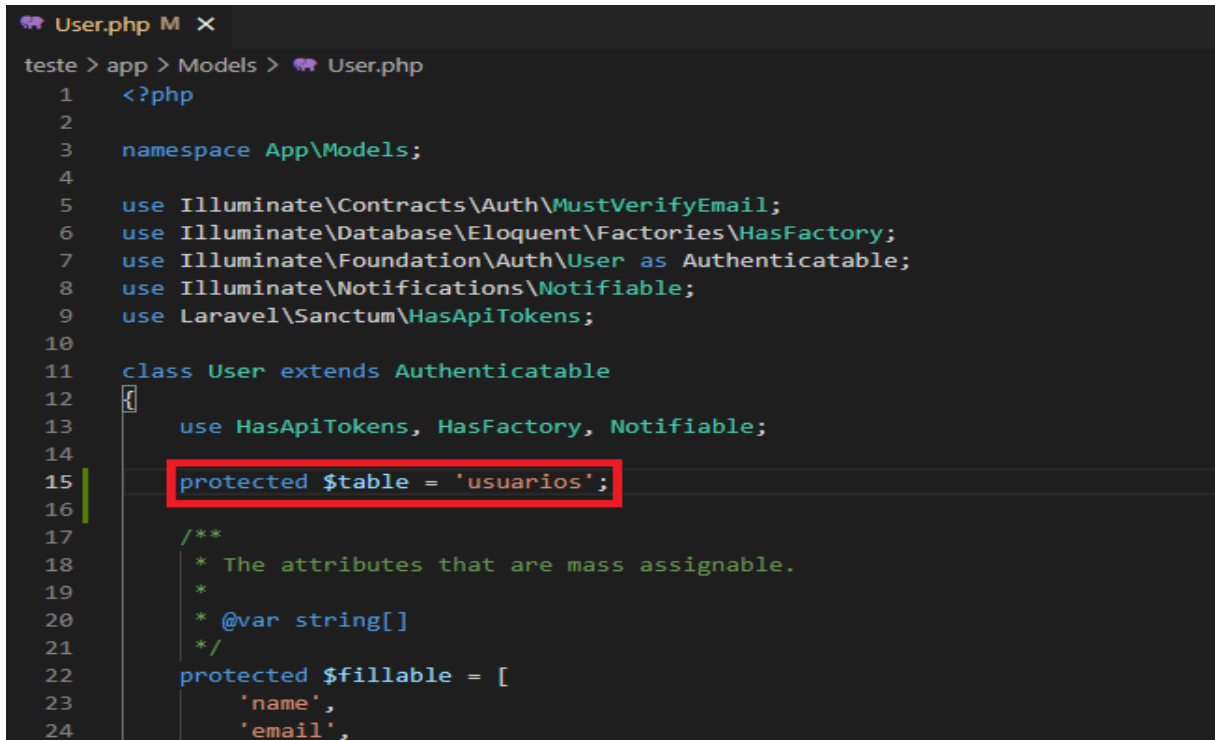


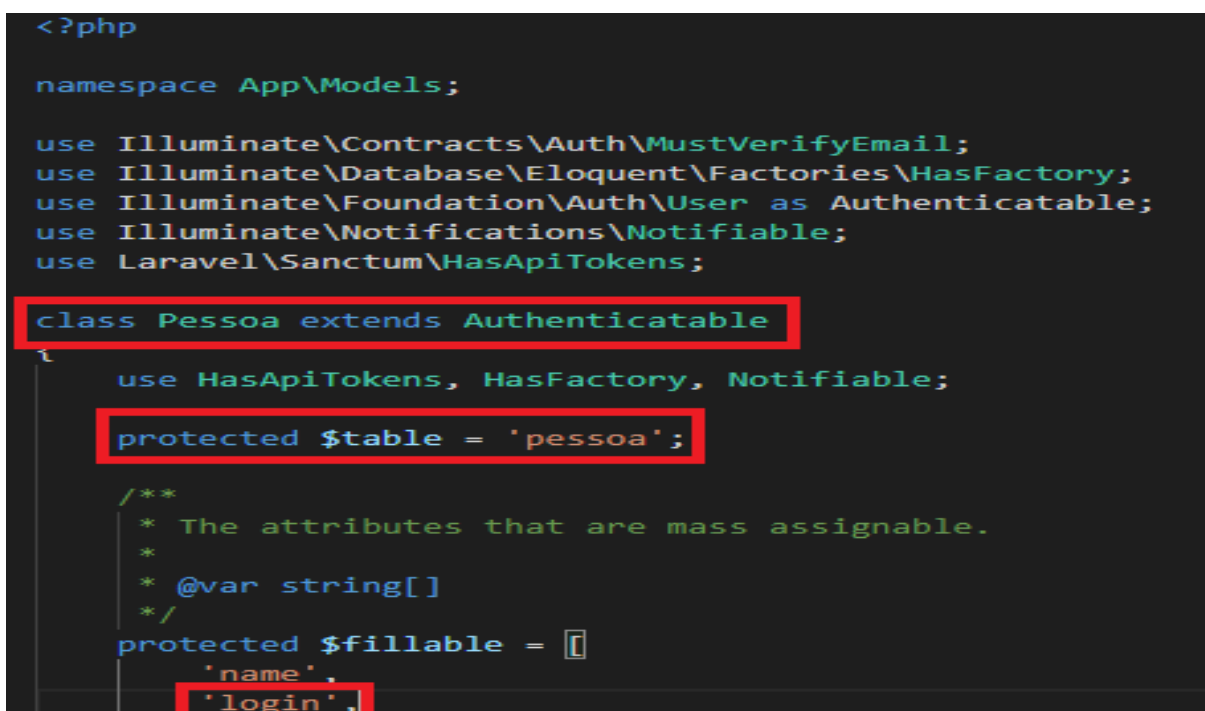
Mudando autenticação de usuário do Laravel:

