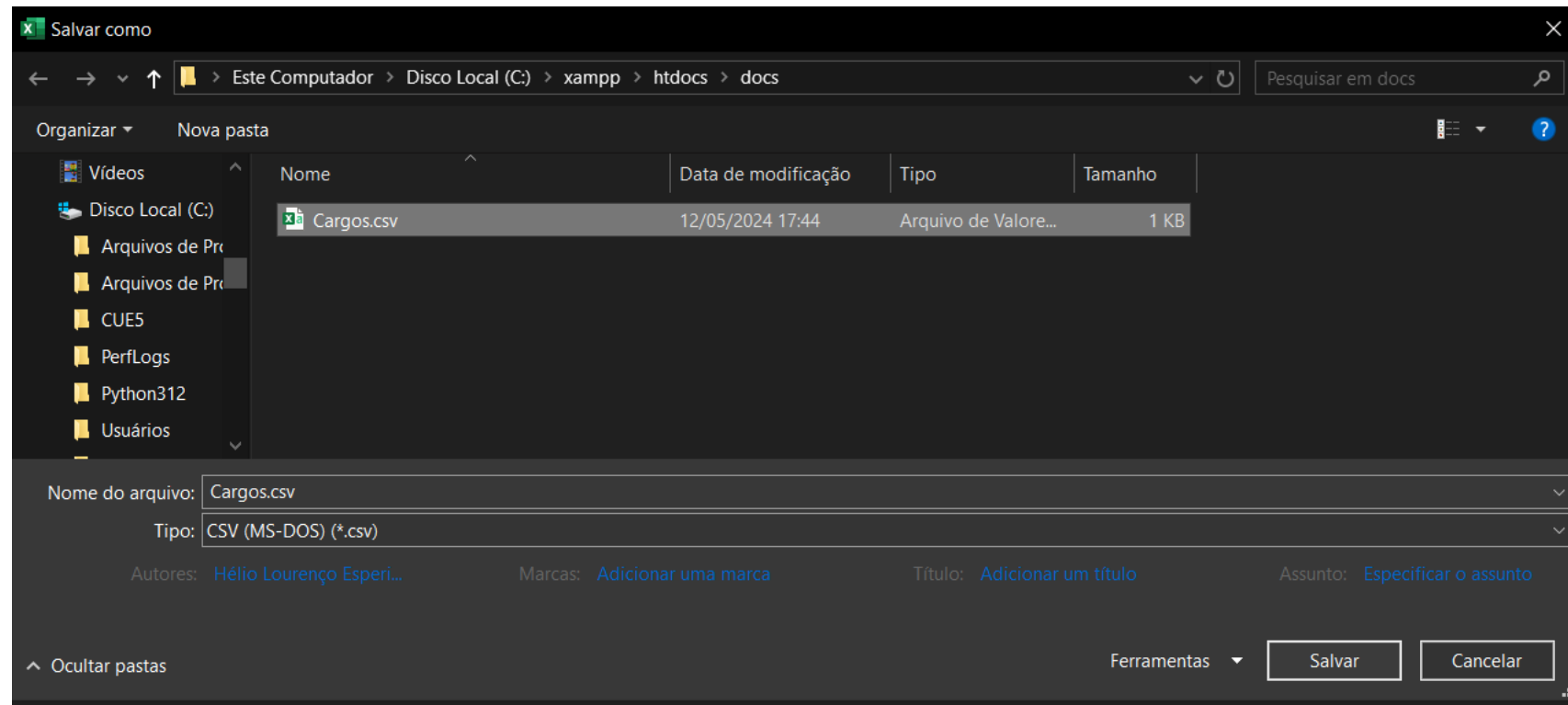


The screenshot shows the Microsoft Excel interface with the 'Cargos.csv' file open. The ribbon is set to 'Página Inicial' (Home). The formula bar shows 'Nome do Cargo' in cell A1. The spreadsheet contains a list of 16 job titles in column A, with column B being empty.

