

Envio de e-mail assíncrono

Crie a configuração para que seja possível enviar a mensagem para o destino. No pacote `br.com.casadocodigo.loja.conf` crie a classe `ConfigureJMSDestination`:

```
@JMSDestinationDefinitions({
    @JMSDestinationDefinition(
        name = "java:/jms/topics/CarrinhoComprasTopico",
        interfaceName = "javax.jms.Topic",
        destinationName = "java:/jms/topics/CarrinhoComprasTopico"
    )
})
@Singleton
public class ConfigureJMSDestination {
```

Na classe `PagamentoService`, não precisamos mais receber o `MailSender` via injeção de dependências. Agora receba um `JMSContext` e um `Destination`.

```
@Path("/pagamento")
public class PagamentoService {

    // ...

    @Inject
    private JMSContext jmsContext;

    @Resource(name = "java:/jms/topics/CarrinhoComprasTopico")
    private Destination destination;

    // método pagar()
}
```

No método `pagar()`, fora da chamada ao `executor.submit()`, crie o `JMSProducer`:

```
@POST
public void pagar(@Suspended final AsyncResponse ar, @QueryParam("uuid") String uuid) {

    // ...

    JMSProducer producer = jmsContext.createProducer();

    executor.submit(() -> {

        // ...
    });
}
```

Dentro do `executor.submit()`, após a chamada ao `pagamentoGateway.pagar()`, chame o método `send()` do `producer`, passando o destino e, o `uuid` da compra como mensagem.

```
executor.submit(() -> {
    try {
        String resposta = pagamentoGateway.pagar(compra.getTotal());

        // chamada ao método send() aqui

        // ...
    });
}
```

Tente finalizar uma compra. Tudo já deve estar funcionando!