1 - Caso seja uma tabela com o nome diferente de “Users” e possua os mesmos campos, pode-se apenas explicitar o nome da tabela conforme imagem abaixo:



```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9  use Laravel\Sanctum\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14
15     protected $table = 'usuarios';
16
17     /**
18      * The attributes that are mass assignable.
19      *
20      * @var string[]
21      */
22     protected $fillable = [
23         'name',
24         'email',
```

2 - Caso decida por utilizar uma nova classe ou uma nova coluna no lugar de e-mail, é necessário que sua classe instancie a classe “Authenticatable”, ajuste o arquivo “config/auth.php” e sobrescrever o método no controller conforme as imagens abaixo:



```
<?php

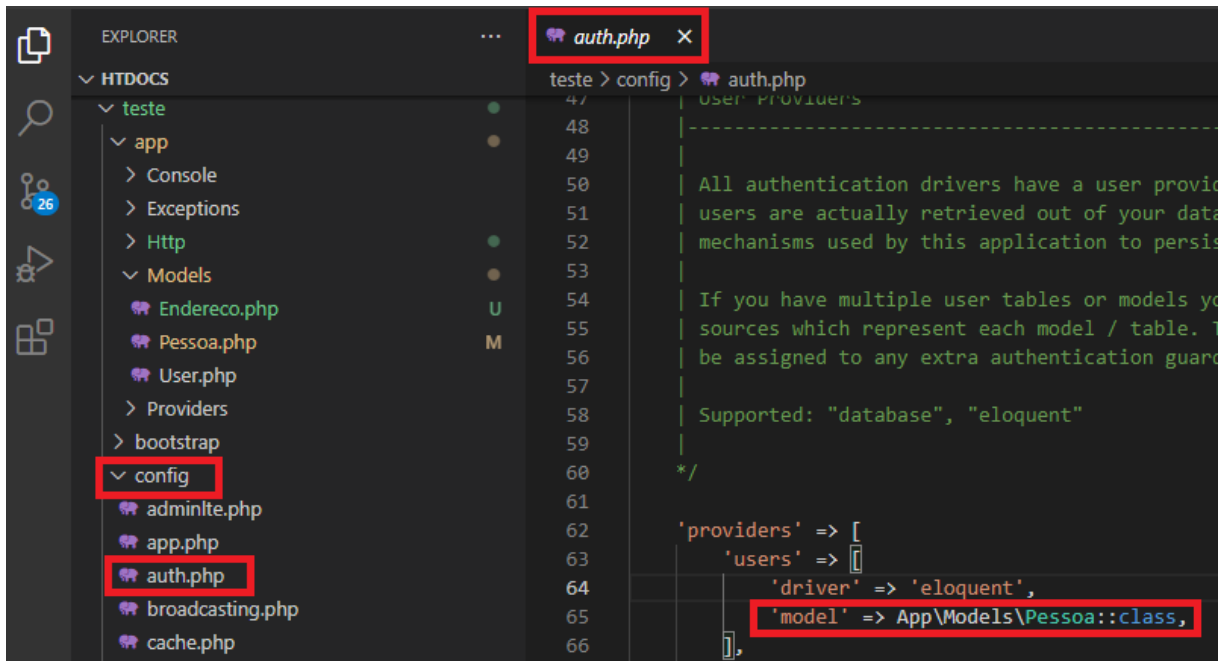
namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class Pessoa extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    protected $table = 'pessoa';

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [
        'name',
        'login',
```



```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Providers\RouteServiceProvider;
use Illuminate\Foundation\Auth\AuthenticatesUsers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;
use App\Models\Pessoa;

class LoginController extends Controller
{
    use AuthenticatesUsers;

    protected function username(){
        return "login";
    }
}
```

3 - Caso seja necessário acrescentar mais campos para que seja realizado a autenticação do usuário, pode-se fazer a autenticação manual criando do método “authenticate” conforme imagem abaixo:

```
public function authenticate(Request $request)
{
    if (Auth::attempt([ 'login' => $request->login,
                        'password' => $request->password,
                        'acessa_sistema' => 'S',
                        'ativo' => 'S'
                    ])) {

        $request->session()->regenerate();
        return redirect()->intended('home');

    }

    return back()->withErrors([
        'login' => 'The provided credentials do not match our records.',
    ]);
}
```

Detalhe: Em alguns caso quando há quebra de linha existindo apenas o if é necessário colocar o else ou então deixar tudo numa linha só.

```
public function authenticate(Request $request)
{
    if (Auth::attempt([ 'email' => $request->email,
                        'password' => $request->password,
                        'acessa_sistema' => 'S',
                        'ativo' => 'S'
                    ])){

        $request->session()->regenerate();
        return redirect()->intended('home');

    }
    else{
        return back()->withErrors([
            'login' => 'The provided credentials do not match our records.',
        ]);
    }
}
```

É necessário que seja explicitado as rotas também:

```
Route::get('/login', [ LoginController::class, 'index' ]->name('login'));
Route::post('/login', [ LoginController::class, 'authenticate' ]->name('autenticar'));
Route::get('/logout', [ LoginController::class, 'logout' ]->name('logout'));
```

Fonte: <https://laravel.com/docs/8.x/authentication>