

Instituto Nacional de Telecomunicações – INATEL

**Curso de Pós-graduação em Desenvolvimento de Aplicações para Dispositivos
Móveis e *Cloud Computing***

**DM112 - Conceituação em *Cloud Computing*, Programação OO, *Web Services* e
SOA**

Higor Augusto Silvério

Rodrigo Ribeiro Prado

Trabalho Final da Disciplina

Análise, Modelagem e Projeto Orientados a Serviços

Santa Rita do Sapucaí - MG

2020

Higor Augusto Silvério

Rodrigo Ribeiro Prado

Trabalho Final da Disciplina

Análise, Modelagem e Projeto Orientados a Serviços

Projeto de levantamento dos requisitos de análise, modelagem e projeto, propriamente dito, orientados a serviços para o Provedor de Serviços de Logística do projeto do curso Pós-graduação em Desenvolvimento de Aplicações para Dispositivos Móveis e *Cloud Computing*.

Orientador: Professor MSc. Roberto Ribeiro Rocha

Santa Rita do Sapucaí – MG

2020

Lista de ilustrações

Figura 1 - Projeto do curso	6
Figura 2 - Visualização do Provedor de Serviços de Logística.....	6
Figura 3 - Diagrama de caso	8
Figura 4 - Modelagem do processo de negócio	9
Figura 5 - Detalhamento (decomposição) do processo de negócio	10
Figura 6 - Identificação das operações candidatas	10
Figura 7 - UML da relação refinada dos serviços	12
Figura 8 - Modelo final de relacionamento entre os serviços	13
Figura 9 - Versão detalhada dos serviços e suas operações.....	14

Sumário

Lista de ilustrações.....	3
Lista de abreviaturas e siglas	5
1. Introdução	6
2. Desenvolvimento	8
3. Conclusão	14
Referências	15

Lista de abreviaturas e siglas

CRUD – *Create, Read, Update and Delete* (Cria, Lê, Atualiza e Deleta).

MSc. – *Master of Science* (Mestre em Ciências).

OO – Orientada a objetos.

REST – *Representational State Transfer* (Transferência Representacional de Estado).

SOA – *Service-Oriented Architecture* (Arquitetura Orientada a Serviços).

SRM – *Supplier Relationship Management* (Gerenciamento de Relacionamento com o Fornecedor).

UML – *Unified Modeling Language* (Linguagem de Modelagem Unificada).

XML – *Extensible Markup Language* (Linguagem de Marcação Extensível).

1. Introdução

Objetivo: Gerar os passos de análise, modelagem e projeto orientados a serviços para o Provedor de Serviços de Logística do projeto do curso, sendo este exemplificado na imagem abaixo:

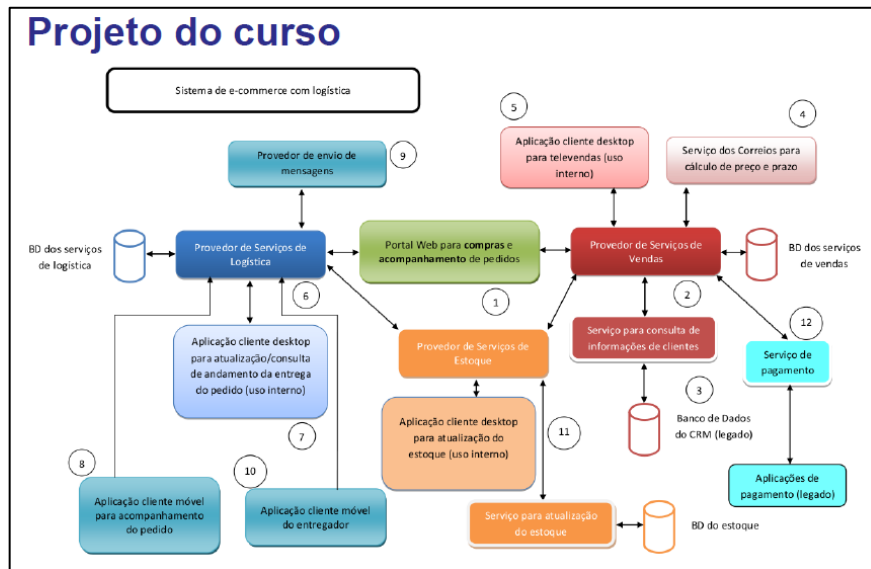


Figura 1 - Projeto do curso

O Provedor de Serviços de Logística, o qual deve ser utilizado para gerar o início do ciclo de vida de desenvolvimento SOA desta atividade, é mostrado em detalhe na imagem abaixo, juntamente às suas interações mais próximas:

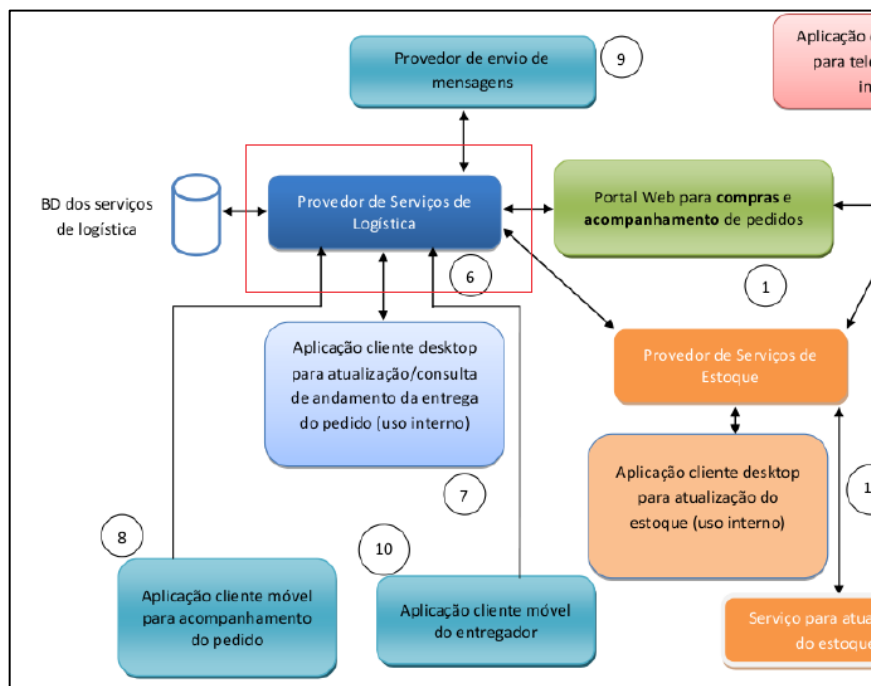


Figura 2 - Visualização do Provedor de Serviços de Logística

Observadas as premissas do trabalho, bem como as imagens de exemplo acima, são elaborados os passos para o início do ciclo de vida deste desenvolvimento SOA em três fases: análise, modelagem e projeto.

2. Desenvolvimento

1ª fase: Análise orientada a serviços (3 passos)

Passo 1. Definição dos requisitos de negócio – requisitos, fronteira de análise, partes envolvidas e partes afetadas da corporação:

- Requisitos:
 - Consultar a lista de pedidos a serem entregues;
 - Registrar a entrega de um pedido;
 - Enviar um e-mail para o cliente quando o pedido for entregue.
- Fronteiras de análise:
 - O entregador consulta a lista de pedidos a serem entregues;
 - O entregador registra a entrega de um pedido;
 - O sistema acessa o servidor de e-mails.
- Partes envolvidas:
 - Cliente;
 - O sistema da loja;
 - Entregador.
- Partes afetadas da corporação:
 - Estoque;
 - Pedido;
 - Logística.

Passo 2. Identificação de sistemas existentes (legados):

- Envio de e-mail.

Passo 3. Modelagem de serviços (candidatos):

Com as informações anteriores, o diagrama de caso de uso fica da seguinte forma:



Figura 3 - Diagrama de caso

Este caso de uso ilustra os dois atores e as funcionalidades que o sistema deve fornecer.

O modelo de processo de negócio do problema analisado possui a seguinte estrutura:

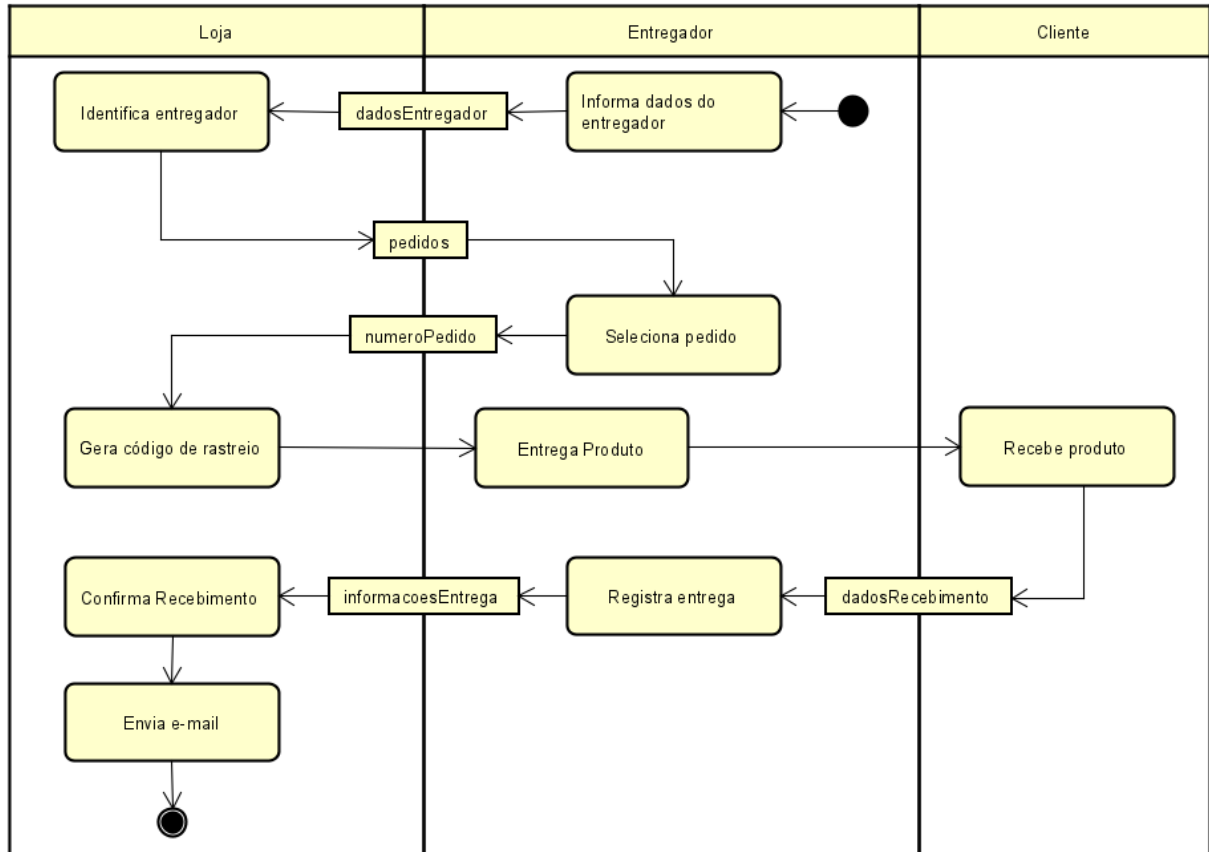


Figura 4 - Modelagem do processo de negócio

2ª fase: Modelagem de serviços candidatos (8 passos)

Passo 1. Decomposição (detalhamento) do processo de negócio:

A decomposição (detalhamento) do processo de negócio foi feita da seguinte forma:

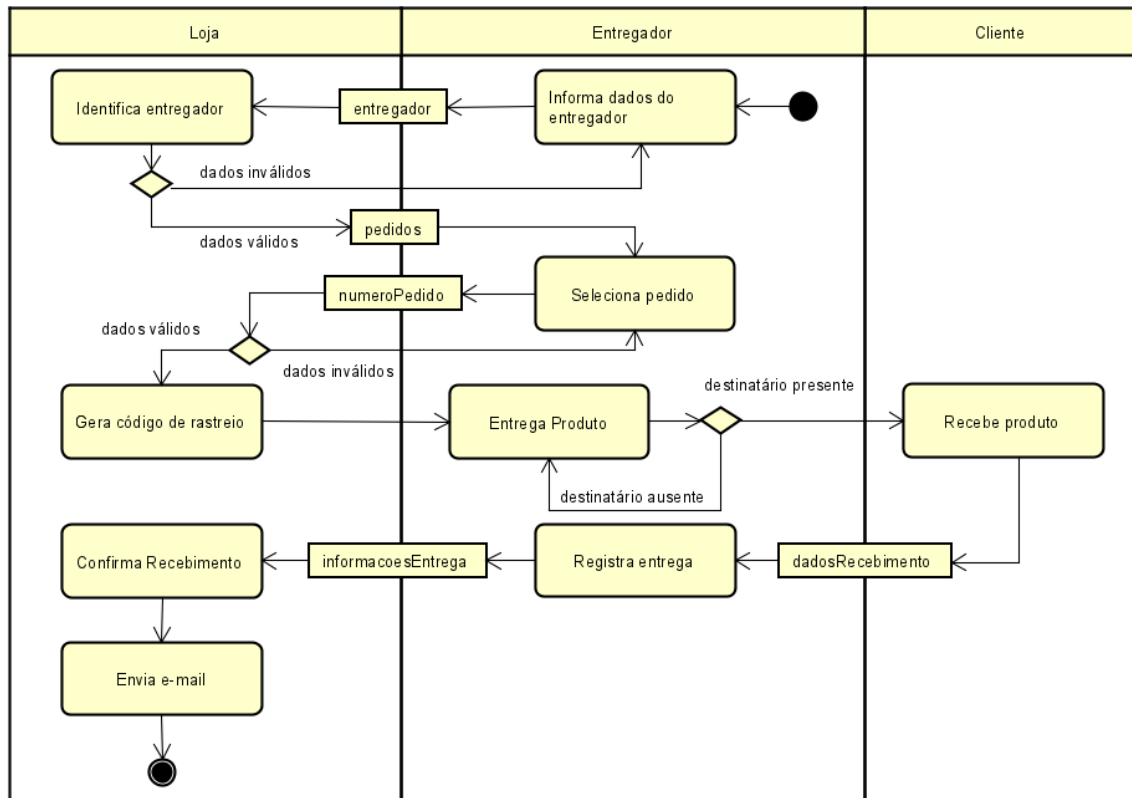


Figura 5 - Detalhamento (decomposição) do processo de negócio

Passo 2 e 3. Identificação de operações candidatas e abstração de lógica de orquestração:

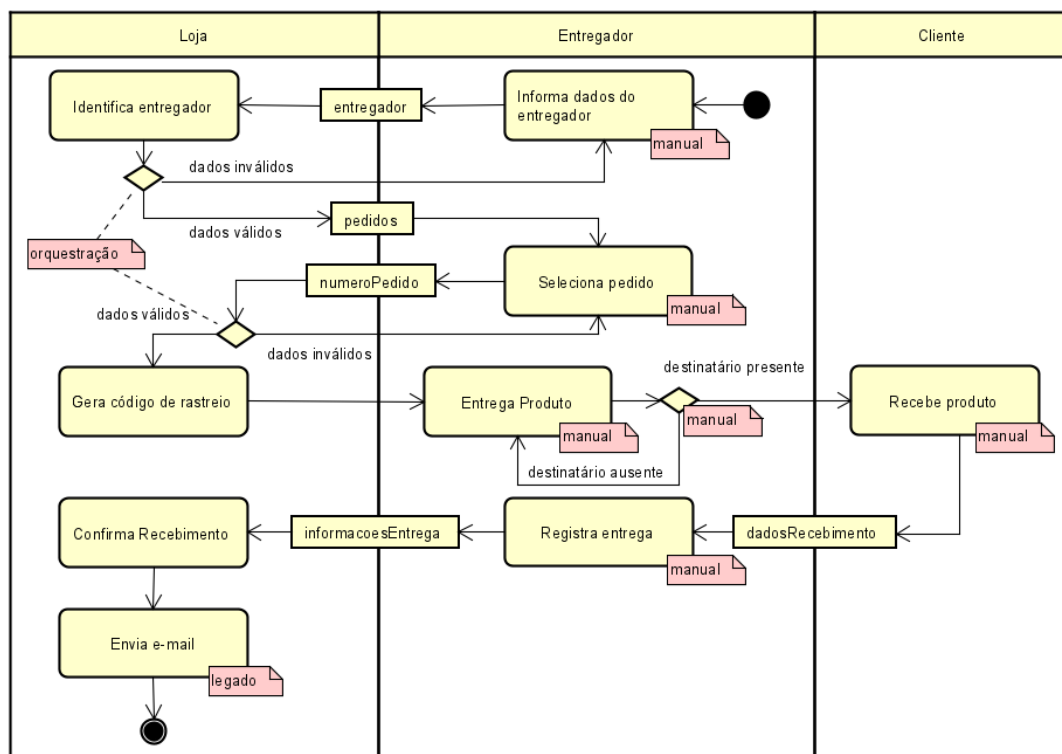


Figura 6 - Identificação das operações candidatas

Passo 4. Criação de serviços candidatos e agrupá-los em contextos lógicos:

Separação dos serviços identificados:

- SRM:
 - Identifica entregador.
- Pedido:
 - Busca e apresenta pedidos;
 - Registra a retirada do pedido.
- Entrega:
 - Gera código de rastreio;
 - Registra entrega efetuada;
 - Salvar informações da entrega realizada.
- Mensageiro:
 - Envia e-mail de confirmação de entrega (legado).

