



ENGENHARIA DE SOFTWARE I

Profº Me. Jones Artur Gonçalves

Cliente contratou um software



O que preciso
fazer?



Quais as primeiras
etapas seguir?

A network diagram is shown, consisting of several colored beads (blue, green, yellow, red) connected by thin brown string. The beads are pinned to a white surface with small silver pins. The background is slightly blurred, showing more of the network structure.

Stakeholders

- Envolvidos ou interessados no sistema

Stakeholders

- Sistema de **biblioteca** da Fatec
 - **Bibliotecária** - entrevista
 - **Atendente** - entrevista, observação do processo(ou do uso de um sistema)
 - **Alunos** - questionário ou obter os requisitos deste perfil na entrevista do coordenador e do atendente.
 - **Coordenação de curso**-- respostas da entrevista do coordenador(bibliotecário) ou entrevista com um coordenador de curso
 - **Professores** - questionário ou obter as informações pelas outras técnicas de levantamento de requisitos
 - **Sistema Acadêmico** (dados do alunos, dados de matrícula, dados do professor, ...) - leitura de documentos técnicos e de usuários e até uso do sistema
 - **Diretoria Acadêmica** - entrevista
 - **Diretoria da Fatec** - entrevista

O que é **escopo**?

- **Escopo** significa um objetivo que se pretende atingir, como um propósito e também pode significar o limite ou abrangência de uma operação, como um trabalho ou uma campanha.

O que é **escopo** do
sistema?

Escopo

- Sistema de **biblioteca** da Fatec - escopo
 - Controlar os empréstimos de livro
 - Reserva de livro
 - Controlar os usuários
 - Integrado ao sistema acadêmico (SIGA) - obter dados de alunos e professores
 - Registro de compras de livro
 - Sistema web
 - Estatísticas
 - Relatórios
 - ...
- Escopo negativo
 - Não terá um aplicativo para consulta



Escopo do sistema

“consiste em definir quais são as funções primárias que o software deve realizar e procura delimitar a quantidade de funções.” Pressman

Requisito



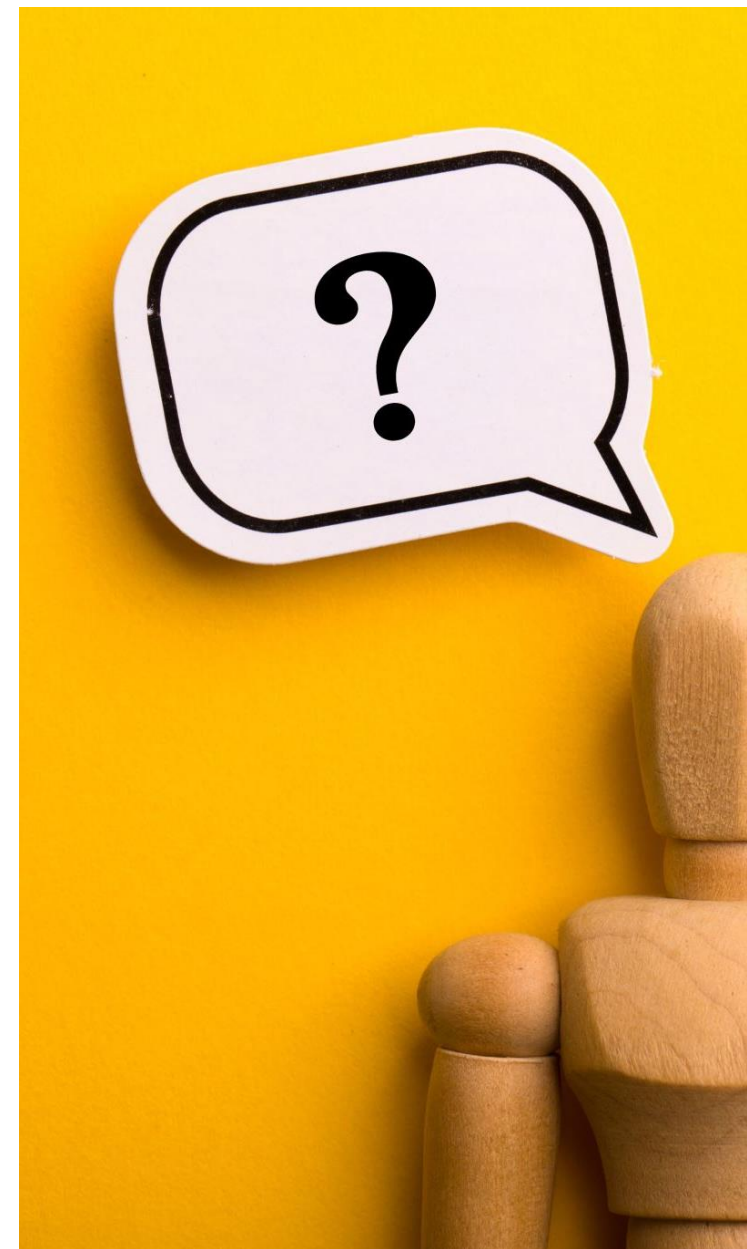
Requisito Funcional



Requisito **não** funcional

O QUE SÃO REQUISITOS?

- **Requisitos são solicitações, desejos, necessidades.**



Requisitos

◎ Funcionais

- Descrevem as funcionalidades que se espera que o sistema disponibilize, de uma forma completa e consistente.
- Relacionados a Entradas, Funções, Saídas, Atores.

◎ Não-funcionais

- Referem-se às restrições nas quais o sistema deve operar ou propriedades emergentes do sistema (como viabilidade ou tempos de resposta).
- Tipos
 - ◎ Produto (Eficiência, Portabilidade, Segurança, etc.);
 - ◎ Organizacionais (Padrões, Entrega, etc.);
 - ◎ Externos (Aspectos Éticos, Legais, etc.).

REQUISITOS FUNCIONAIS



Requisitos

◎ **Requisitos funcionais**

- ◎ Descrevem explicitamente as funcionalidades e serviços do sistema.
- ◎ Documenta como o sistema deve reagir a entradas específicas como deve se comportar em determinadas situações o que o sistema não deve fazer.

Requisitos

⦿ Atributos dos Requisitos Funcionais

⦿ Completude

⦿ Todos os serviços devem estar definidos.

⦿ Consistência

⦿ Os requisitos não devem ter definições contraditórias.

⦿ Na prática, é quase impossível atingir completude e consistência dos requisitos.

Requisitos

Exemplos de Requisitos Funcionais

[RF - 01] Cadastrar Produtos

O sistema deverá permitir cadastrar novos produtos com todos os seus atributos (descrição, preço, quantidade, etc.) e também a variação do produto livro, salvando seus dados específicos (autor, editora, ano de edição). O cadastro não poderá ser realizado no caso de já existir no estoque um produto com o mesmo código de barra.

[RF - 02] Remover Produto

O sistema deverá permitir a exclusão de produtos por nome ou código de barra. Atualizando a base de dados e cancelamento de pedidos pendentes.

[RF - 03] Alterar Produto

O sistema atualizará os dados específico de cada produto existente na base de dados. Exceto código de barras.

Requisitos

Exemplos de Requisitos Funcionais



[RF - 04] Consultar Produto

O sistema permitirá a consulta de um produto já existente no banco de dados. Através do sistema padrão, como também por uma página web, a consulta sempre será feita pelo código identificador do produto ou pelo seu nome. A consulta web será possível somente para o cliente pesquisar se o produto contém estoque na empresa, através de uma página web da empresa.



[RF - 05] Registrar Movimentação Caixa

O sistema registrará toda movimentação diária do caixa. Como entrada e saída de capital, venda de produto, pagamento de contas, pagamento a fornecedor e recebido cliente



[RF - 06] Preencher Horário

O sistema irá conter uma tela aonde o funcionário irá preencher o seu horário de entrada e saída, através de dois botões, que irão pegar o dia, mês, ano e hora do sistema e cadastrar na base de dados.



[RF - 07]Gerar Relatório

O sistema possuirá um modulo de geração de relatórios, aonde poderá ser acessado por um usuário gerente. O gerente irá escolher um relatório, dentre uma lista, para que seja gerado e posteriormente impresso. Os relatórios possuíram as mais diferentes naturezas, como financeiros, horários cumpridos no mês por determinado funcionário entre outros tipos de relatórios.

Requisitos

Gerar nota de restituição

Identificação:	Nome:	
RF 018	Gerar nota de restituição	
Descrição:		
<p>O usuário pode gerar uma nota que será enviada via correios para contribuintes que tenham direito a restituição. Na nota deve constar o endereço do imóvel correspondente e os dados do proprietário, além de informar os passos para realizar a solicitação de restituição do valor informado, juntamente com o valor a ser restituído.</p>		
<i>Usuários: DPLAN e ROOT</i>		
<input type="checkbox"/> Essencial	<input checked="" type="checkbox"/> Importante	<input type="checkbox"/> Desejável

REQUISITO NÃO FUNCIONAL

- Características
- Restrições
- Precisa descrever métrica (como medir/validar o requisito)

Requisitos

- © **Requisitos não-funcionais** são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade e tecnologias envolvidas.
- © Não é preciso o cliente dizer sobre eles, pois eles são características mínimas de um software de qualidade, ficando a cargo do desenvolvedor optar por atender esses requisitos ou não.

Requisitos

- ◎ **Classificação dos Requisitos Não Funcionais**
- ◎ • **Requisitos de produtos** : Requisitos que especificam o comportamento do produto.Ex. portabilidade; tempo na execução; confiabilidade,mobilidade, etc.
- ◎ • **Requisitos da organização**: Requisitos decorrentes de políticas e procedimentos corporativos. Ex. padrões, infra-estrutura,etc.

Requisitos

- ◎ **Classificação dos Requisitos Não Funcionais**
- ◎ • **Requisitos externos:** Requisitos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento.
- ◎ Ex. requisitos de interoperabilidade, legislação, localização geográfica etc.
- ◎ • **Requisitos de facilidade de uso.** Ex.: usuários deverão operar o sistema após um determinado tempo de treinamento.

