

MODELAGEM DE SOFTWARE

Prof. Saulo Popov Zambiasi

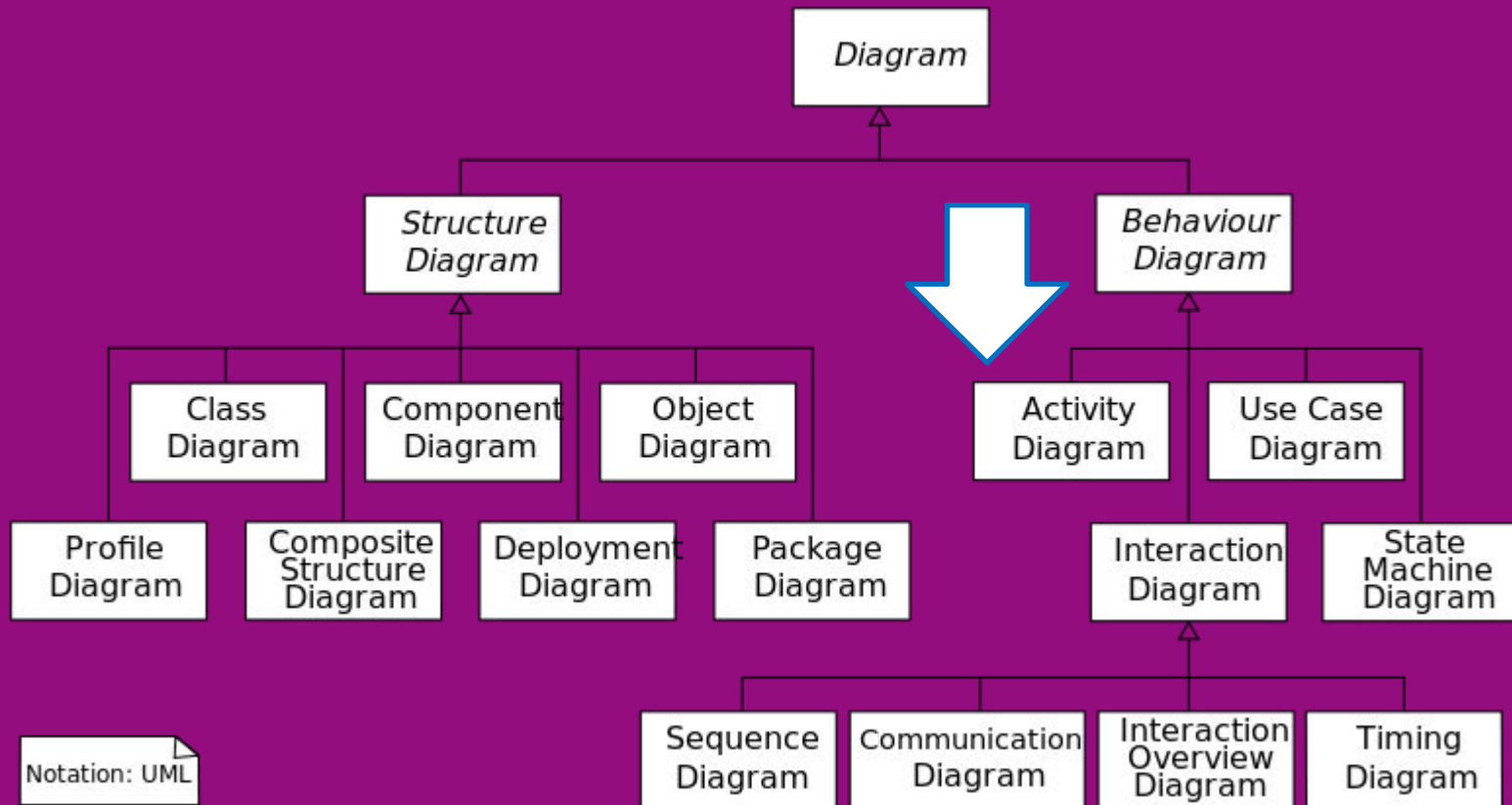
Prof. Richard Henrique de Souza

Prof. Ricardo Ribeiro Assink

Prof. Edson Lessa

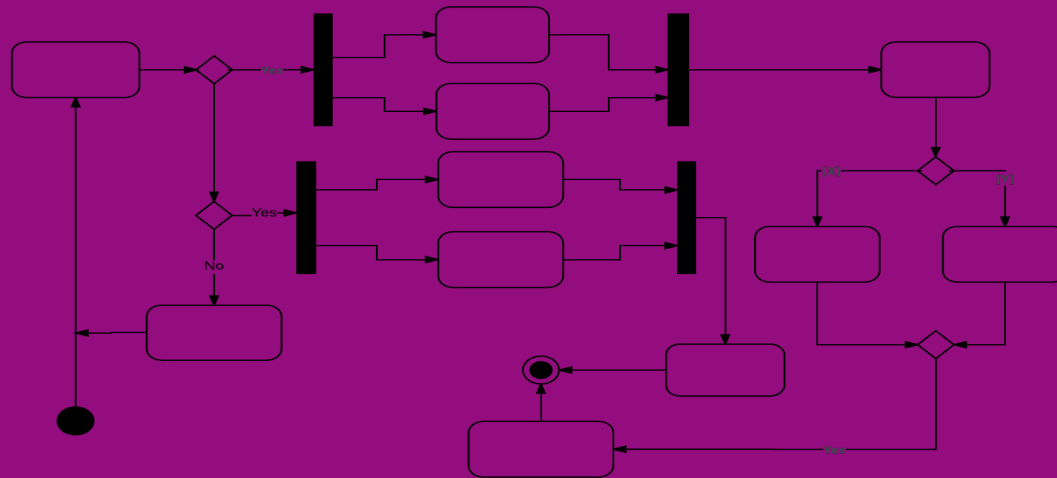


Unified Modeling Language - UML



Unified Modeling Language - UML

Diagrama de atividades



```
private $host;
private $username;
private $password;
private $database;
private $charset;

public function __construct($link = null) {
    $this::$link = mysql_connect($this::$host, $this::$username, $this::$password);
    if (!$this::$link) {
        throw new MySQLException($this::$link);
    }
    mysql_select_db($this::$database, $this::$link);
    mysql_set_charset($this::$charset, $this::$link);
}
```

Diagrama de atividades

Diagrama de Atividades

– Diagrama de atividades permite modelar o comportamento do sistema, denotando os caminhos lógicos que um processo pode seguir. Ele é um dos diagramas que compõem a visão dinâmica da UML.

É necessário entender um processo para poder escrever ou gerar o código correto para o comportamento.