Passo 5. Aplicação de princípios de orientação a serviços – reusabilidade, autonomia, *statelessness* etc.:

- Pedido:
 - Consulta pedido (engloba “Busca e apresenta pedidos” e “Identifica entregador”);
 - Atualiza pedido.
- Entrega:
 - Gera código de rastreio;
 - Atualizar *status* da entrega
 - Salvar informações de entrega realizada
- Mensageiro:
 - Envia e-mail de confirmação de entrega (legado).

Passo 6. Identificação de composições de serviços:

Neste passo, criamos então o serviço de entrega, que irá orquestrar todo o processo de logística do pedido e organizá-lo, de modo a chamar os outros serviços, tomando decisões a partir de suas respostas.

Passo 7. Revisão dos agrupamentos de operações:

Um novo serviço no grupo de “Entrega” pode ser criado, chamado “Registra entrega” (engloba “Atualizar *status* da entrega” e “Salvar informações de entrega realizada”)

- Pedido:
 - Consulta pedido;
 - Atualiza pedido.
- Entrega:
 - Gera código de rastreio;
 - Registra entrega (engloba “Atualizar *status* da entrega” e “Salvar informações de entrega realizada”).
- Mensageiro:
 - Envia e-mail de confirmação de entrega (legado).

Passo 8. Análise dos requisitos de processamento:

Assim como feito no exercício em sala de aula, esse passo não será executado.

Por conseguinte, obtemos a seguinte relação refinada dos serviços, na representação de classes UML mostrada abaixo:

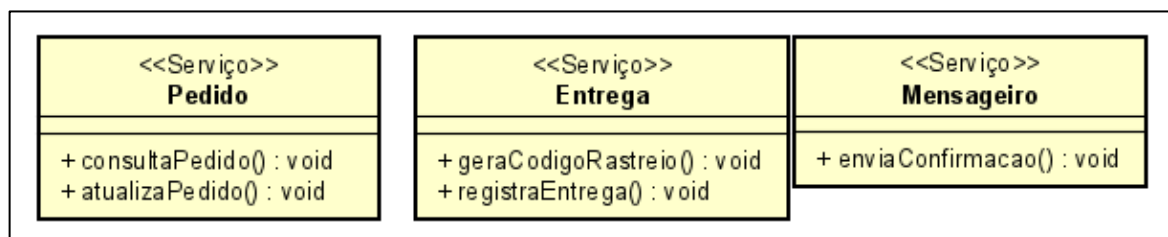


Figura 7 - UML da relação refinada dos serviços

3ª fase: Projeto orientado a serviços (5 passos)

Passo 1. Composição da arquitetura orientada a serviços – padrões, bibliotecas, recursos específicos de uma linguagem etc.:

Para a implementação dos serviços do Provedor de Serviços de Logística serão utilizadas as três camadas: *Task*, *Entity* e *Utility*.

Como padrões de implementação, assim como utilizado em exemplo de sala de aula, será utilizada a linguagem Java em conjunto com a biblioteca REST, que fornecem suporte para *web services*.

Serão utilizadas anotações da linguagem Java, para reduzir ao máximo as configurações com arquivos XML.

Passo 2. Projeto de serviços baseados em entidades – operações CRUD: Pedido.

Passo 3. Projeto de serviços de aplicação: Mensageiro.

Passo 4. Projeto de serviços baseados em tarefa: Entrega.