	A	B
1	Nome do Cargo	
2	Desenvolvedor Web	
3	Analista de Dados	
4	Engenheiro de Software	
5	Designer Gráfico	
6	Gerente de Projetos	
7	Analista de Suporte	
8	Arquiteto de Sistemas	
9	Desenvolvedor Mobile	
10	Analista de Segurança da informação	
11	Especialista em Redes	
12	Administrador de Banco de Dados	
13	Engenheiro de Machine Learning	
14	Analista de Marketing Digital	
15	Analista de Qualidade de Software	
16	Scrum Master	

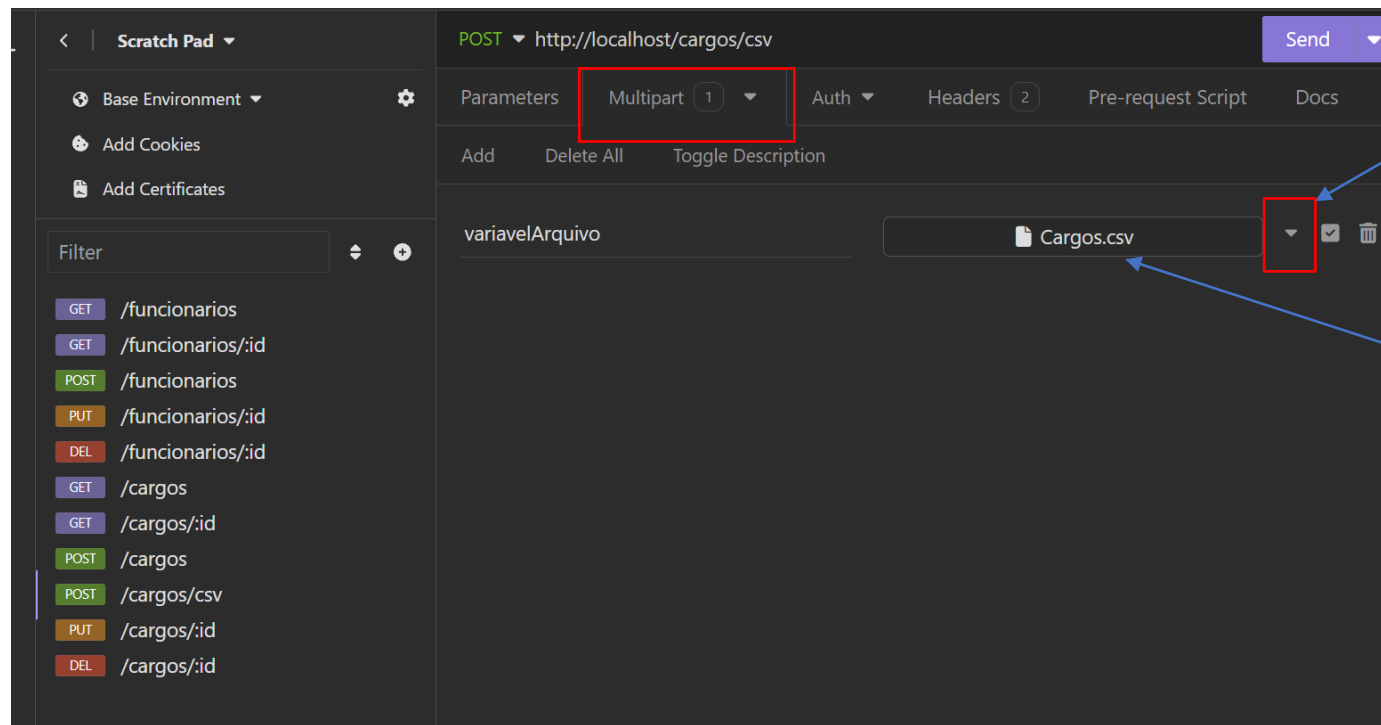
Como gerar?

- Salve uma tabela simples de linhas e colunas apenas.
- Salve Como o tipo CSV.



Como enviar um arquivo no insomnia:

Crie uma nova rota post



Trocar para Arquivo ou file

Selecione o arquivo

Crie uma rota para receber o arquivo

```
// Define uma rota para enviar um arquivo CSV para cadastrar todos os cargos
$roteador->post("/cargos/csv", function () {
    // Requer o arquivo de controle responsável por processar o arquivo CSV e
    cadastrar os cargos
    require_once ("controle/cargos/controle_cargos_create_from_csv.php");
});
```

Determine um controle

- controle_cargos_create_from_csv.php
 - Responsável por receber o arquivo e listar todos os cargos

controle_cargos_create_from_csv.php

```
<?php
// Obtém o nome temporário do arquivo CSV enviado pelo formulário HTML
$nomeArquivo = $_FILES["variavelArquivo"]["tmp_name"];

// Abre o arquivo CSV no modo de leitura
$ponteiroArquivo = fopen($nomeArquivo, "r");

// Loop que lê cada linha do arquivo CSV
// O número 1000 representa o tamanho máximo da linha no arquivo CSV em bytes
// Isso é usado para garantir que cada linha não exceda esse tamanho
while (($linhaArquivo = fgetcsv($ponteiroArquivo, 1000, ";")) !== false) {
    // Codifica cada elemento da linha para UTF-8 para garantir consistência na exibição
    $linhaArquivo = array_map("utf8_encode", $linhaArquivo);
    // Imprime os valores dos campos da linha atual do CSV
    // Cada valor é separado por vírgula e seguido de uma quebra de linha HTML
    echo $linhaArquivo[0] . "<br>";
}
?>
```

