

02

## Criando o MailSender

Se você quiser fazer o download do que foi desenvolvido até o capítulo anterior, [clique aqui](https://github.com/alura-cursos/java-ee-webapp-3/archive/0ab37de16c27c40cf0f3b418f6281a0a288c14d5.zip) (<https://github.com/alura-cursos/java-ee-webapp-3/archive/0ab37de16c27c40cf0f3b418f6281a0a288c14d5.zip>).

Crie a classe `MailSender`, no pacote `br.com.casadocodigo.loja.infra`. Esta classe irá conter o código de envio de e-mail. Essa classe deve conter um método chamado `send()`, que recebe o remetente, o destinatário, o assunto, e o corpo do e-mail. O escopo da classe deve ser de aplicação:

```
@ApplicationScoped
public class MailSender {

    public void send(String from, String to, String subject, String body) {

    }
}
```

Dentro do método `send()`, crie um `MimeMessage` que recebe um `Session` no construtor. Aproveite para criar o `Session` como atributo da classe:

```
@ApplicationScoped
public class MailSender {

    private Session session;

    public void send(String from, String to, String subject, String body) {

        MimeMessage message = new MimeMessage(session);
    }
}
```

Iremos receber o `Session` como um recurso da aplicação. Utilize a anotação `@Resource`, passando o nome da JNDI que será associada ao recurso:

```
@Resource(mappedName = "java:jboss/mail/gmail")
private Session session;
```

Agora chame os métodos referentes à construção do e-mail:

```
public void send(String from, String to, String subject, String body) {

    MimeMessage message = new MimeMessage(session);

    try {
        message.setRecipients(javax.mail.Message.RecipientType.TO, InternetAddress.parse(to));
        message.setFrom(new InternetAddress(from));
        message.setSubject(subject);
        message.setContent(body, "text/html");
    }
}
```

```
        Transport.send(message);
    } catch (MessagingException e) {
        throw new RuntimeException(e);
    }
}
```

Na classe `PagamentoService` , no método `pagar()` , dentro do `executor` , logo após a linha que constrói a resposta:

```
Response response = Response.seeOther(responseUri).build();
```

chame o método `send()` da classe `MailSender` . Defina um atributo do tipo `MailSender` na classe `PagamentoService` e receba esse `MailSender` via injeção de dependências.