Requisitos

⊙ Classificação dos Requisitos Não Funcionais

- ⊙ • **Requisitos de confiabilidade.** Ex.: o sistema deverá ter alta disponibilidade, p.exemplo, 99% do tempo.
- ⊙ • **Requisitos de portabilidade.** Ex.: o sistema deverá rodar em qualquer plataforma.
- ⊙ • **Requisitos de entrega.** Ex.: um relatório de acompanhamento deverá ser fornecido toda segunda-feira.
- ⊙ • **Requisitos de implementação.** Ex.: o sistema deverá ser desenvolvido na linguagem Java.

Requisitos

◎ **Classificação dos Requisitos Não Funcionais**

◎ • **Requisitos de padrões.:** Ex. uso de programação orientada a objeto sob a plataforma A.

◎ • **Requisitos de interoperabilidade.:**Ex. o sistema deverá se comunicar com o SQL Server.

◎ • **Requisitos éticos.** Ex.: o sistema não apresentará aos usuários quaisquer dados privativos.

Requisitos

- ◎ **Classificação dos Requisitos Não Funcionais**
- ◎ • **Requisitos de eficiência.** Ex.: o sistema deverá processar n requisições por um determinado tempo.
- ◎ • **Requisitos legais.** Ex.: o sistema deverá atender às normas legais, tais como padrões, leis, etc.
- ◎ • **Requisitos de Integração.** Ex.: o sistema integra com outra aplicação.

Requisitos

◎ Classificação dos Requisitos Não Funcionais

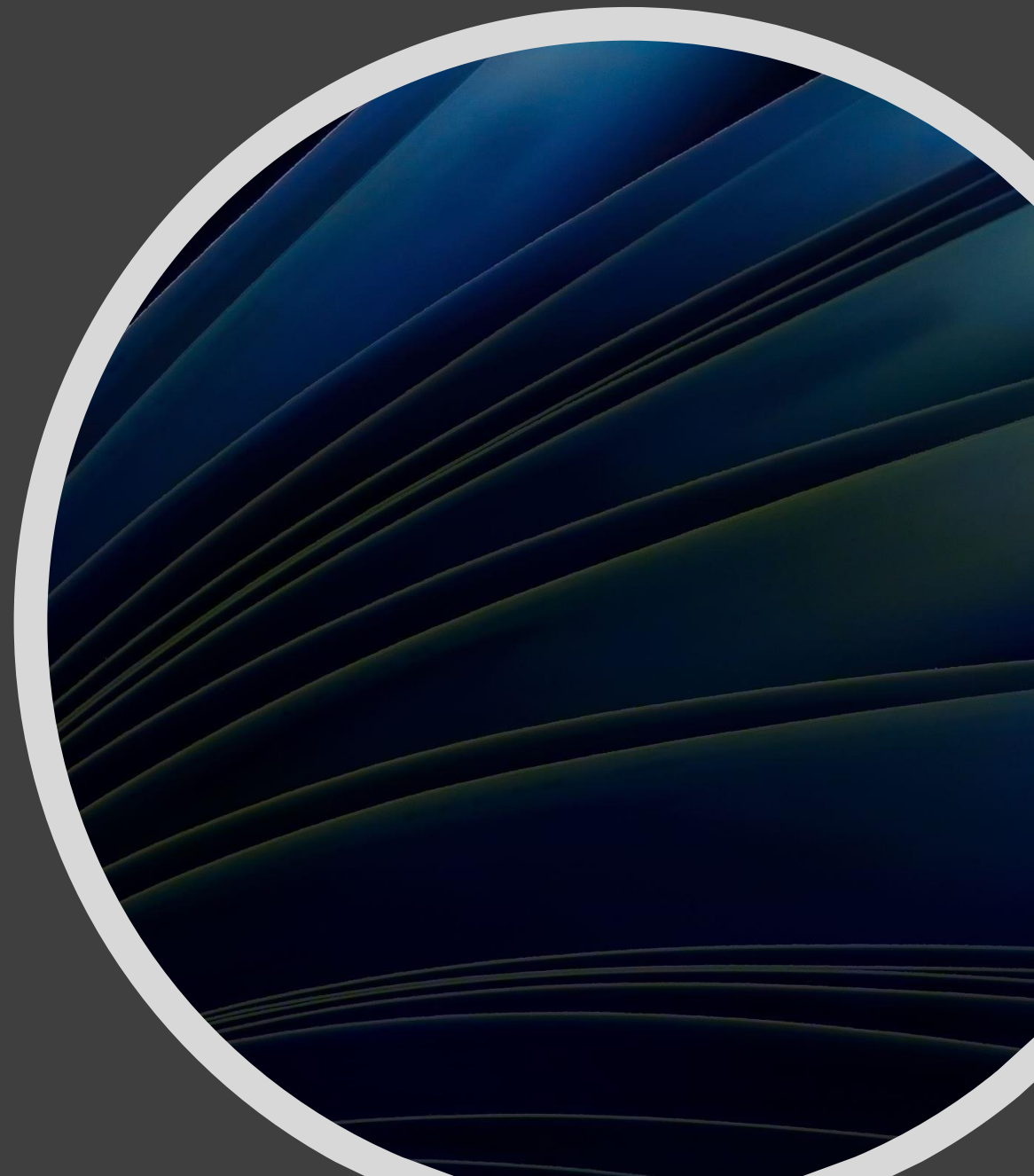
Especificação de hardware

Recurso	Quantidade	Nome e Tipo
Estação de Trabalho	12	*Processador Intel Core i3 M370 2.40GHz *3,00 GB Memória RAM *250 GB de Disco Rígido *Placa Gigabit ethernet 10/100/1000
Servidor Dell Power Edge T110	1	Processador Intel Xeon X3430 (2.4GHz, 8M Cache, Turbo) Sistema Operacional Windows Server 2008 Foundation R2 - Em Português Memória de 8GB, 2x4GB, 1333MHz, UDIMM 2 Discos Rígidos de 250 GB SATA, 7.2Krpm, Cabled, 3.5 *Placa Gigabit ethernet 10/100/1000

Requisitos de produtos

Requisitos que especificam o comportamento do produto.

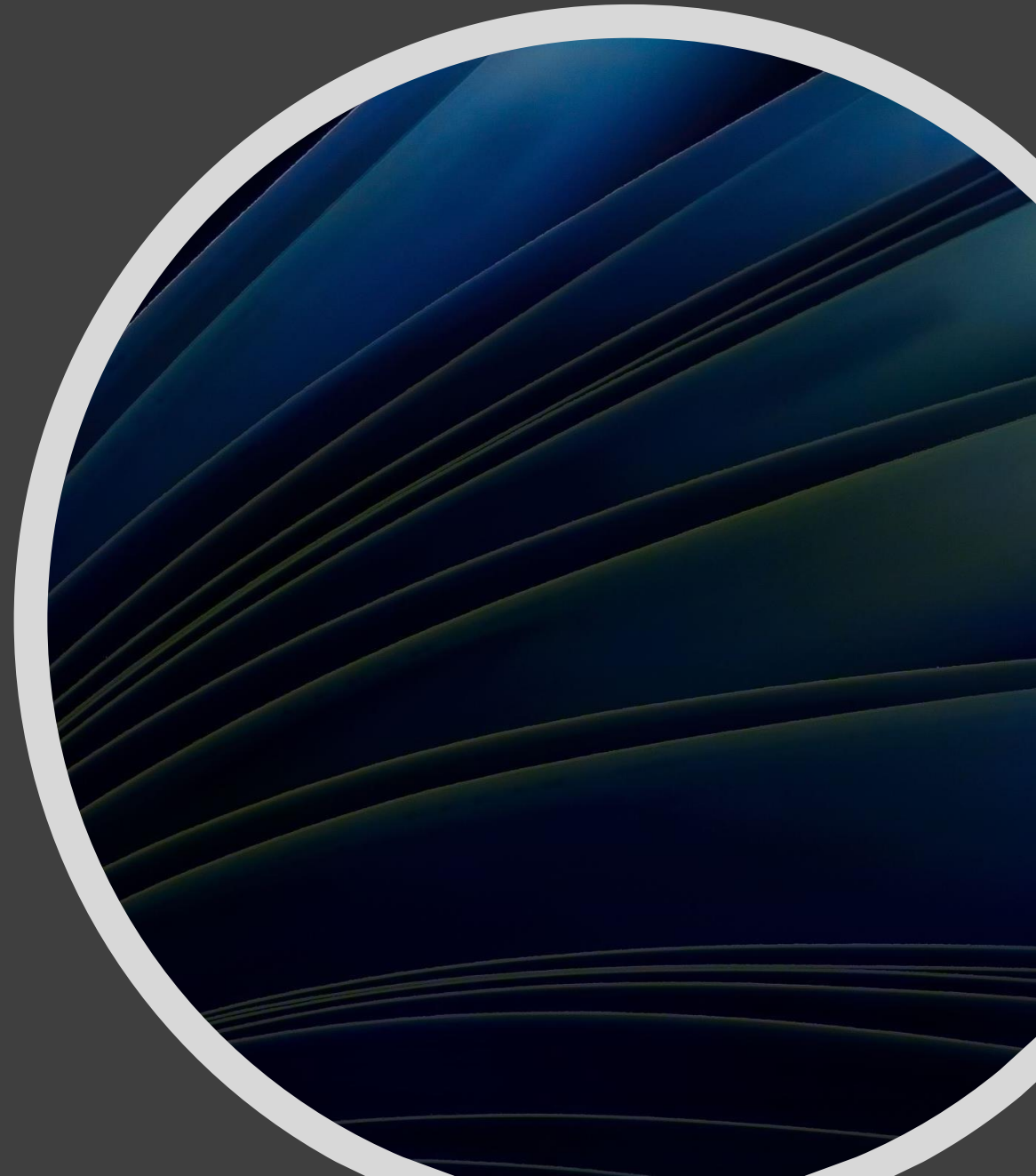
Ex. portabilidade; velocidade de execução; confiabilidade, etc.



Requisitos da organização

Requisitos decorrentes de políticas e procedimentos organizacionais.

Ex. padrões, infra-estrutura, etc.