POST  http://localhost/cargos/csv

Send 

200 OK6.7 ms84 B

ParametersMultipart

1

Auth Headers

2

Pre-request ScriptDocs

Preview Headers

5

CookiesTimelineMock Re

AddDelete AllToggle Description

variavelArquivo

 Cargos.csv

Cargos
Analista de Sistemas
Técnico em informática
Programador PHP Jr

formularioUploadCSV.php

Formulário HTML

```
<html>
<head>
  <title>Upload de Arquivos com PHP</title>
</head>
<body>

  <form method="post" action="variavelArquivo.php" enctype="multipart/form-data">
    <label>Selecione o arquivo CSV</label>
    <input type="file" name="arquivo" /><br>
    <input type="submit" value="Enviar" />
  </form>

</body>
</html>
```

Modifique o controle

- Vamos modificar o controle, ao invés de apenas ler e apresentar os cargos do arquivo csv aproveitaremos a estrutura da classe cargo para inserir os cargos no banco de dados.

Utilizar a classe Cargo no controle

```
<?php
require_once ("modelo/Cargo.php");
// Obtém o nome temporário do arquivo CSV enviado pelo formulário HTML
$nomeArquivo = $_FILES["variavelArquivo"]["tmp_name"];

// Abre o arquivo CSV no modo de leitura
$ponteiroArquivo = fopen($nomeArquivo, "r");

// Loop que lê cada linha do arquivo CSV
$i = 0;    $cargos = array();
while (($linhaArquivo = fgetcsv($ponteiroArquivo, 1000, ";")) !== false) {
    // Converte os valores da linha para UTF-8, caso necessário
    $linhaArquivo = array_map("utf8_encode", $linhaArquivo);
    // Cria um novo objeto da classe Cargo

    $cargos[$i] = new Cargo();
    // Define o nome do cargo recebido da coluna zero do arquivo csv

    $cargos[$i]->setNomeCargo($linhaArquivo[0]);
    // Chama o método para criar o cargo no banco de dados

    if ($cargos[$i]->create() == true) { $i++;}
}

$resposta = new stdClass();
$resposta->status = true;
$resposta->msg = "Cargos cadastrados com sucesso";
$resposta->cadastrados = $cargos;
$resposta->totalCargos = $i;
echo json_encode($resposta);
?>
```

Resultado

POST http://localhost/cargos/csv Send 200 OK 5.96 ms 285 B Just Now

Parameters Multipart 1 Auth Headers 2 Pre-request Script Docs

Add Delete All Toggle Description

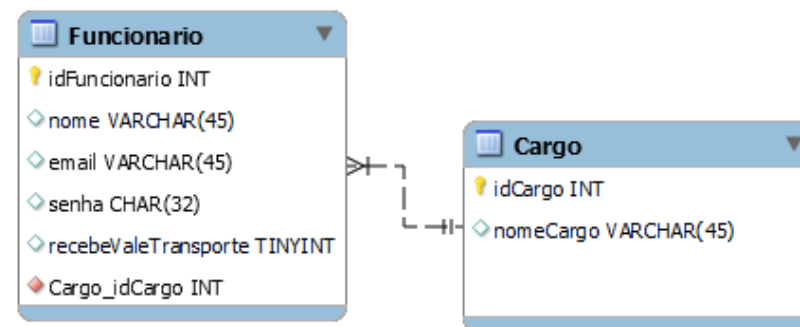
variavelArquivo Cargos.csv

```
1 {
2   "status": true,
3   "msg": "Cargos cadastrados com sucesso",
4   "cadastrados": [
5     {
6       "idCargo": 86,
7       "nomeCargo": "Cargos"
8     },
9     {
10      "idCargo": 87,
11      "nomeCargo": "Analista de Sistemas"
12    },
13    {
14      "idCargo": 88,
15      "nomeCargo": "Técnico em informática"
16    },
17    {
18      "idCargo": 89,
19      "nomeCargo": "Programador PHP Jr"
20    }
21  ],
22   "totalCargos": 4
23 }
```

E os funcionários?

- Observe que no arquivo csv temos o nome do cargo e o modelo de banco de dados apresenta o campo Cargo_idCargo como uma chave estrangeira de cargo. O que fazer?

