



5407 Sistemas de informação – fundamentos

Sessão 3

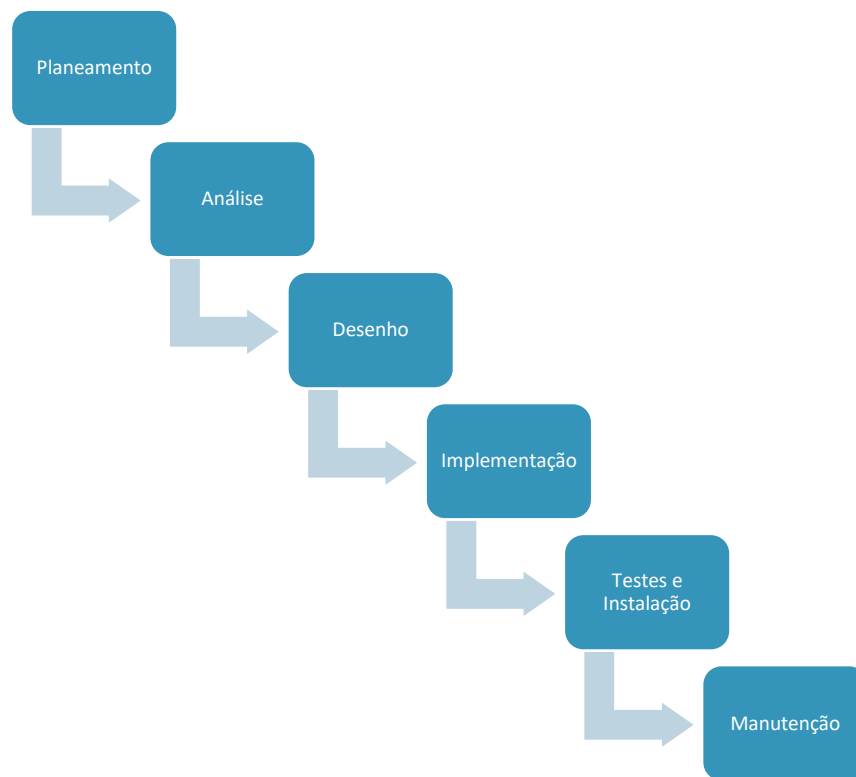
06/12/2023

Margarida Fachada



Fases do desenvolvimento dos sistemas de informação

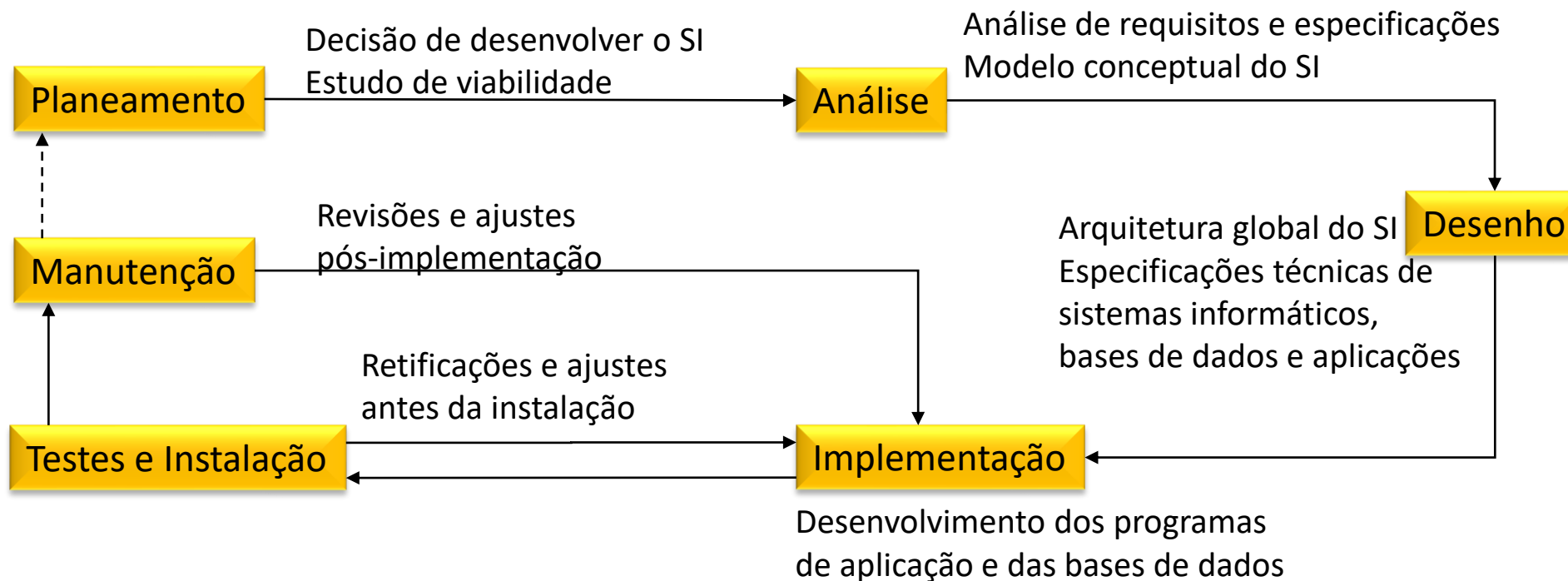
- O desenvolvimento de um sistema de informação desenrola-se através de várias fases distintas.



