

02

Enviando emails

O sistema está bonito mas gostaria de notificar o vendedor do interesse do comprador, isto é, enviar um email para nosso vendedor.

Primeiro, logo após salvar nossa venda no banco de dados, desejamos carregar a biblioteca de email:

```
public function nova() {
    $usuario = autoriza();

    $this->load->model(array("vendas_model", "produtos_model", "usuarios_model"));
    $venda = array(
        "produto_id" => $this->input->post("produto_id"),
        "comprador_id" => $usuario["id"],
        "data_de_entrega" =>
            dataPtBrParaMysql($this->input->post("data_de_entrega"))
    );
    $this->vendas_model->salva($venda);

    $this->load->library("email");
}
```

E agora configuramos a conta de email que enviará os mesmos:

```
$config["protocol"] = "smtp";
$config["smtp_host"] = "ssl://smtp.gmail.com";
$config["smtp_user"] = "codeigniteralura@gmail.com";
$config["smtp_pass"] = "123456";
$config["charset"] = "utf-8";
$config["mailtype"] = "html";
$config["newline"] = "\r\n";
$config["smtp_port"] = '465';
$this->email->initialize($config);
```

Lembre-se de usar as configurações de seu servidor de email. No nosso caso usamos uma conta do gmail. Vou dizer quem é o meu from:

```
$this->email->from("codeigniteralura@gmail.com", "Mercado");
```

Direi também para quem vai o email (pode ser uma array), no nosso caso o vendedor:

```
$this->email->to(???email_do_vendedor???);
```

Não tenho o email dele, somente o id do produto e o id do comprador ainda, teremos que buscar em breve. Configuramos também o subject do email e o corpo, finalmente enviando ele:

```
$this->email->subject("Seu produto {$produto['nome']} foi vendido!");
$this->email->message("<html>Seu produto {$produto['nome']} foi vendido!</html>");
$this->email->send();
```

Também não temos o nome do produto ainda, precisamos buscá-lo. Buscar produto é fácil, já existe no `produtos_model` o método busca:

```
$produto = $this->produtos_model->busca($venda["produto_id"]);
```

Mas ainda não temos como buscar vendedor por id, teremos que fazer isso:

```
$vendedor = $this->usuarios_model->busca($produto["usuario_id"]);
```

Sem esquecer de carregar tais modelos, claro.

```
$this->load->model(array("vendas_model", "produtos_model", "usuarios_model"));
```

E implementar a busca de usuário por id:

```
public function busca($id) {
    $this->db->where("id", $id);
    return $this->db->get("usuarios")->row_array();
}
```

O método nova fica grande mas agora capaz de enviar emails:

```
public function nova() {
    $usuario = autoriza();

    $this->load->model(array("vendas_model", "produtos_model", "usuarios_model"));
    $venda = array(
        "produto_id" => $this->input->post("produto_id"),
        "comprador_id" => $usuario["id"],
        "data_de_entrega" =>
            dataPtBrParaMysql($this->input->post("data_de_entrega"))
    );
    $this->vendas_model->salva($venda);

    $this->load->library("email");
    $config["protocol"] = "smtp";
    $config["smtp_host"] = "ssl://smtp.gmail.com";
    $config["smtp_user"] = "codeigniteralura@gmail.com";
    $config["smtp_pass"] = "123456";
    $config["charset"] = "utf-8";
    $config["mailtype"] = "html";
    $config["newline"] = "\r\n";
    $config["smtp_port"] = '465';
    $this->email->initialize($config);
```

```

$produto = $this->produtos_model->busca($venda["produto_id"]);
$vendedor = $this->usuarios_model->busca($produto["usuario_id"]);

$this->email->from("codeigniteralura@gmail.com", "Mercado");
$this->email->to($vendedor["email"]);
$this->email->subject("Seu produto {$produto['nome']} foi vendido!");
$this->email->message("<html>Seu produto {$produto['nome']} foi vendido!</html>");
$this->email->send();

$this->session->set_flashdata("success", "Pedido de compra efetuado com sucesso");
redirect("/");
}

```

O email chega em nossa caixa de entrada, maravilha. Mas ainda existe algo de estranho: o código html do email está escrito dentro de nosso controller, algo muito ruim. Seria legal extrair isso em um arquivo separado, algo como uma view html. Então dentro da minha view, dentro de vendas posso criar um arquivo chamado `email.php`:

```

<html>
<h1>Venda no mercado</h1>
Seu produto <b><?= $produto['nome'] ?></b> foi vendido!
</html>

```

Como faço para carregar o conteúdo desta view dentro do meu email?

```

$conteudo = ???;
$this->email->message($conteudo);

```

Basta renderizarmos a view:

```

$dados = array("produto" => $produto);
$conteudo = $this->load->view("vendas/email.php", $dados);

```

Mas ao invés de redirecionarmos, queremos o resultado na variável, então passamos um terceiro parâmetro para nossa view:

```

$dados = array("produto" => $produto);
$conteudo = $this->load->view("vendas/email.php", $dados, TRUE);

```

O `TRUE` indica que não queremos renderizar a página para o usuário que está navegando no site, mas simplesmente rodar a template engine e atribuir a variável `conteudo` o resultado dela.

Dessa maneira conseguimos enviar por email o conteúdo de um arquivo que é um template como uma view qualquer.