Analista de Marketing					
	A	B	C	D	E
1	nome	email	senha	recebeValeTransporte	cargo
2	Lucas Oliveira	lucas.oliveira@example.com		Sim	Analista de Marketing
3	Isabela Santos	isabela.santos@example.com		Sim	Desenvolvedora Frontend
4	Pedro Almeida	pedro.almeida@example.com		Não	Gerente de Vendas
5	Carolina Lima	carolina.lima@example.com		Sim	Designer Gráfico
6	Gabriel Silva	gabriel.silva@example.com		Não	Analista de Recursos Humanos
7	Ana Souza	ana.souza@example.com		Não	Coordenadora de Projetos
8	Mateus Pereira	mateus.pereira@example.com		Sim	Analista de Qualidade
9	Mariana Costa	mariana.costa@example.com		Sim	Analista Financeiro
10	Rafael Rodrigues	rafael.rodrigues@example.com		Sim	Desenvolvedor Full Stack
11	Juliana Ferreira	juliana.ferreira@example.com		Não	Especialista em Suporte Técnico
12					



SQL – O que isso faz?

INSERT

```
into Funcionario (nomeFuncionario, email, senha,  
recebeValeTransporte, Cargo_idCargo)  
VALUES ("Hélio", "helioesperidiao@gmail.com", "", 1,  
(SELECT idCargo FROM cargo WHERE nomeCargo = ? ));
```

Rota

```
// Define uma rota para enviar um arquivo CSV para cadastrar todos os cargos
$roteador->post("/funcionarios/csv", function () {
    require_once ("controle/funcionario/controle_funcionario_create_from_csv.php");
});
```

controle_funcionario_create_from_csv.php 1/3

```
require_once ("modelo/Cargo.php");
require_once ("modelo/Funcionario.php");

//recupera o arquivo que foi enviado pelo cliente
$nomeArquivo = $_FILES["variavelArquivo"]["tmp_name"];
//abre o arquivo enviado pelo cliente
$ponteiroArquivo = fopen($nomeArquivo, "r");

$qtdeCargos = 0;
$qtdeFuncionarios = 0;
$cargos = array();
$funcionarios = array();
```

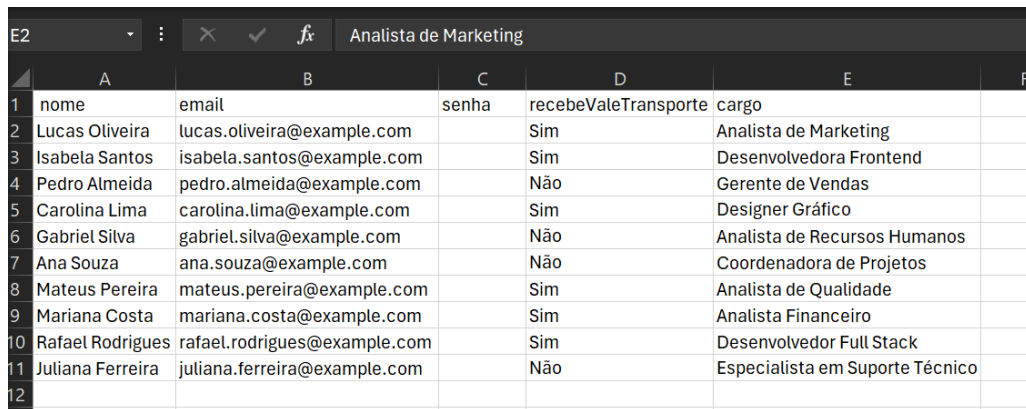
```
while (($linhaArquivo = fgetcsv($ponteiroArquivo, 1000, ";")) !== false) {  
    $linhaArquivo = array_map("utf8_encode", $linhaArquivo);  
    $cargoExistente = false;  
    foreach ($cargos as $cargo) { // Verifica se o cargo já existe no vetor $cargos  
        if ($cargo->getNomeCargo() == $linhaArquivo[4]) {  
            $cargoExistente = true;  
        }  
    }  
    if ($cargoExistente == false) { // Se o cargo não existir, cria um novo objeto Cargo e adiciona ao vetor $cargos  
        $cargos[$qtdCargos] = new Cargo();  
        $cargos[$qtdCargos]->setNomeCargo($linhaArquivo[4]);  
        if ($cargos[$qtdCargos]->create() == true) {$qtdCargos++;}  
    }  
    $funcionarios[$qtdFuncionarios] = new Funcionario();  
    $funcionarios[$qtdFuncionarios]->setNomeFuncionario($linhaArquivo[0]);  
    $funcionarios[$qtdFuncionarios]->setEmail($linhaArquivo[1]);  
    $funcionarios[$qtdFuncionarios]->setSenha($linhaArquivo[2]);  
    $funcionarios[$qtdFuncionarios]->setRecebeValeTransporte(($linhaArquivo[3] == "Sim" || "sim" ) ? 1 : 0);  
    $funcionarios[$qtdFuncionarios]->getCargo()->setNomeCargo($linhaArquivo[4]);  
    if ($funcionarios[$qtdFuncionarios]->createFromCSV() == true) {  
        $qtdFuncionarios++;  
    }  
}
```