Diagramas de Atividades

- Os Diagramas de Atividades mostram o fluxo entre atividades (ações não-atômicas)
- São semelhantes aos antigos fluxogramas
- São muito usados para modelar atividades concorrentes

Concorrência, Forks e Joins

- Barras de sincronização são usadas para especificar forks e joins
- Um fork representa um único fluxo de controle em vários fluxos de controle concorrentes
- Um join representa a sincronização de dois ou mais fluxos de controle concorrentes

Concorrência, Forks e Joins

- Atividades depois de um fork podem ser realizadas em qualquer ordem, ou ao mesmo tempo
- Para que as atividades depois de um join possam ser realizadas, todas as atividades antes do join devem ser concluídas

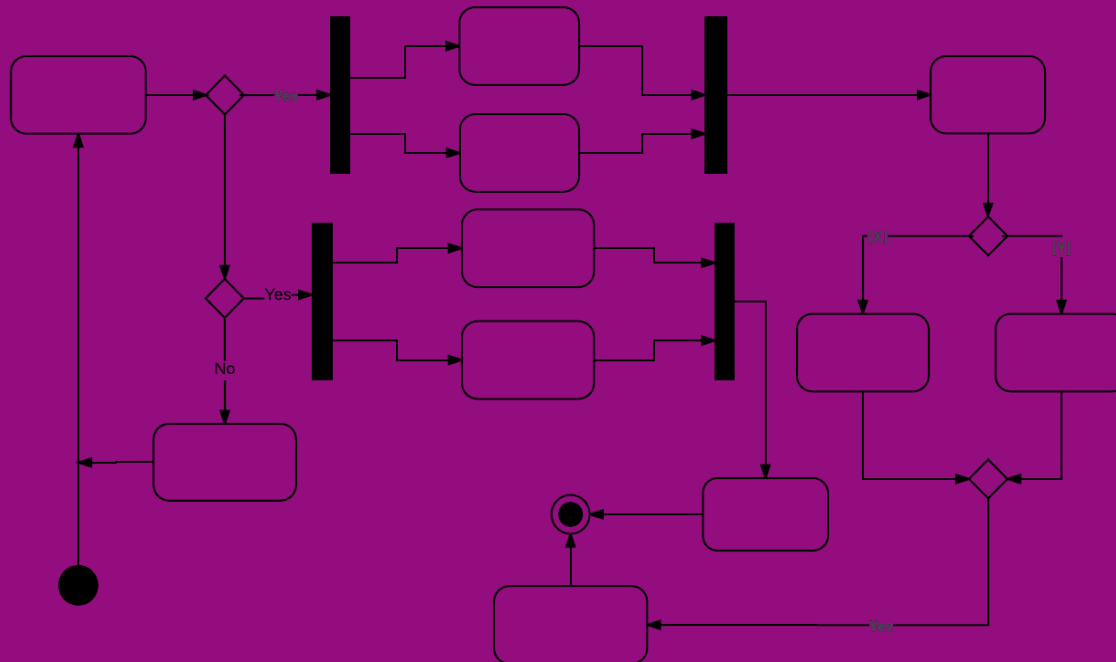


Diagrama de Atividades

Atividades e transições

– *Atividade* é uma etapa em um processo, onde algum trabalho esta sendo realizado.

Sacar Dinheiro

– Um diagrama de Atividades cotem uma série de atividades ligadas por transições, setas.