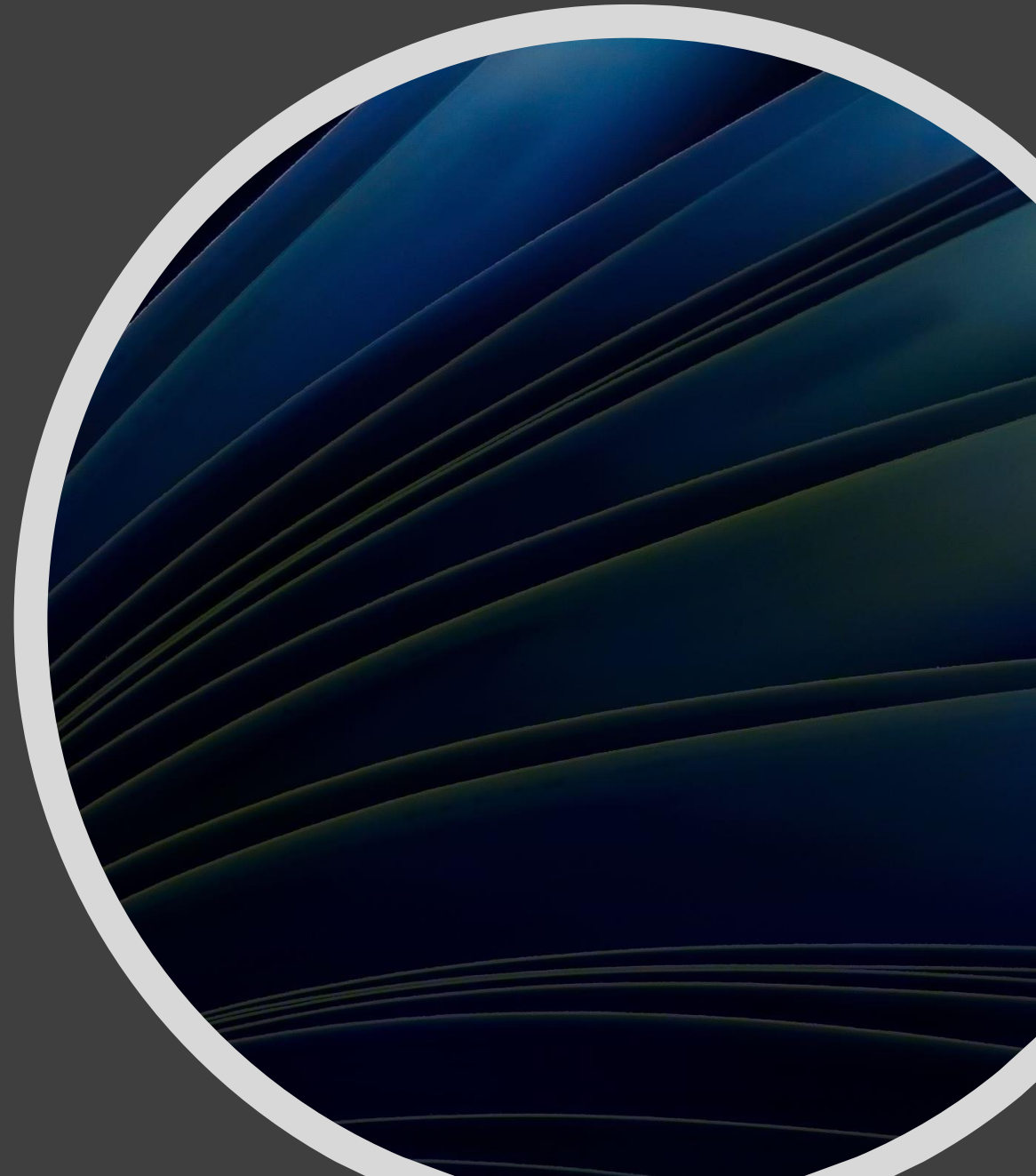
Requisitos externos

Requisitos decorrentes de fatores externos ao sistema e ao processo de desenvolvimento.

Ex. requisitos de interoperabilidade, legislação, etc.

Requisitos de interoperabilidade.

Ex.: **o sistema deverá se comunicar com o banco SQL Server.**

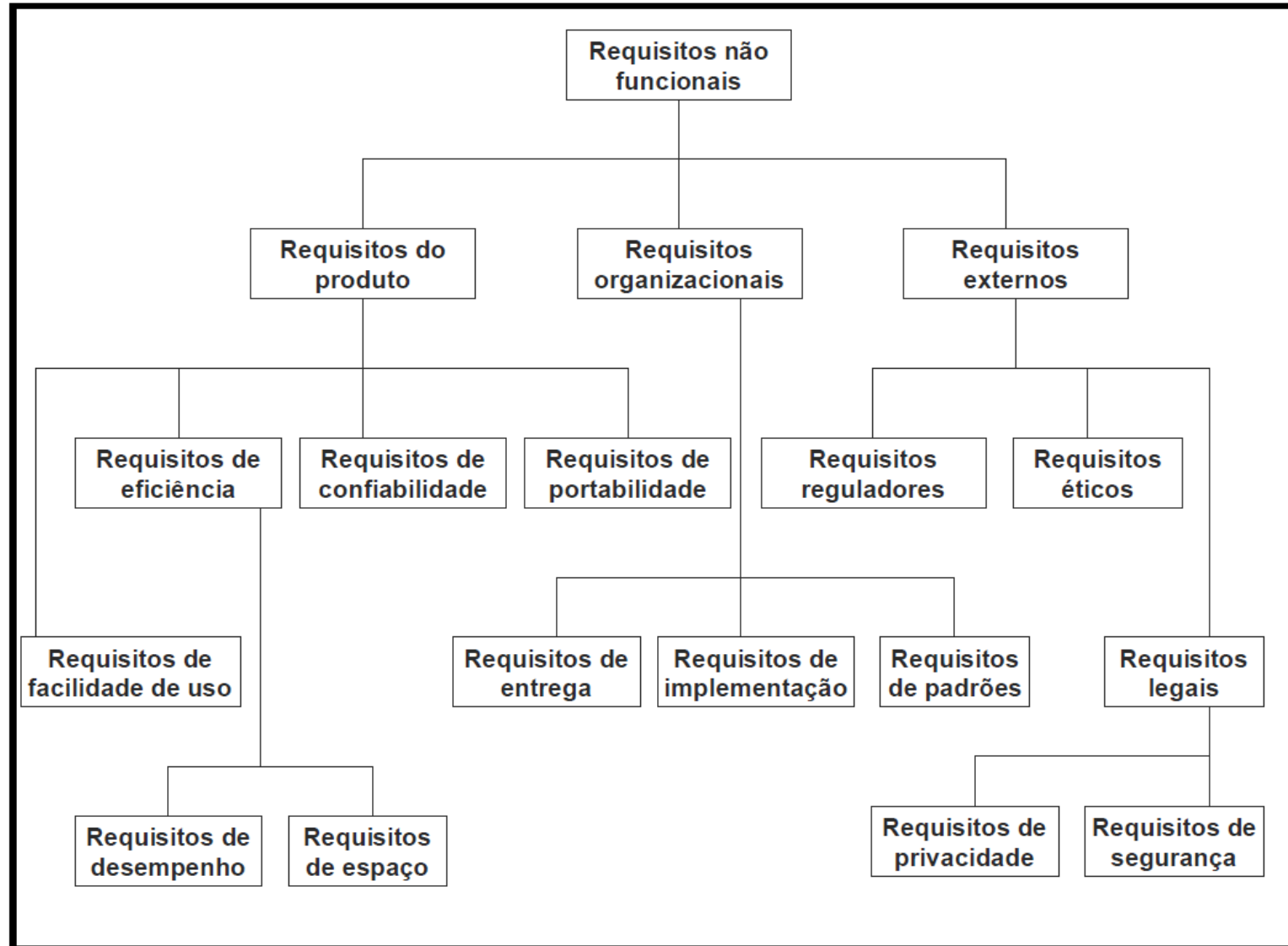


Requisitos Não-Funcionais

Métricas para Verificar Requisitos

Requisito	Métrica
Velocidade	Transações processadas/segundo Tempo de resposta ao usuário/evento Tempo de refresh da tela
Tamanho	Bytes Tamanho específico de memória
Facilidade de uso	Tempo de treinamento Número de telas de help
Confiabilidade	Tempo médio de falha Disponibilidade Taxa de ocorrência de falhas
Robustez	Tempo de reinício depois de falha Porcentagem de eventos que causam falhas Probabilidade de que dados sejam corrompidos por falhas

Requisitos



Problemas

- ◎ Grande parte dos problemas de um projeto decorre de:
 - Falta / Ineficiente compreensão dos requisitos;
 - Pouco / Inexistente feedback do cliente;
 - Requisitos mal especificados.

REQUISITO FUNCIONAL

Funcionalidades do sistema

- Usar **VERBO + SUBST**
- Exemplo:
 - **Realizar login** - o sistema deverá realizar a autenticação do usuário pelo email e senha cadastrado para cada usuário.
 - **Emprestar livro** - cada aluno poderá emprestar um livro ou revista da biblioteca e o empréstimo deverá ficar registrado no sistema. O prazo de devolução para cada livro é de 1 semana.



Sistema bancário

Identifique:- Os stakeholders

- RF

- RNF

- **Stakeholders:**
 - Clientes
 - Gerente do banco
 - Atendente(Posso ajudar?)
 - Caixa
 - Diretoria do banco
 - Acionistas
- **Sistema Bancário**
- **RF:**
 - Obter extrato
 - Criar conta bancária
 - Movimentar dinheiro
 - Realizar pagamento
 -
- **RNF:**
 - Tempo de resposta - no máximo 2 segundos por operação
 - Facilidade de uso - os usuários (clientes) precisam usar sem treinamento.
 - Segurança de Informação: precisa ter 2 tipo de validação para garantir quem está utilizando o sistema.



UML

A **Unified Modeling Language (UML)** é uma linguagem de modelagem ¹. A UML não é um método de desenvolvimento mas ele auxilia a visualizar seu desenho e a comunicação entre objetos.