Passo 5. Projeto de processo orientado a serviços: Este é extraído do mapeamento, que indica onde cada serviço será inserido no processo de negócio, constituindo o seguinte modelo final de relacionamento entre os serviços e suas respectivas camadas:

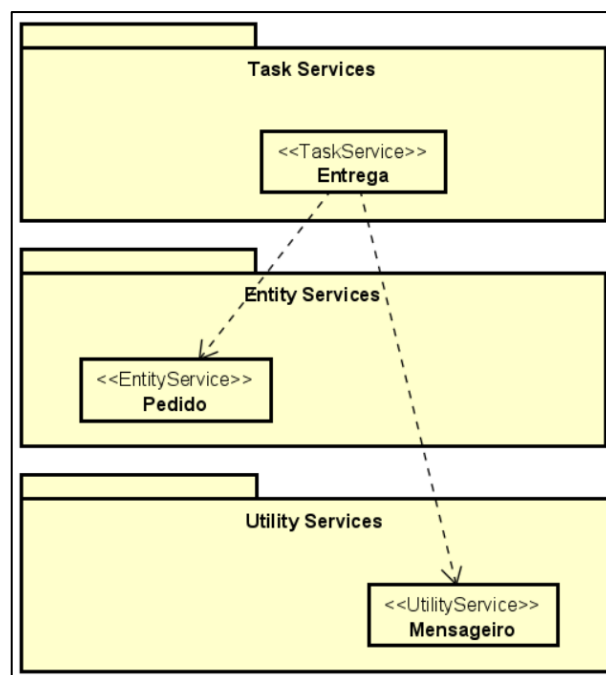


Figura 8 - Modelo final de relacionamento entre os serviços

E uma versão mais detalhada dos serviços é mostrada na figura abaixo, pronta para implementar:

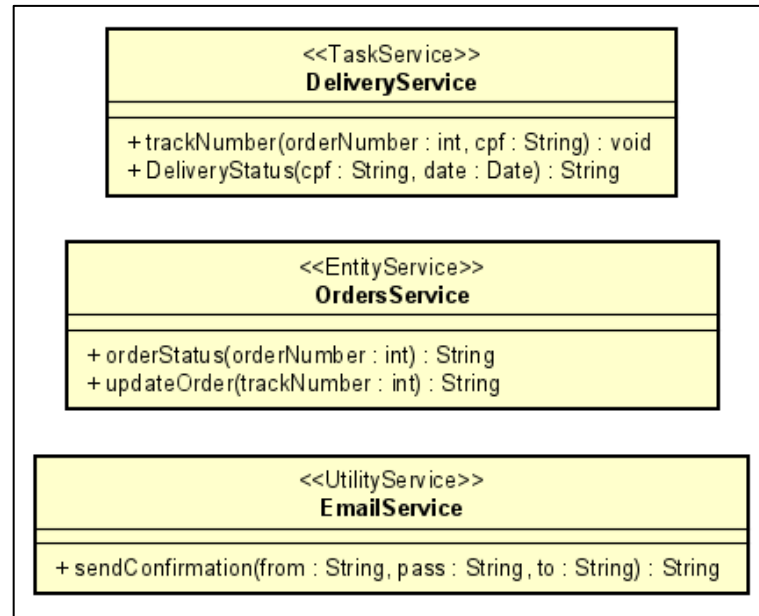


Figura 9 - Versão detalhada dos serviços e suas operações

3. Conclusão

Com isto, conclui-se a análise do processo de serviços para o Provedor de Serviços de Logística, que deixa evidente a interação entre os dois agentes no processo: entregador e sistema da loja.

A modelagem, findou-se por indicar de forma simplificada quais os serviços a serem utilizados na implementação do projeto, mostrando claramente o serviço de entrega, mensageiro e pedido, com dados e atualizações em todo o processo sobre o serviço prestado.

Por fim, como desfecho do trabalho realizado, a etapa de projeto indicou as camadas de serviço: *Task*, *Entities* e *Utility*, o que facilita a visualização do diagrama UML final, já com seus métodos esquematizados para a implementação de código.

Referências

Rocha, R. 2018. Aula Prática 2 [Apostila de sala de aula] Pós-graduação em Desenvolvimento de Aplicações para Dispositivos Móveis e *Cloud Computing*, Santa Rita do Sapucaí, Minas Gerais.

Rocha, R. 2020. Atividade Avaliativa [Apostila de sala de aula] Pós-graduação em Desenvolvimento de Aplicações para Dispositivos Móveis e *Cloud Computing*, Santa Rita do Sapucaí, Minas Gerais.

Rocha, R. 2020. Cloud Computing, Programação OO, Web Services e SOA [Apostila de sala de aula] Pós-graduação em Desenvolvimento de Aplicações para Dispositivos Móveis e *Cloud Computing*, Santa Rita do Sapucaí, Minas Gerais.