Diagrama de Atividades

Exemplo de Atividades e transições

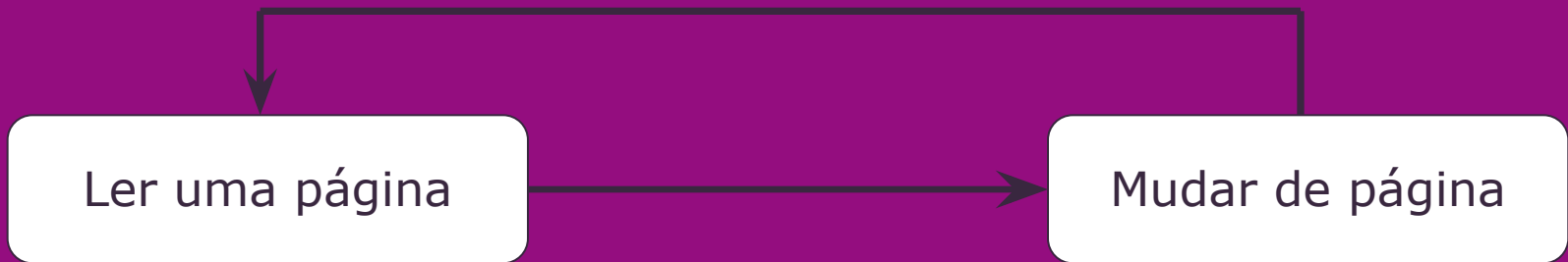
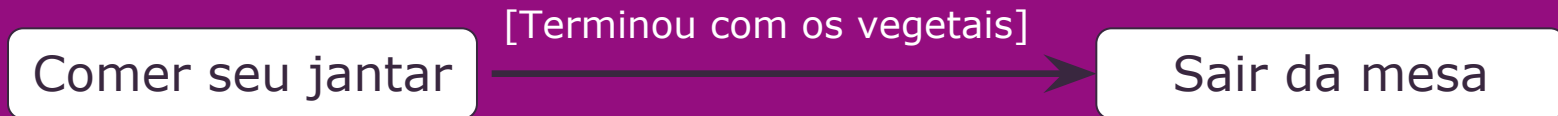


Diagrama de Atividades

Condição de guarda

A condição guarda pode ser atribuída a uma transição para restringir seu uso.



O segmento do diagrama de atividades acima, diz que não pode sair da mesa do jantar a menos que tenha acabado de comer seus vegetais.

Diagrama de Atividades

Decisões

- O losango do diagrama de Atividades é um ícone de decisão, assim como nos fluxogramas.

*No exemplo ao lado, o Caixa Eletrônico fornecerá o dinheiro ao Cliente **Se** o Saldo for Suficiente, **Senão** o sistema irá negar o saque*

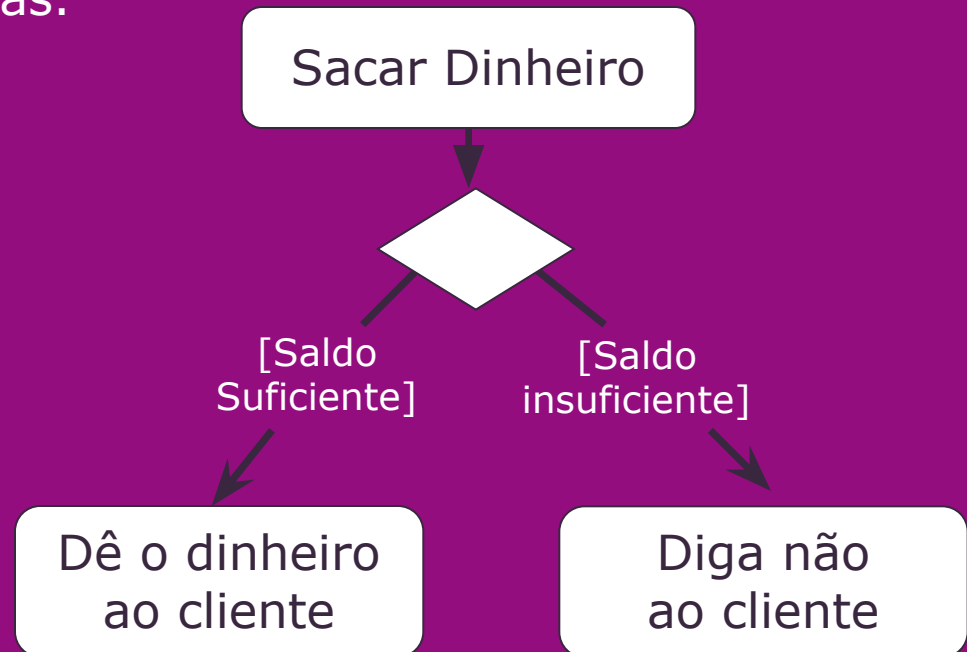


Diagrama de Atividades

Decisões

Cada opção é identificada por meio de uma condição guarda. Cada condição deve ser mutuamente exclusiva, de modo que somente uma opção seja possível em qualquer ponto de decisão.