Basicamente, a UML permite que desenvolvedores visualizem os produtos de seu trabalho em diagramas padronizados

Porque adotar UML?

◎ Padrão

- Academia, Indústria, etc.

◎ Notação Gráfica

- Facilita a comunicação
 - ◎ Equipe-Clientes;
 - ◎ Equipe-Equipe.

◎ Suporte de Ferramentas

- Visual Paradigma, Rational Rose, Visio, Poseidon, ArgoUML.



Casos de Uso



CASOS DE USO

- Após levantamento dos requisitos, podemos documentar as funcionalidades do sistema. Para isso utilizamos os diagramas de casos de uso

Caso de Uso

Representar as **funcionalidades** do sistema, seus atores e suas **interações**

Utiliza linguagem natural(não técnica)

Não está atrelado a tecnologia da solução

Criador: Analista de Sistemas

Quem lê: Cliente, Analista de Sistemas, Testadores, Programadores, etc

CASOS DE USO SÃO BASEADOS EM DOIS CONCEITOS

- DIAGRAMAS;
- ESPECIFICAÇÕES (PARA DOCUMENTAÇÃO);

Nome do caso de uso é bem simples (nome do requisito) e depois especificado (detalhes) nas especificações de casos de uso

Diagrama de caso de uso

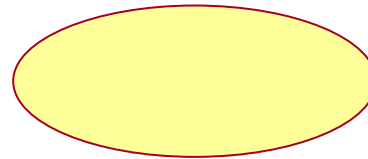
O Diagrama de Caso de Uso descreve a funcionalidade proposta para o novo sistema. Um Caso de Uso representa uma interação entre um usuário (humano ou máquina) e o sistema.

- Capturar o comportamento;
- Particiona o sistema em funcionalidades;
- Elementos
 - Atores
 - Casos de Uso
 - Relacionamentos

Diagrama de caso de uso

◎ Caso de uso

- Na Engenharia de Software, um **caso de uso** (ou *use case*) é um tipo de classificador representando uma unidade funcional coerente provida pelo sistema.

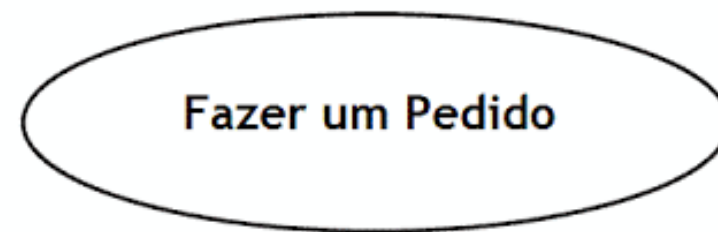


gerarRelatório

Os casos de uso foram propostos inicialmente por Ivar Jacobson em sua metodologia de desenvolvimento de sistemas orientados a objetos. Posteriormente foi incorporado à UML tornando seu uso uma prática frequente na identificação de requisitos de um sistema.

Caso de Uso

Representa o Rf ou as funcionalidades do sistema.

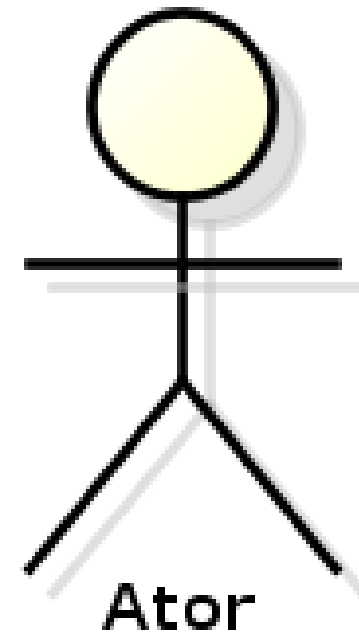


Nome = Verbo + Substantivo (indicação de ação)

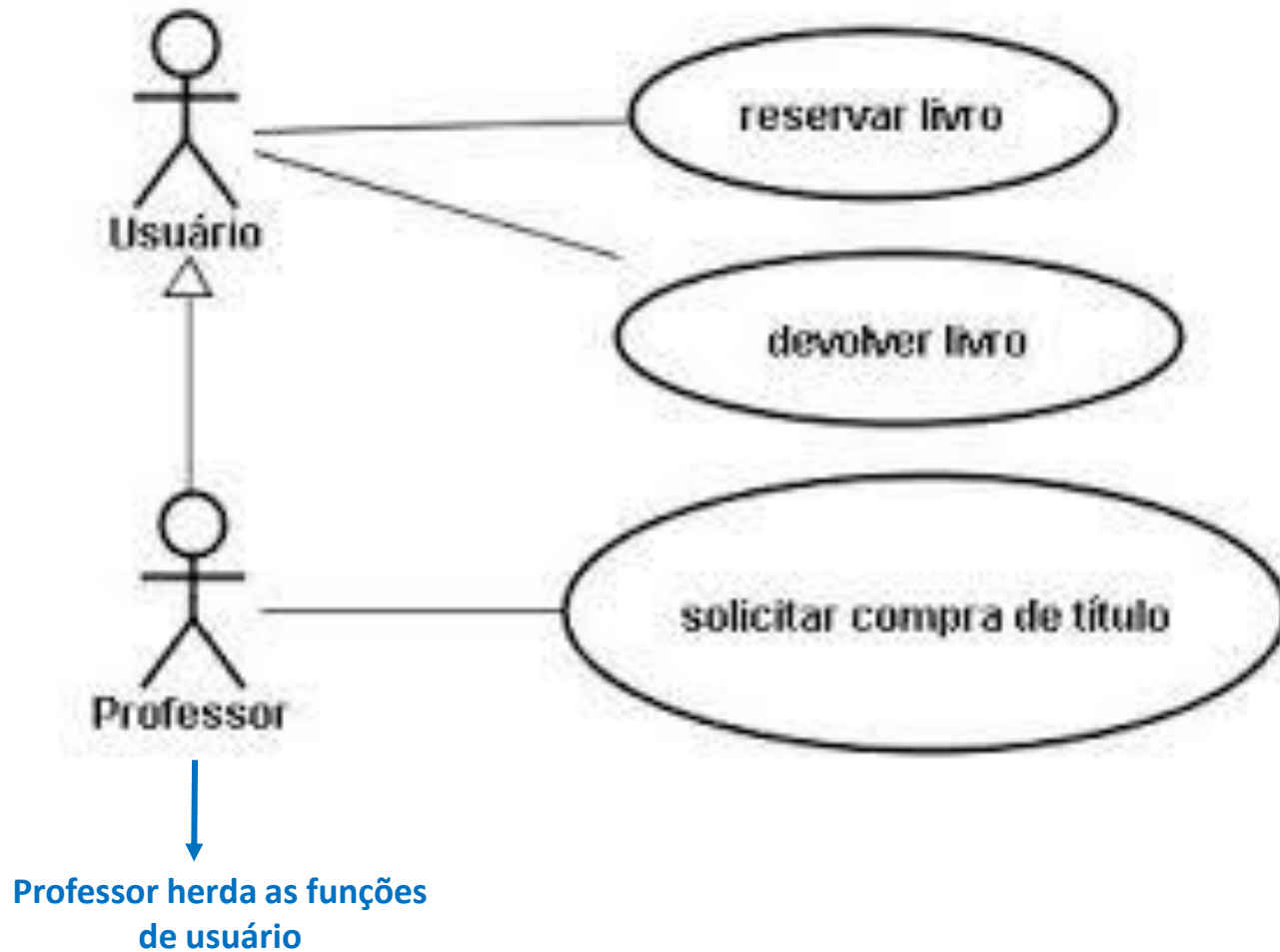
Ator

- Os stakeholders do sistema que interagem diretamente com o sistema viram Atores no Diagrama de Caso de Uso.

O ATOR NÃO É UMA PESSOA. REPRESENTA UM GRUPO DE PESSOAS (ATENDENTE, FUNCIONÁRIO, SISTEMA EXTERNO (sistema de pagamento, por exemplo), QUE INTERAGEM COM O SISTEMA



Generalização/Especialização



- Ator - Ator
- Herança
- O filho faz tudo o que o pai faz e mais algo específico

Diagrama de caso de uso

Ator(es)

- Tipicamente, um ator representa um papel que um ser humano, um dispositivo de hardware ou até outro sistema desempenha com o sistema.

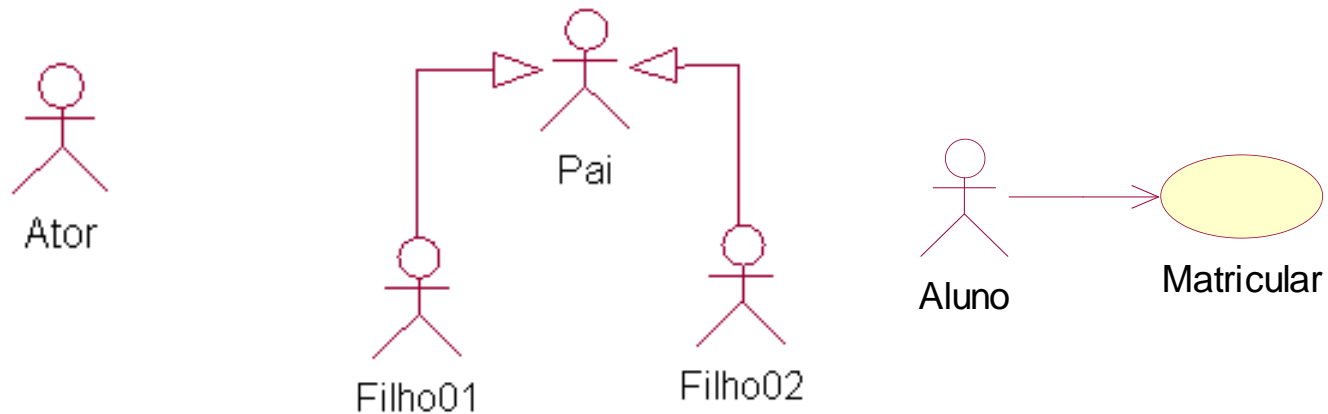
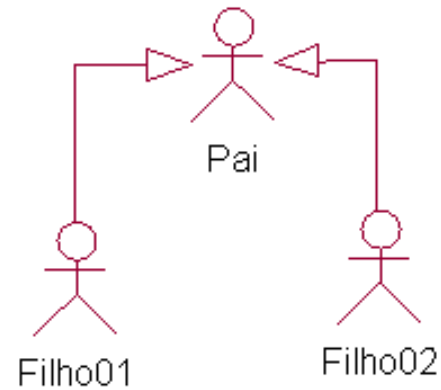