controle_funcionario_create_from_csv.php 3/3

```
$resposta = new stdClass();  
$resposta->status = true;  
$resposta->msg = "Cadastrados com sucesso";  
$resposta->cargosCadastrados = $cargos;  
$resposta->totalCargos = $qtdCargos;  
$resposta->totalFuncionarios = $qtdFuncionarios;  
$resposta->FuncionariosCadastrados = $funcionarios;  
echo json_encode($resposta);
```

Método: createFromCSV()

- É preciso modificar a classe Funcionario e criar um novo método
- O método é exclusivo para cadastrar dados que vem do arquivo csv.
- Observe que no arquivo não há o campo idCargo.
- Então é necessária uma manipulação especial dos dados.



	A	B	C	D	E	F
1	nome	email	senha	recebeValeTransporte	cargo	
2	Lucas Oliveira	lucas.oliveira@example.com		Sim	Analista de Marketing	
3	Isabela Santos	isabela.santos@example.com		Sim	Desenvolvedora Frontend	
4	Pedro Almeida	pedro.almeida@example.com		Não	Gerente de Vendas	
5	Carolina Lima	carolina.lima@example.com		Sim	Designer Gráfico	
6	Gabriel Silva	gabriel.silva@example.com		Não	Analista de Recursos Humanos	
7	Ana Souza	ana.souza@example.com		Não	Coordenadora de Projetos	
8	Mateus Pereira	mateus.pereira@example.com		Sim	Analista de Qualidade	
9	Mariana Costa	mariana.costa@example.com		Sim	Analista Financeiro	
10	Rafael Rodrigues	rafael.rodrigues@example.com		Sim	Desenvolvedor Full Stack	
11	Juliana Ferreira	juliana.ferreira@example.com		Não	Especialista em Suporte Técnico	
12						

```
public function createFromCSV(){
    $conexao = Banco::getConexao(); // Obtém a conexão com o banco de dados
    // Define a consulta SQL para inserir um novo funcionário
    $SQL = "INSERT into Funcionario
    (nomeFuncionario, email, senha, recebeValeTransporte,Cargo_idCargo)
    VALUES(?,?,?,?(SELECT idCargo FROM cargo WHERE nomeCargo = ? ))";
    $prepararSQL = $conexao->prepare($SQL); // Prepara a consulta
    $nomeCargo = $this->cargo->getNomeCargo(); // Obtém o nome do cargo do funcionário
    $prepararSQL->bind_param("sssis",$this->nomeFuncionario,$this->email,$this->senha,$this->recebeValeTransporte,$nomeCargo) ;
    // Executa a consulta
    $executar = $prepararSQL->execute();
    // Obtém o ID do funcionário cadastrado
    $idCadastrado = $conexao->insert_id;
    // Define o ID do funcionário na instância atual da classe
    $this->setIdFuncionario($idCadastrado);
    // Fecha a consulta
    $prepararSQL->close();
    // Retorna se a operação foi executada com sucesso
    return $executar;
}
```

Método – Classe Funcionário

POST http://localhost/funcionarios/csv

Send

200 OK34.5 ms2.5 KB4 Minutes Ago

ParametersMultipart1AuthHeaders2Pre-request ScriptDocs

AddDelete AllToggle Description

variavelArquivo

Funcionarios.csv

Preview

Headers5CookiesTimelineMock Response

```
41 {
42   "idCargo": 84,
43   "nomeCargo": "Desenvolvedor Full Stack"
44 },
45 {
46   "idCargo": 85,
47   "nomeCargo": "Especialista em Suporte Técnico"
48 }
49 ],
50 "totalCargos": 11,
51 "totalFuncionarios": 11,
52 "FuncionariosCadastrados": [
53   {
54     "idFuncionario": 28,
55     "nomeFuncionario": "nome",
56     "email": "email",
57     "recebeValeTransporte": 1,
58     "idCargo": null,
59     "nomeCargo": "cargo"
60   },
```