Essa construção está relacionada a instruções **case** ou estruturas **if-else**

Diagrama de Atividades

Decisões

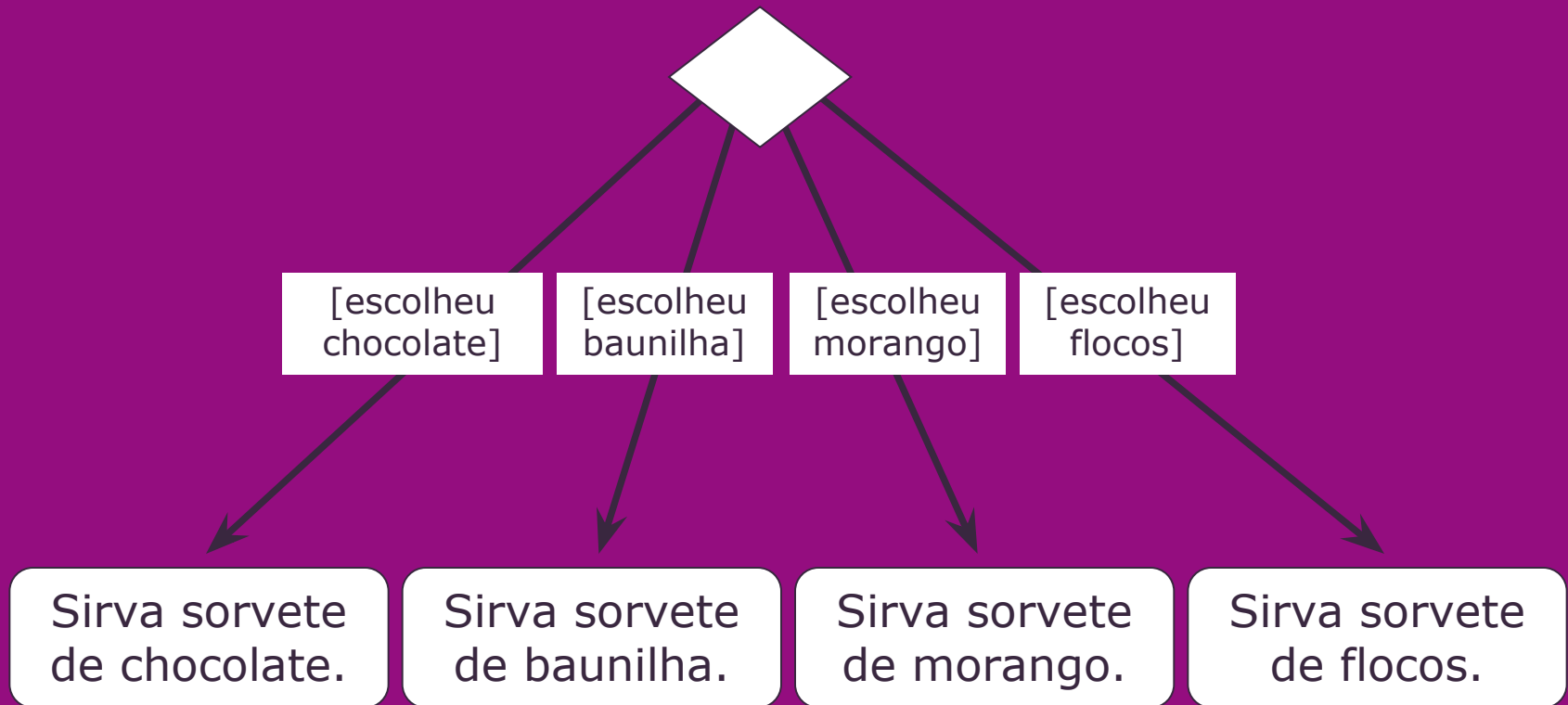


Diagrama de Atividades

Para mostrar as escolhas resultantes de uma atividade, basta modelar as transições saindo da atividade, cada uma com uma condição de guarda diferente.

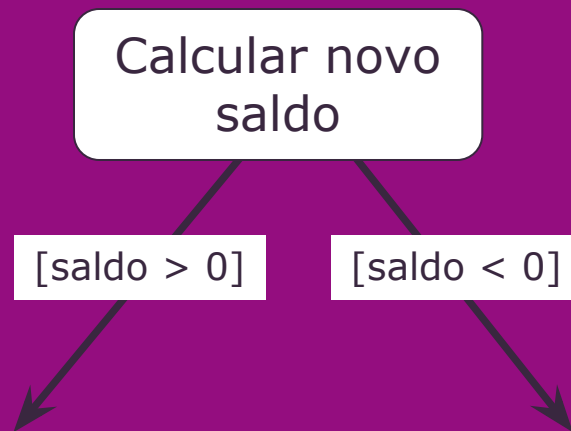


Diagrama de Atividades

Ponto de Merge

– O ícone de losango também é usado para modelar um ponto de merge, o local onde dois caminhos alternativos se juntam e continuam como um.

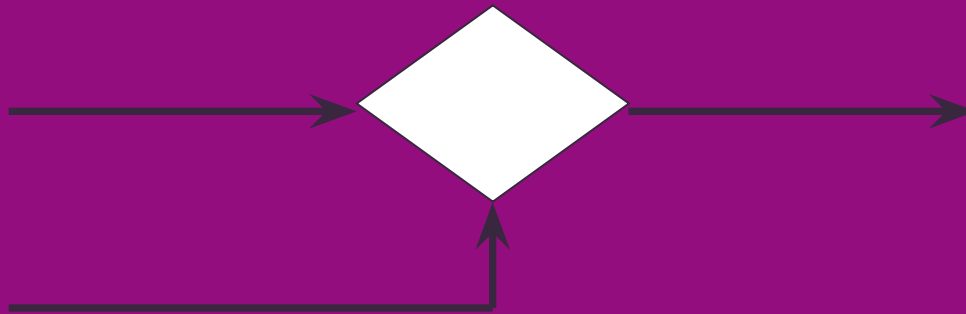


Diagrama de Atividades

Início e fim

-Ponto Inicial



-Ponto Final



Pode haver mais de um ponto final em um diagrama de atividades, pois normalmente o diagrama de atividades possui alguma lógica de decisão que resulta em caminhos alternativos, cada um com seu próprio resultado exclusivo.