Diagrama de caso de uso

Relações:

- Entre atores



- Entre casos de uso

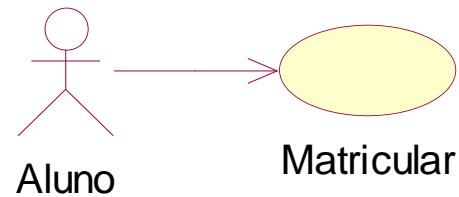
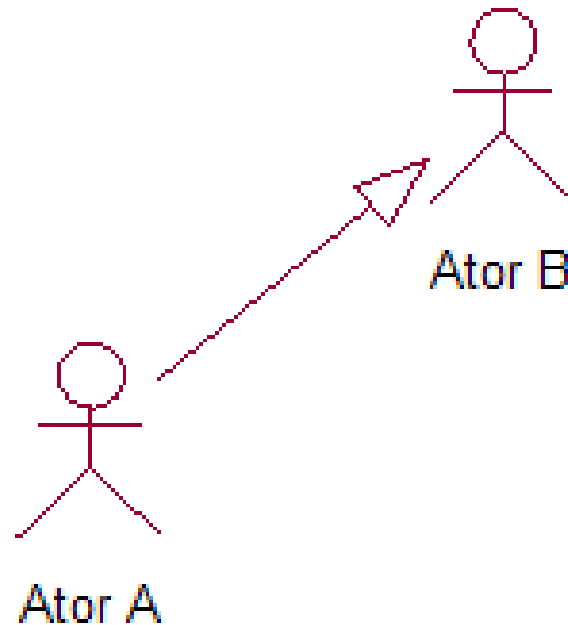
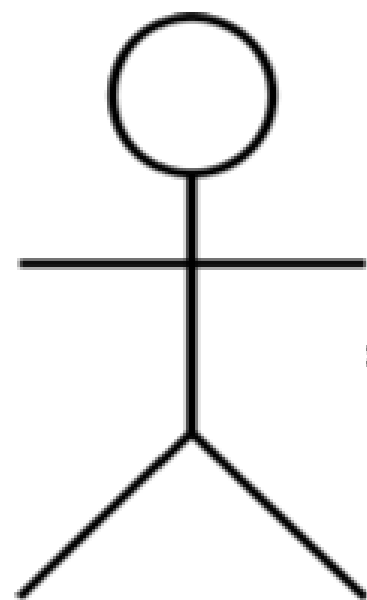


Diagrama de caso de uso

- © Os *casos de uso* de B são também *casos de uso* de A
- © A tem seus próprios *casos de uso*



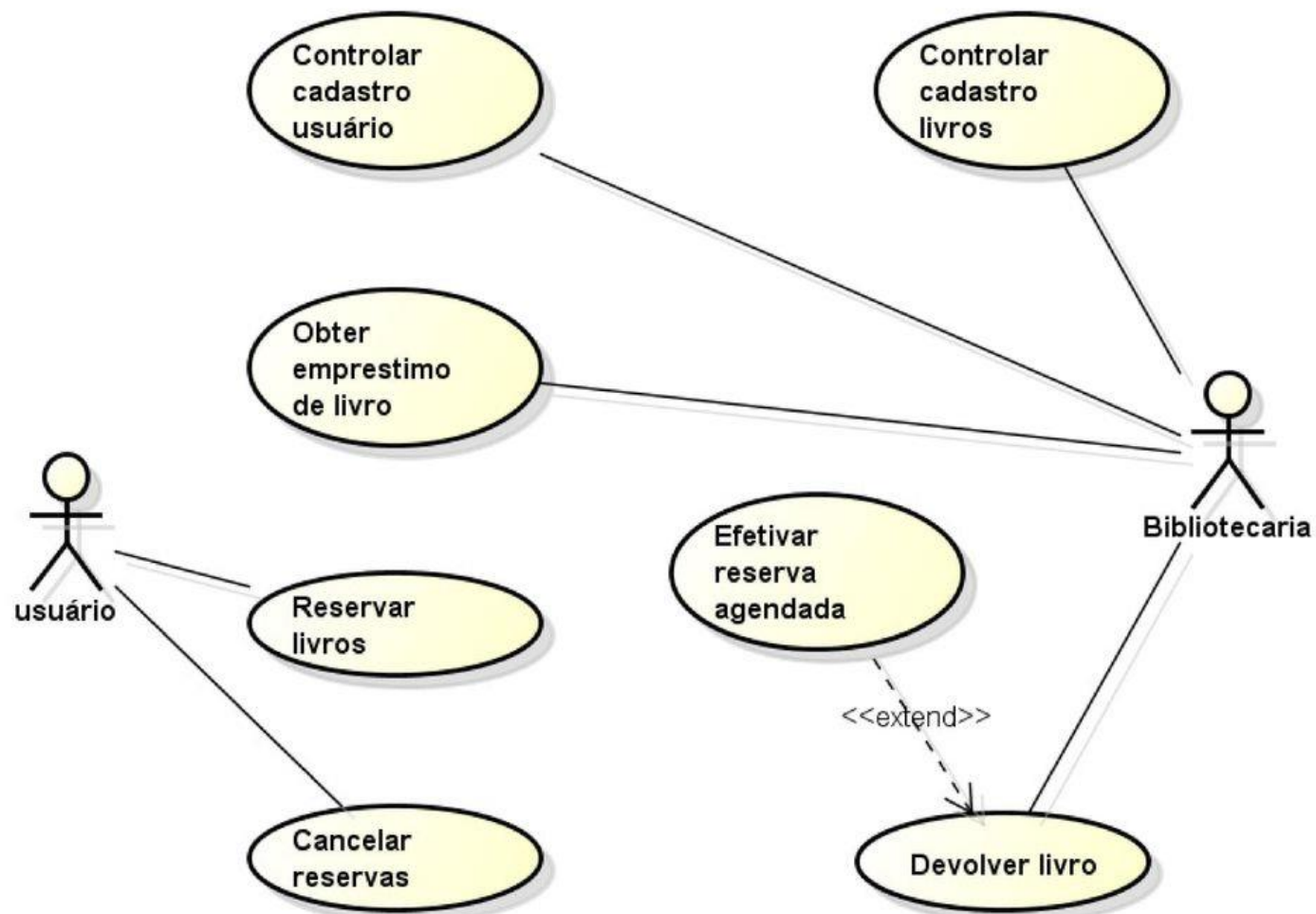


Ator

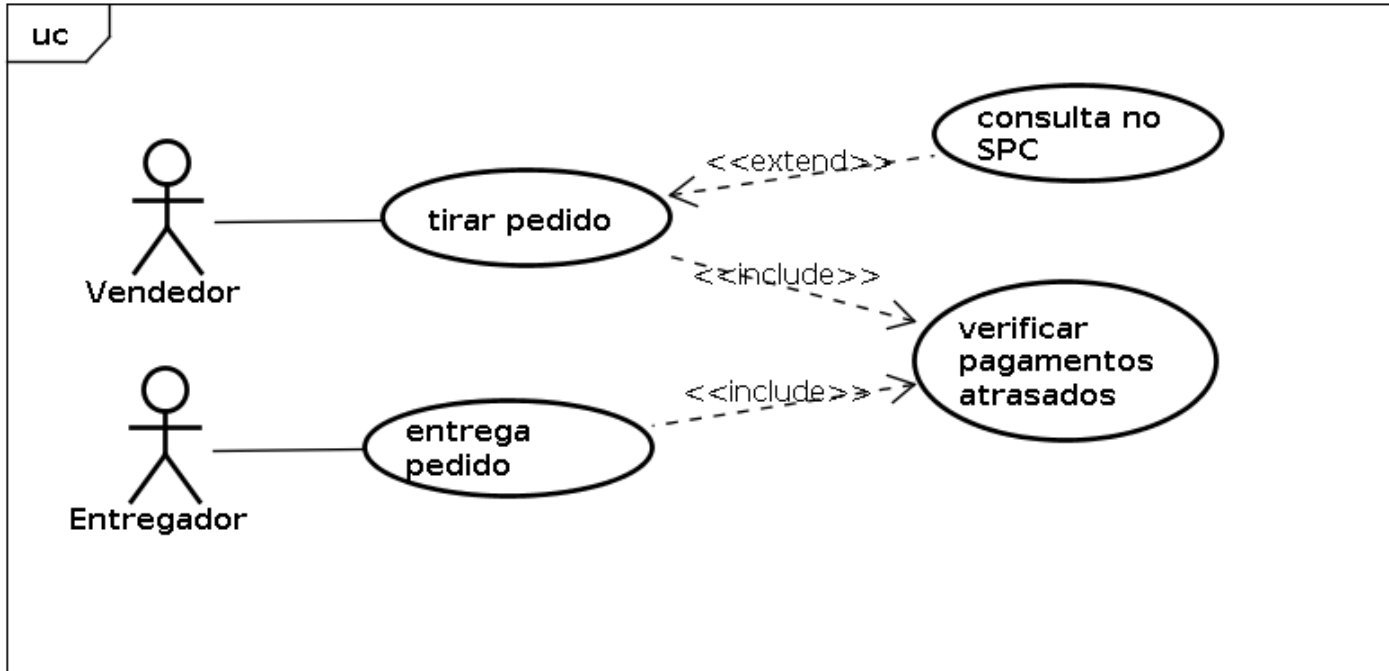
Caso de Uso 1

Caso de Uso 2

Diagrama de Caso de Uso

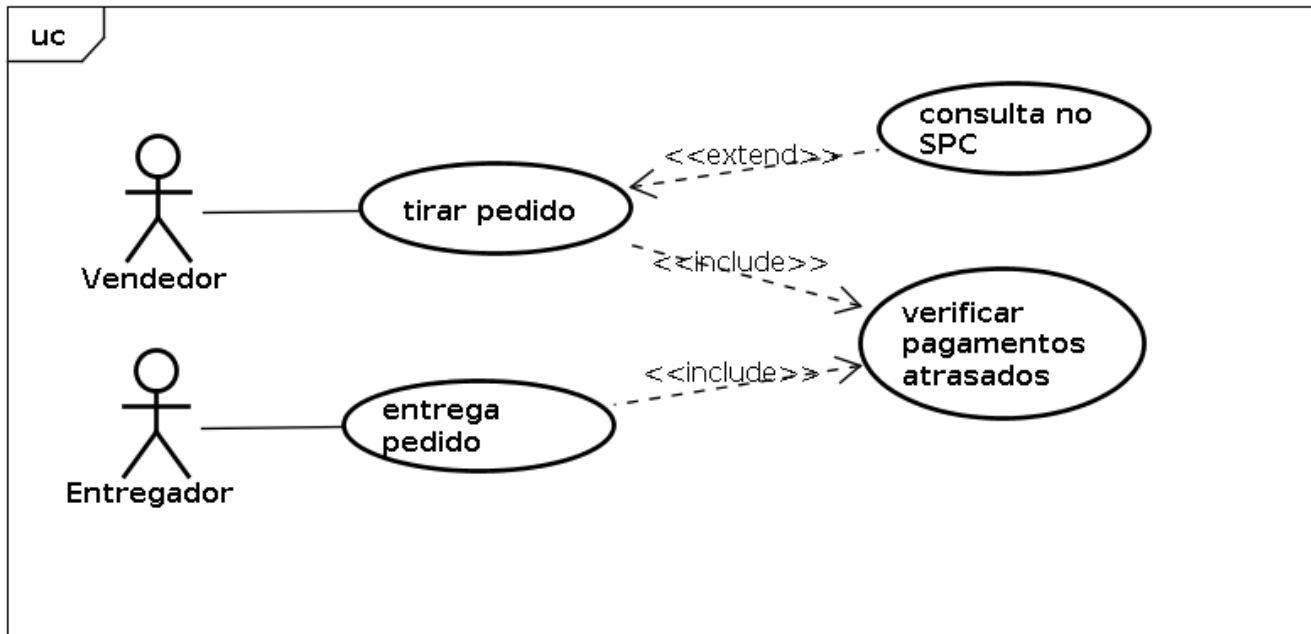


Relação :Caso de Uso - Caso de Uso



- Include ou Extend

Include

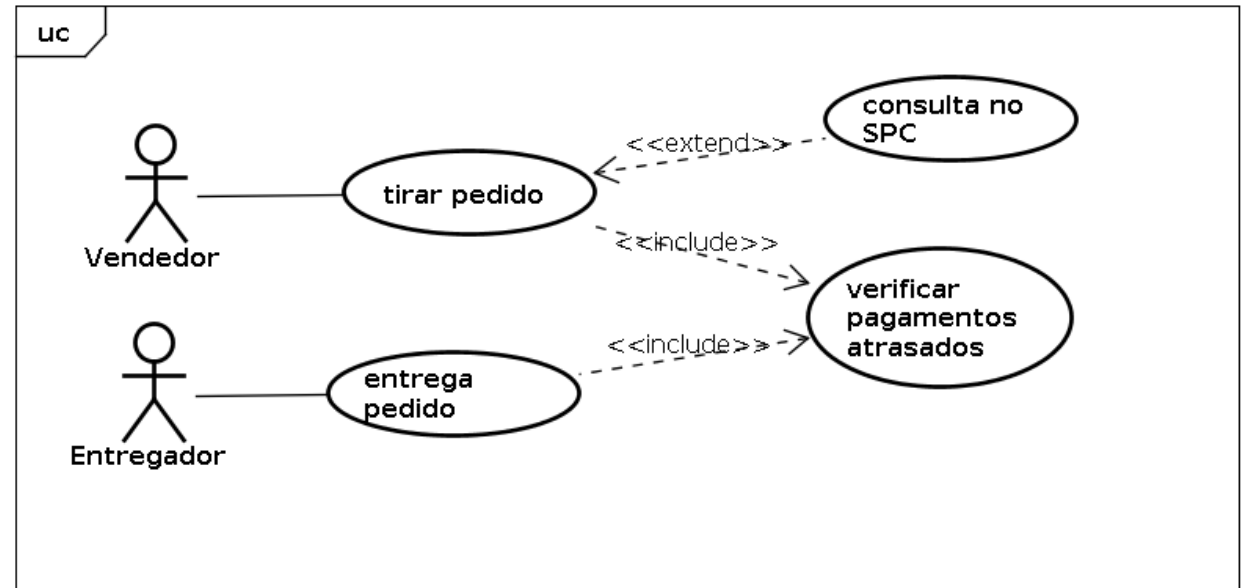


Obs.: tanto o vendedor como o entregador precisam verificar pagamentos atrasados

- **Obrigatório**(para a execução de um caso de uso acontecer, é necessária a execução de outro caso de uso)
- Aponta para o caso de uso secundário
- <<include>>

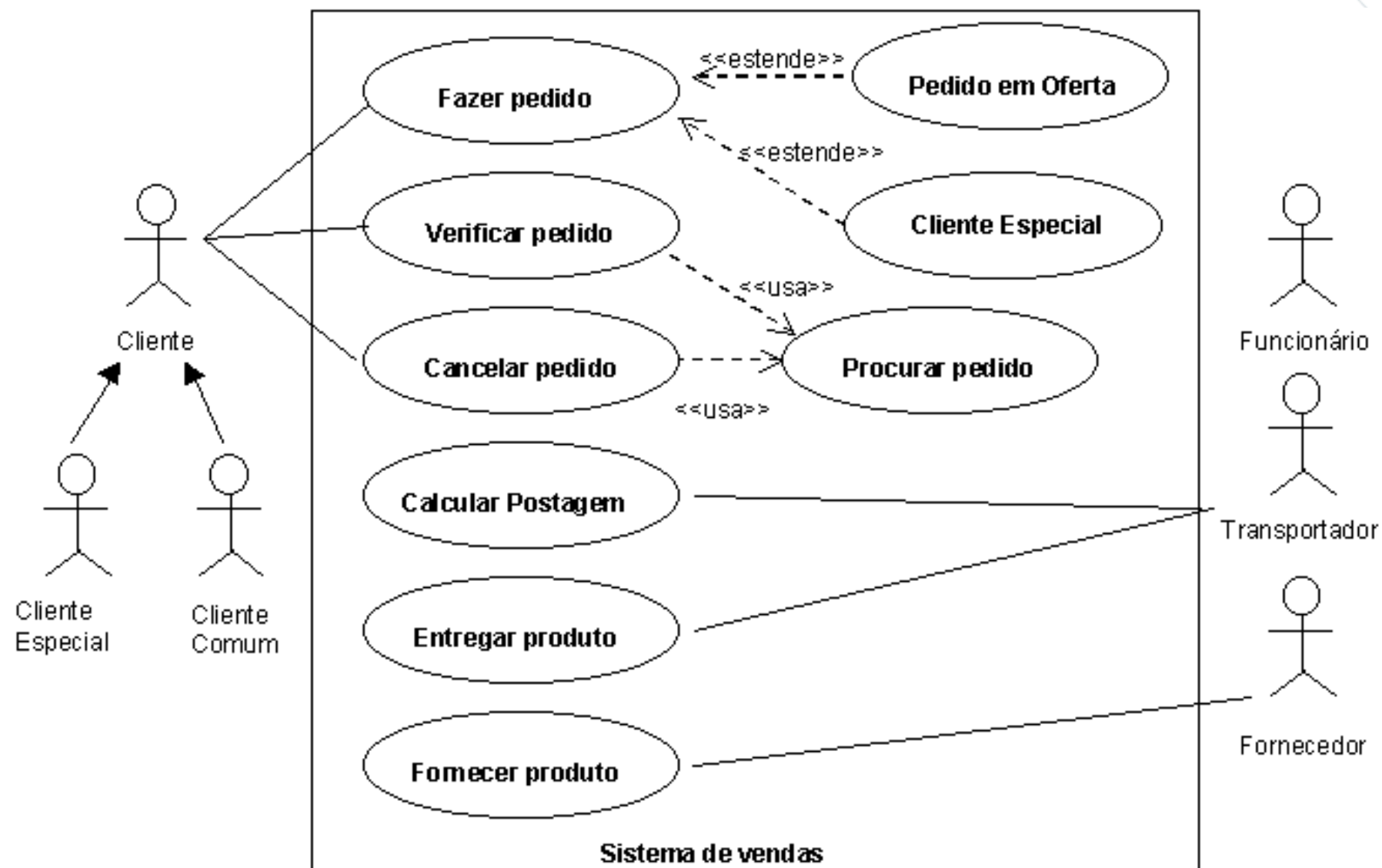
Extend

- É executado a partir de uma condição (é uma extensão, não é obrigatório)
- Aponta para o caso de uso principal
- <<extend>>



obs.: vendedor que opcionalmente consulta o SPC

Diagrama de caso de uso



IDENTIFICANDO ATORES E CASOS DE USO



QUEM UTILIZA O
SISTEMA?



COMO É O USO DO
SISTEMA?



QUAIS INFORMAÇÕES
SÃO FORNECIDAS OU
OBTIDAS PELO SISTEMA?



COMO O SISTEMA É
MANTIDO?