Diagrama de Atividades

Concorrência

- Para mostrar que um processo simples inicia vários threads ou processos concorrentes (simultâneos).

Bifurcação

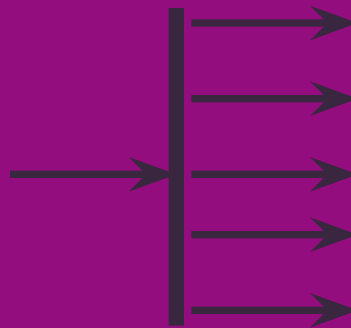


Diagrama de Atividades

Exemplo/Exercício

Nome do caso de uso: Pedido

Diálogo do caso de uso:

- O usuário Consulta Cliente

Se não for encontrado o Cliente,

- Cadastrar Cliente

Senão (localizado)

- prossegue
- Abrir Pedido

Diagrama de Atividades

Atividade Validar Cliente

Ações:

Consultar Cliente

Cadastrar Cliente

Abrir Pedido

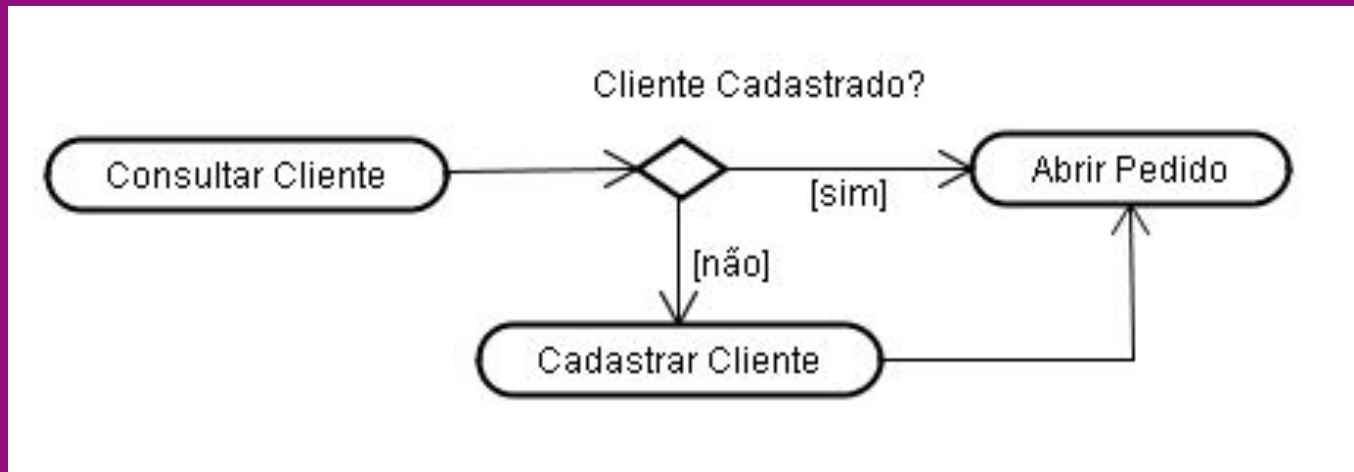


Diagrama de Atividades

Nome do caso de uso: Pedido

Diálogo do caso de uso:

- O usuário Consulta o Produto
- O usuário Consulta o Estoque
- Adiciona o Produto

Se houver mais produtos

- Consultar Produto

Senão

- Fechar Pedido

Diagrama de Atividades

Atividade Fazer Pedido

Ações:

Consultar Produto

Cadastrar Estoque

Adicionar Produto ao Pedido

Fechar Pedido

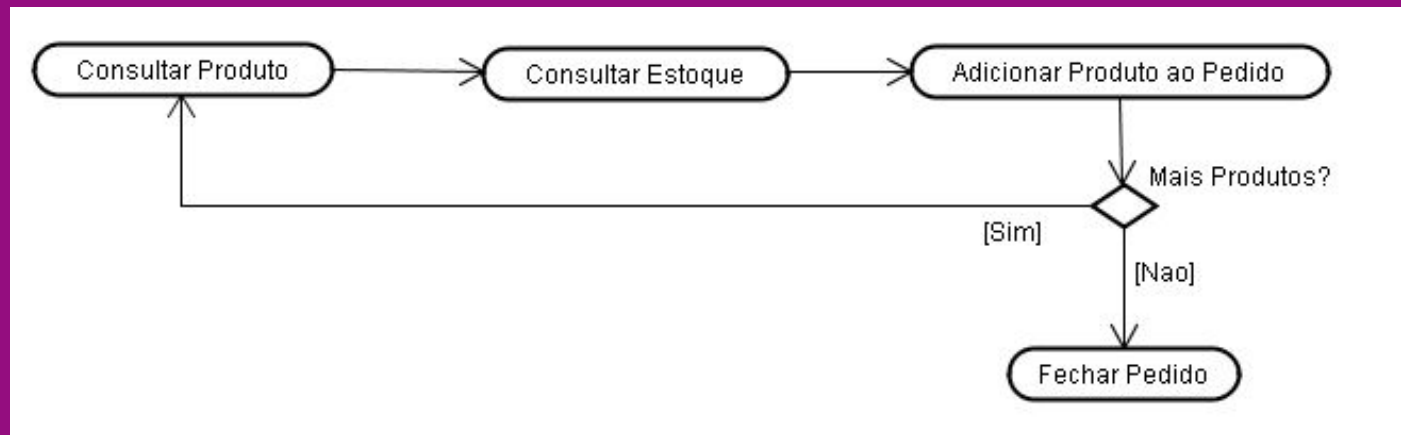


Diagrama de Atividades

Nome do caso de uso: Pedido

Diálogo do caso de uso:

- Emitir Nota Fiscal (NF)

- Confirmar Pagamento

Se inválido

- Cancelar NF

- Cancelar Pedido

Senao

- Prosseguir

- Separar Produto

- Baixar Estoque

Processos
Concorrentes

- Entregar Produto

Diagrama de Atividades

Finalização do Pedido

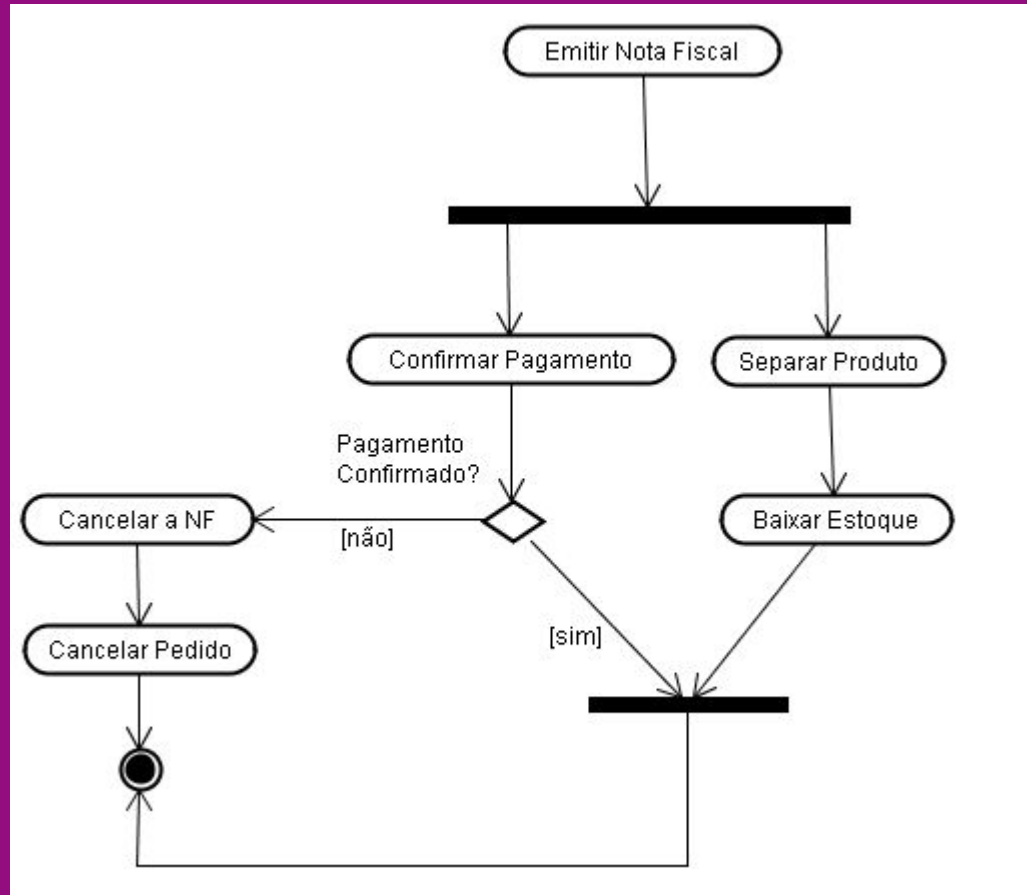
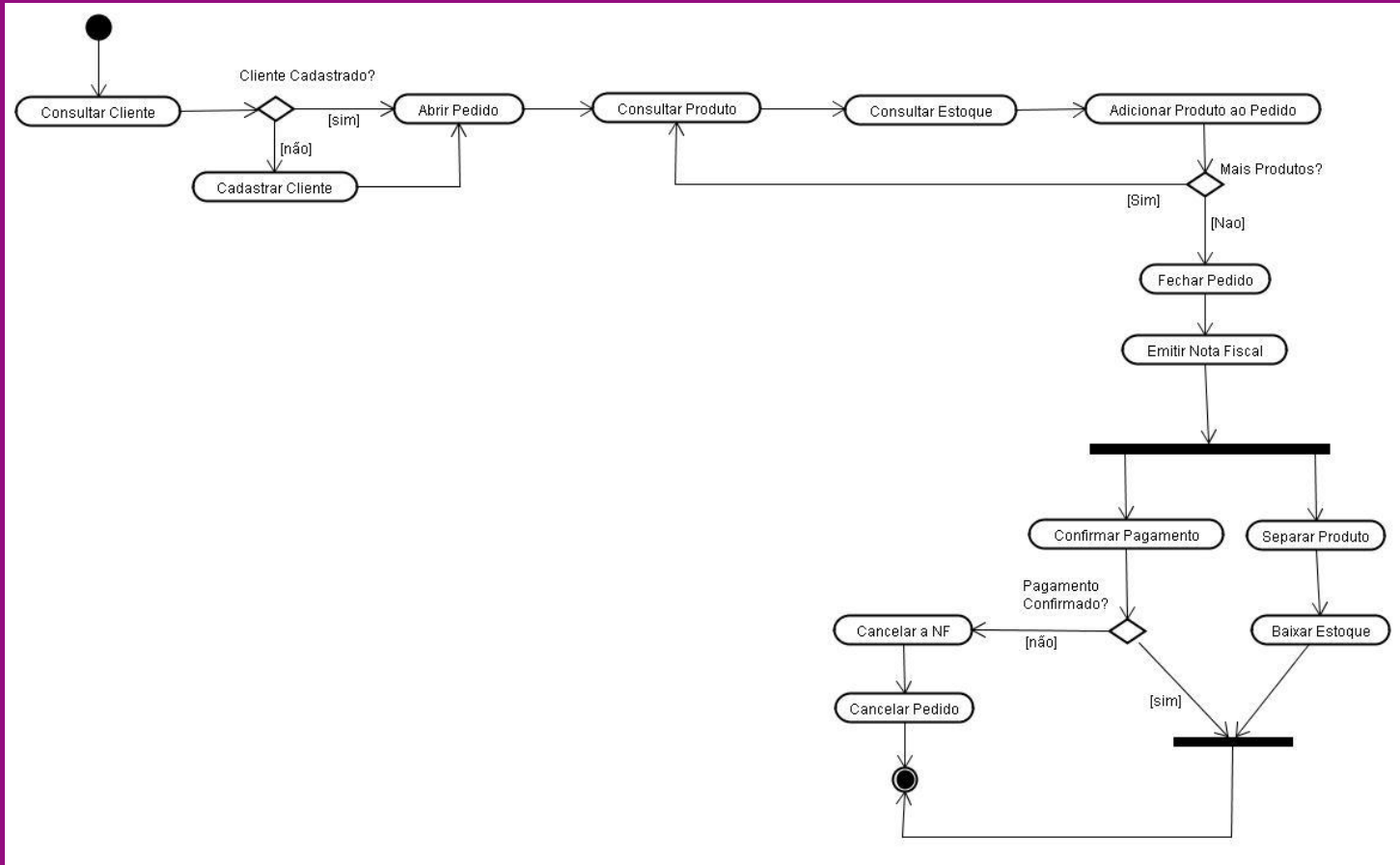
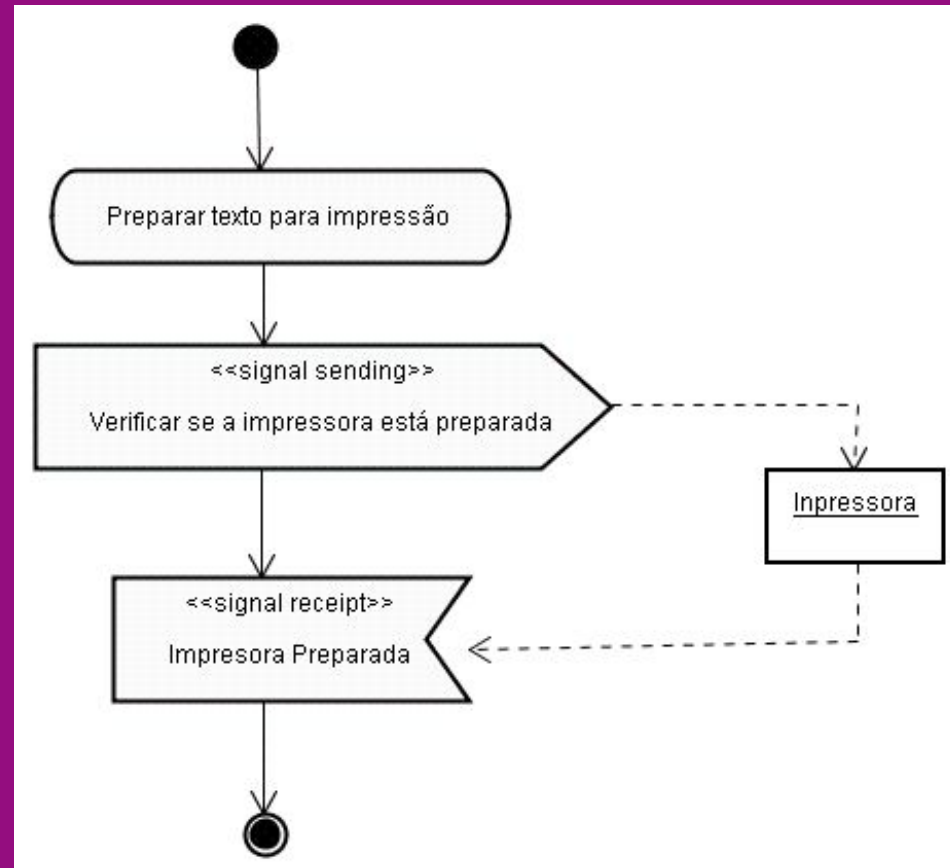


Diagrama de Atividades (Completo)