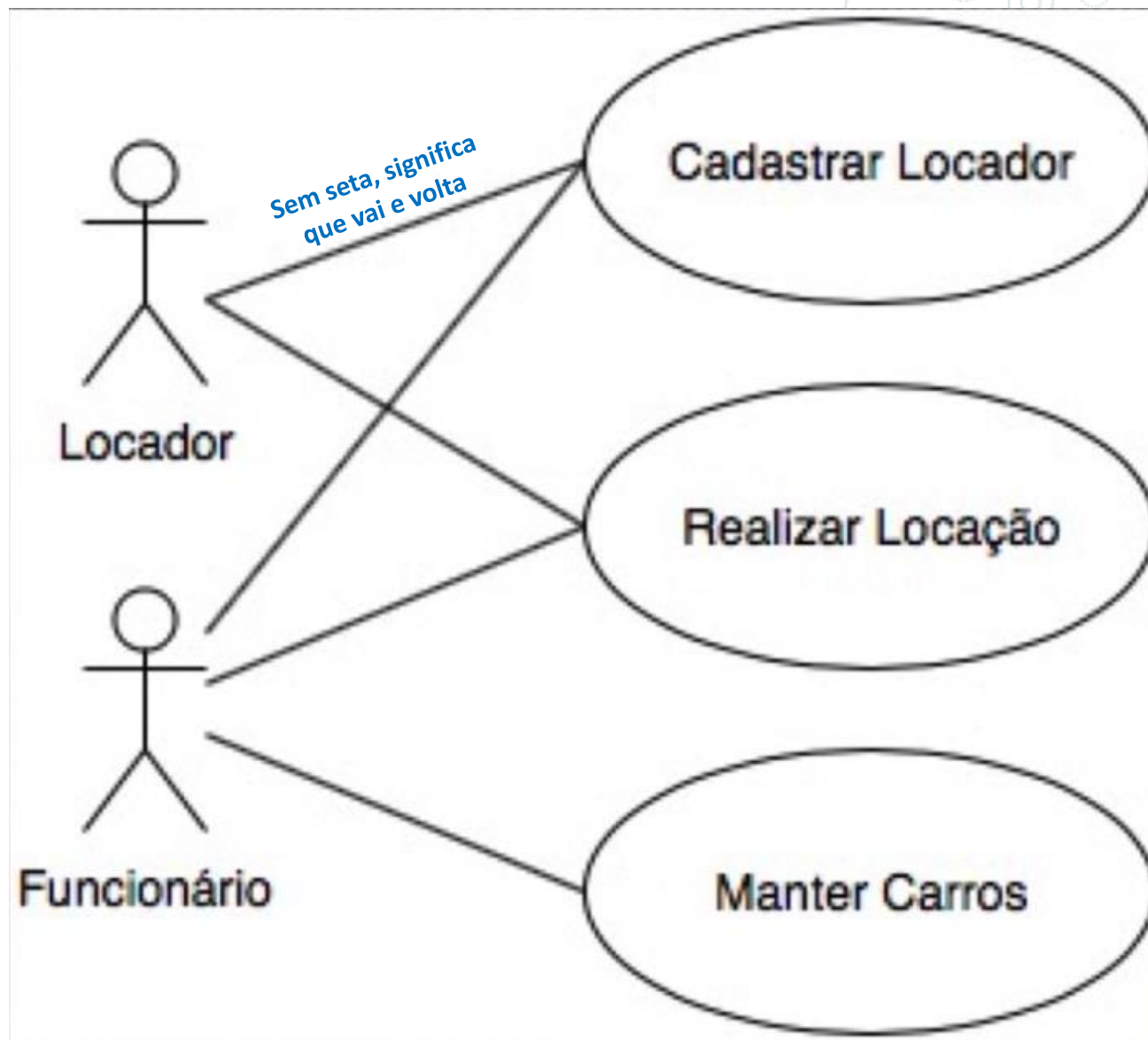
QUAIS OUTROS
SISTEMAS INTERAGEM
COM O SISTEMA?

1 - AO REALIZAR UMA LOCAÇÃO, O LOCADOR INFORMA SEU CÓDIGO. SE O CÓDIGO NÃO ESTIVER CADASTRADO, A LOCAÇÃO DEVE SER CANCELADA E O CLIENTE INFORMADO DE COMO PROCEDER.

2 - CASO ESTEJA CADASTRADO, VERIFICAR SE O LOCADOR NÃO POSSUI PENDÊNCIAS. CASO O LOCADOR NÃO TENHA PENDÊNCIAS, O FUNCIONÁRIO DEVE REGISTRAR A LOCAÇÃO E FORNECER O CARRO.

3 - O FUNCIONÁRIO DEVE REALIZAR A MANUTENÇÃO DOS CARROS, POR EXEMPLO CADASTRAR UM NOVO CARRO.

Uma possível
solução de
acordo com as
informações
fornecidas



Especificações

Nome	Realizar Locação
Atores	Locador, Funcionário
Resumo	Descreve a sequência de funções realizadas para fazer a locação de um carro.
Pré-Condições	Existir carro disponível
Pós-Condições	Pagar valor da locação
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Informar código	
	2. Verificar situação do locador
	3. Locar Carro para locador
Fluxo Alternativo (1)	
Cliente não cadastrado, encaminhar para cadastro	
Fluxo Alternativo (2)	
Locador possui pendências de pagamento, cancelar locação até que sejam satisfeitas	

Chamo um ou vários casos de usos, se houver

Se nada der errado

Descritos no texto, fazem parte de um fluxo alternativo

Exemplo de Caso de uso

◎ Realizar um saque no caixa eletrônico

Identificação	UC_01
Função	Retirar Dinheiro do caixa eletrônico
Atores	Cliente, Caixa eletrônico
Prioridade	Essencial
Pré-condição	Cliente precisa ter em mãos o cartão do banco
Pós-condição	Dinheiro sacado com sucesso
Fluxo Principal	<ul style="list-style-type: none">• Cliente insere cartão no dispositivo• Cliente digita a senha• Máquina autoriza login [FS001]• Cliente digita o montante• Máquina checa o saldo [FS002]• Máquina debita o dinheiro sacado do saldo inicial• Máquina dispõe cédulas para cliente• Máquina mostra na tela o novo saldo• Máquina ejeta cartão• Cliente retira cartão

Fluxo Secundário [FS001]	<ul style="list-style-type: none">• Senha digitada é inválida• Máquina ejeta cartão• Cliente retira cartão
-----------------------------	--

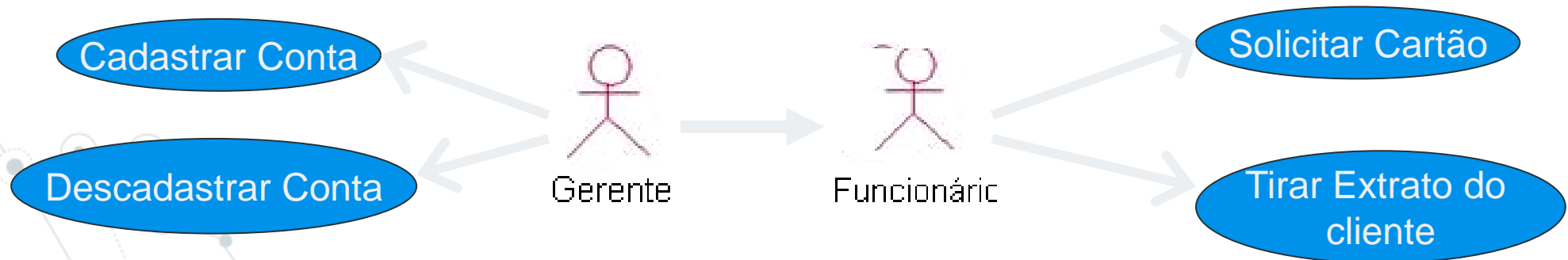
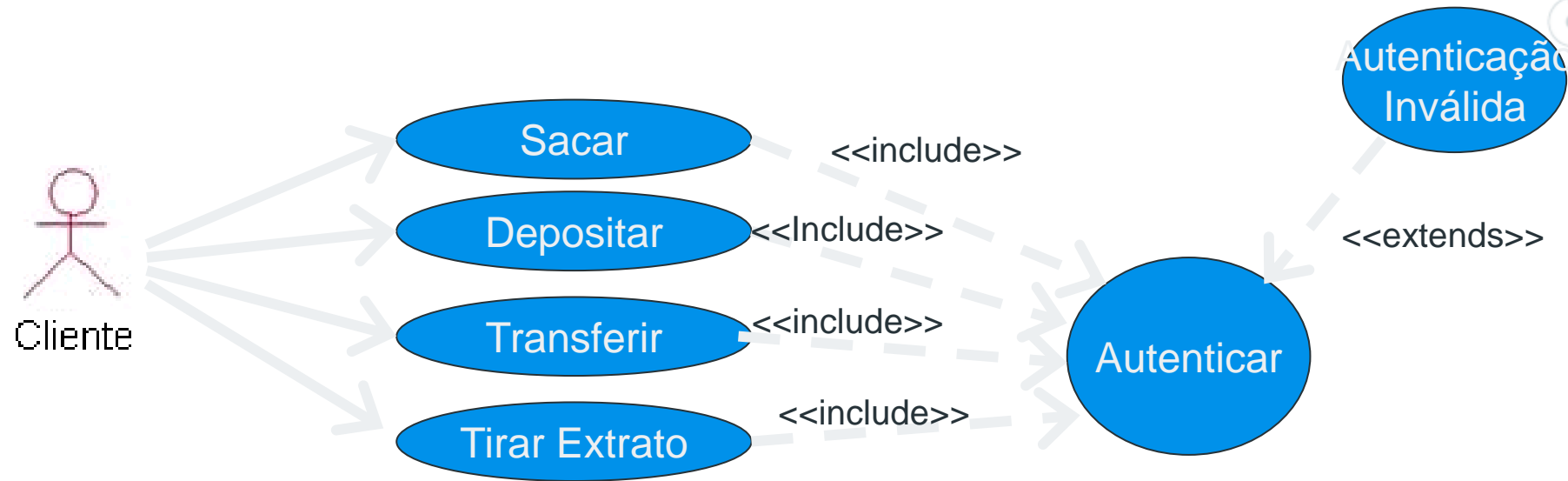
Fluxo Secundário [FS002]	<ul style="list-style-type: none">• Saldo é menor que o montante requerido• Máquina mostra na tela o saldo• Máquina ejeta o cartão• Cliente retira o cartão
-----------------------------	--

Exemplo

◎ Um sistema de Banco:

- ◎ O cliente poderá:
 - Sacar, Depositar, Transferir e Tirar Extrato;
- ◎ Para cada operação o cliente deve se autenticar;
- ◎ Qualquer funcionário poderá:
 - Tirar Extrato do cliente;
 - Solicitar Cartão de crédito para cliente;
- ◎ O Gerente pode fazer qualquer operação dos funcionários;
- ◎ Somente o Gerente pode cadastrar ou descadastrar conta;

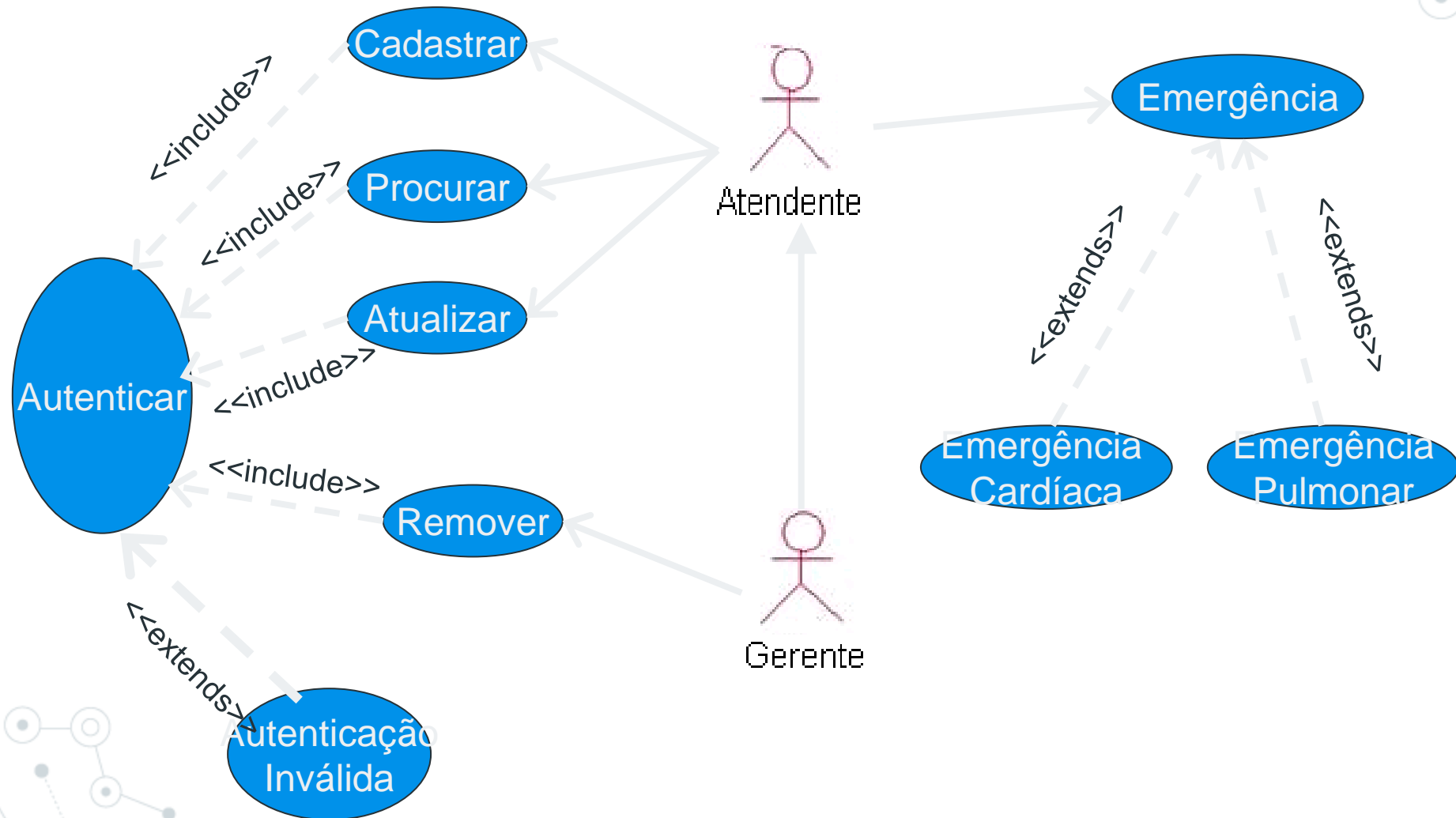
Resposta



Tarefa 1

- ◎ Um sistema de controle de hospital
 - A atendente pode acionar a emergência
 - ◎ Existem dois tipos de emergência: cardíaca e pulmonar.
 - A atendente pode cadastrar, procurar e atualizar uma emergência.
 - O gerente pode fazer tudo que a atendente faz.
 - O gerente pode remover uma emergência
 - Para cada tarefa, o usuário (qualquer que seja) deve se autenticar no sistema.

Resposta 1



The background consists of a dark blue field with a lighter blue square on the left and a semi-transparent blue circle on the right. The word "Atividades" is centered in white text.

Atividades

PI – PROJETO INTEGRADOR

- CRIAÇÃO DE UMA EMPRESA;
- GRUPOS DE 3 OU 4 INTEGRANTES;
- PARA A NOSSA DISCIPLINA:

TUDO CONTEÚDO MINISTRADO EM SALA DE AULA + APLICATIVO



PI – PROJETO INTEGRADOR

- A PARTIR DA IDEIA DO PROJETO, LISTE OS REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA QUE SERÁ CRIADO.



BIBLIOGRAFIA



BÁSICA:

BEZERRA, EDUARDO. PRINCÍPIOS DE ANÁLISE E PROJETO DE SISTEMAS COM UML. 3 ED. RIO DE JANEIRO: ELSEVIER, 2015.

PRESSMAN, ROGER; MAXIM, BRUCE. ENGENHARIA DE SOFTWARE. 8 ED. SÃO PAULO: MCGRAW HILL BRASIL, 2016.

SOMMERVILLE, IAN. ENGENHARIA DE SOFTWARE. 10 ED. SÃO PAULO: PEARSON BRASIL, 2019.

COMPLEMENTAR:

LARMAN, Craig. Utilizando UML e padrões. 3 ed. Porto Alegre: Bookman, 2007.

REZENDE, Denis Alcides. Engenharia de software e sistemas de informação. 3 ed. Rio de Janeiro: Brasport, 2005

■ Referências

[HTTPS://PM2ALL.BLOGSPOT.COM/2015/05/A-CRIE-NO-DESENVOLVIMENTO-DE-SOFTWARE.HTML](https://pm2all.blogspot.com/2015/05/a-crie-no-desenvolvimento-de-software.html) DATA DE ACESSO: 01/07/2015.

[HTTP://WWW.ALEXANDRE.ELETRICA.UFU.BR/ESOF/AULA01.PDF](http://www.alexandre.eletrica.ufu.br/esof/aula01.pdf) DATA DE ACESSO: 01/07/2023

[HTTP://WWW.ALEXANDRE.ELETRICA.UFU.BR/ESOF/AULA02.PDF](http://www.alexandre.eletrica.ufu.br/esof/aula02.pdf) DATA DE ACESSO: 01/07/2023

[HTTPS://WWW.DIO.ME/ARTICLES/DESCUBRA-O-QUE-E-A-CRIE-DO-SOFTWARE](https://www.dio.me/articles/Descubra-o-que-e-a-crie-do-software) DATA DE ACESSO: 05/12/2023

[HTTPS://CELSOKITAMURA.COM.BR/A-CRIE-DO-SOFTWARE-O-INICIO-DA-ENGENHARIA-DE-SOFTWARE/](https://celsokitamura.com.br/a-crie-do-software-o-inicio-da-engenharia-de-software/) DATA DE ACESSO: 05/12/2023

BASEADO NOS SLIDES DE GLAUCO TODESCO, TIAGO VINÍCIUS E GLEIBSON RODRIGO, JAIR C LEITE, ALEXANDRE VASCONCELOS, DANIEL VILLELA, MARIANA VIEIRA.

[HTTPS://SILO.TIPS/DOWNLOAD/ENGENHARIA-DE-SOFTWARE-01-INTRODUAO-MARCIO-DANIEL-PUNTEL](https://siilo.tips/download/engenharia-de-software-01-introducao-marcio-daniel-punzel) DATA DE ACESSO: 05/12/2023

KOTONYA, GERALD; SOMMERVILLE, IAN; REQUERIMENTS ENGINEERING: PROCESSES AND TECHNIQUES . JOHN WILEY & SONS, 1998.

■ Referências

- PRESMMAN,R.S. *ENGENHARIA DE SOFTWARE* 6A.EDIÇÃO, EDITORA MCGRAW-HILL,2006
- SOMMERVILLE,I. *ENGENHARIA DE SOFTWARE*, 8A. EDIÇÃO,EDITORA:ADDISON-WESLEY,2007
- BASEADO NOS SLIDES DE GLAUCO TODESCO, TIAGO VINÍCIUS E GLEIBSON RODRIGO,JAIR C LEITE, ALEXANDRE VASCONCELOS, DANIEL VILLELA.
- [HTTP://WWW.DEVMEDIA.COM.BR/ENGENHARIA-DE-SOFTWARE-2-TECNICAS-PARA-LEVANTAMENTO-DE-REQUISITOS/915](http://www.devmedia.com.br/engenharia-de-software-2-tecnicas-para-levantamento-de-requisitos/915). DATA DE ACESSO: 23/07/2014
- [HTTP://REDEETEC.MEC.GOV.BR/IMAGES/STORIES/PDF/EIXO_INFOR_COMUN/TEC_INF/081112_ANAL_SIS T.PDF](http://redeetec.mec.gov.br/images/stories/pdf/eixo_infor_comun/tec_inf/081112_anal_sist.pdf) DATA DE ACESSO: 23/07/2014
- [HTTP://WWW.LINHADECODIGO.COM.BR/ARTIGO/916/ESTRATEGIA-PARA-ENTREVISTAS-LTI-LEVANTAMENTO-TECNICO-INICIAL.ASPX#IXZZ3KBOPC6N0](http://www.linha decoding.com.br/artigo/916/estrategia-para-entrevistas-lti-levantamento-tecnico-inicial.aspx#IXZZ3KBOPC6N0) DATA DE ACESSO: 23/07/2014



OBRIGADO