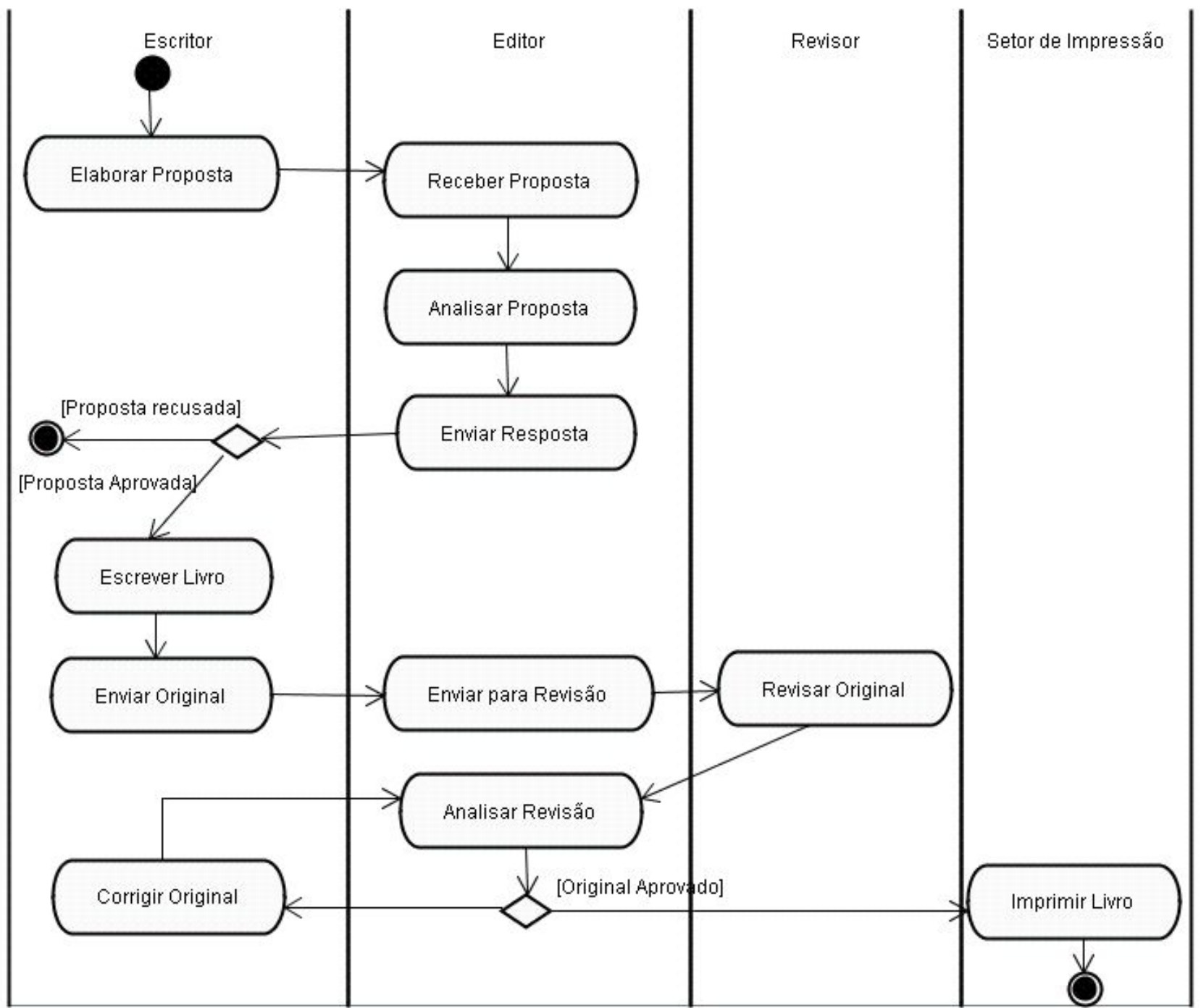
Recebimento de Sinal

- Representa o recebimento de um sinal de um dispositivo externo, normalmente um item de hardware.



Raias de Natação

- São uma extensão do Diagrama de Atividades, onde procura-se identificar os diversos setores, departamentos ou mesmo os atores que interagem com um processo.



Oracle Academy

Pessoal de Florianópolis (dib) e Palhoça (PB)

- Enviar os 4 certificados para o email: Richard.Souza@animaeducacao.com.br
Se não receber confirmação de recebimento em 24 horas entre em contato novamente por email ou presencialmente.
- No título do email: Oracle Academy – Modelagem e Programação
- No corpo do email: nome completo, email, unidade (PB ou dib) e curso
- Presencialmente na PB nas segundas, terças, quartas e sextas no LAB1 (bloco e) E na DIB as quintas a partir das 17:45

Pessoal de Tubarão

- Para informações sobre a UC de Programação de Soluções Computacionais entrar em contato com o professor Luciano.luciano.savio@animaeducacao.com.br pelo Ulife e presencialmente nas aulas.
- Para informações sobre a UC de Modelagem de Softwares entrar em contato com o professor Richard Schmitz. Richard.schmitz@unisociesc.com.br ou pelo Ulife e presencialmente nas aulas.

Referência bibliográfica

A stack of five books is shown, with the top book open. The books are stacked vertically, and the open book is placed on top of the stack. The background is a solid purple color.

FOWLER, Martin e SCOTT, Kendall. Uml Essencial. 2a. Edição. Bookman. Porto Alegre, 2000.

SCHNEIDER, Geri. Applying use case: a practical guide. Addison-Wesley, 1998.

OESTEREICH, Bernd. Developing Software with UML. Addison-Wesley, 1999.

CRÉDITOS

COORDENAÇÃO



Vera Rejane Niedersberg
Schuhmacher

PROFESSORES



Rafael Lessa
Daniella Vieira