Fases do desenvolvimento dos sistemas de informação

- Estas fases podem ser vistas como etapas sucessiva, desde um ponto inicial, em que é decidido criar o sistema, até ao momento final, em que o sistema está pronto a ser usado.
- Cada vez mais um Sistema de Informação é visto como algo que surge e evolui através de um ciclo de vida.
- Neste ponto de visto, o desenvolvimento de um sistema de informação não termina quando está pronto a ser utilizado na organização.
- O ciclo de vida continua com a sua utilização e manutenção, mas também com a perspetiva do seu melhoramento e renovação.

Fases do desenvolvimento dos sistemas de informação



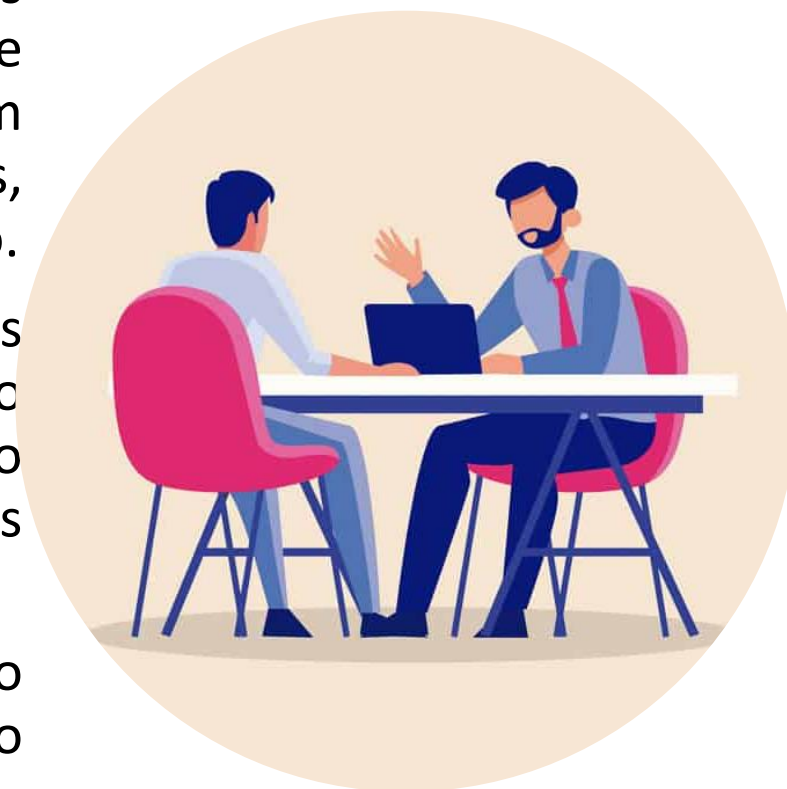
Planeamento

- Corresponde à tomada de decisão de construir o sistema de informação ou renová-lo, após terem identificado as necessidades gerais da organização e as possíveis soluções.
- É costume considerar nesta fase o estudo de viabilidade, ou seja, uma análise dos custos e benefícios em relação ao projeto.



Análise

- Fazer um levantamento de requisitos, ou seja, identificar as necessidades e restrições da organização em termos de sistema de informação, para isso os analistas fazem entrevistas ou questionários aos futuros utilizadores, analisam documentação e observam processos de trabalho.
- Definir as especificações do sistema, ou seja, estabelecer as indicações técnicas de como o sistema deverá ser pensado para ir ao encontro dos requisitos analisados no ponto anterior; para isso, os analistas utilizam ferramentas técnicas, como UML.
- Desta fase deve resultar o chamado Modelo Conceptual do sistema – que consiste numa descrição técnica da solução que é necessária à organização em termos de SI.



Fonte: <https://www.catho.com.br/carreira-sucesso/15-respostas-que-os-recrutadores-nao-gostam-de-ouvir-na-entrevista/>

Desenho

- Os analistas de sistemas elaboram a arquitetura global do sistema a implementar, com as indicações técnicas detalhadas para os programadores saberem como devem desenvolver os programas necessários.
- Estas indicações costumam ser referidas como o modelo lógico do sistema.
- Nesta fase trata-se de especificar “como” é que o sistema deve fazer o que tem de fazer.
- Os analistas identificam e avaliam as possíveis soluções técnicas ao nível das TI (hardware e software) que deverão ser incorporadas no SI.

Desenho

- Deve ser especificado que tipo de sistema informático vai ser necessário (tipo de computadores, armazenamento de dados, redes informáticas, telecomunicações, etc.)
- Devem ser feitas as especificações técnicas quanto ao sistema de gestão de base de dados (SGBD) a utilizar, esquemas ou estrutura da base de dados e programas de aplicação a desenvolver.

Desenho

- Tendo em conta que um SI assenta em grande parte na utilização de base de dados, na fase de desenho devem ser “desenhados” os seguintes elementos:
 - Desenho de modelos lógicos e físicos das bases de dados (a estrutura da base de dados em termos de tabelas e relações entre elas).
 - Desenho de esquemas lógicos (fluxogramas, algoritmos) dos processos e fluxos de dados existentes no sistema (que vão dar origem aos programas)
 - Desenho de interfaces (menus, formulários) dos programas a desenvolver.

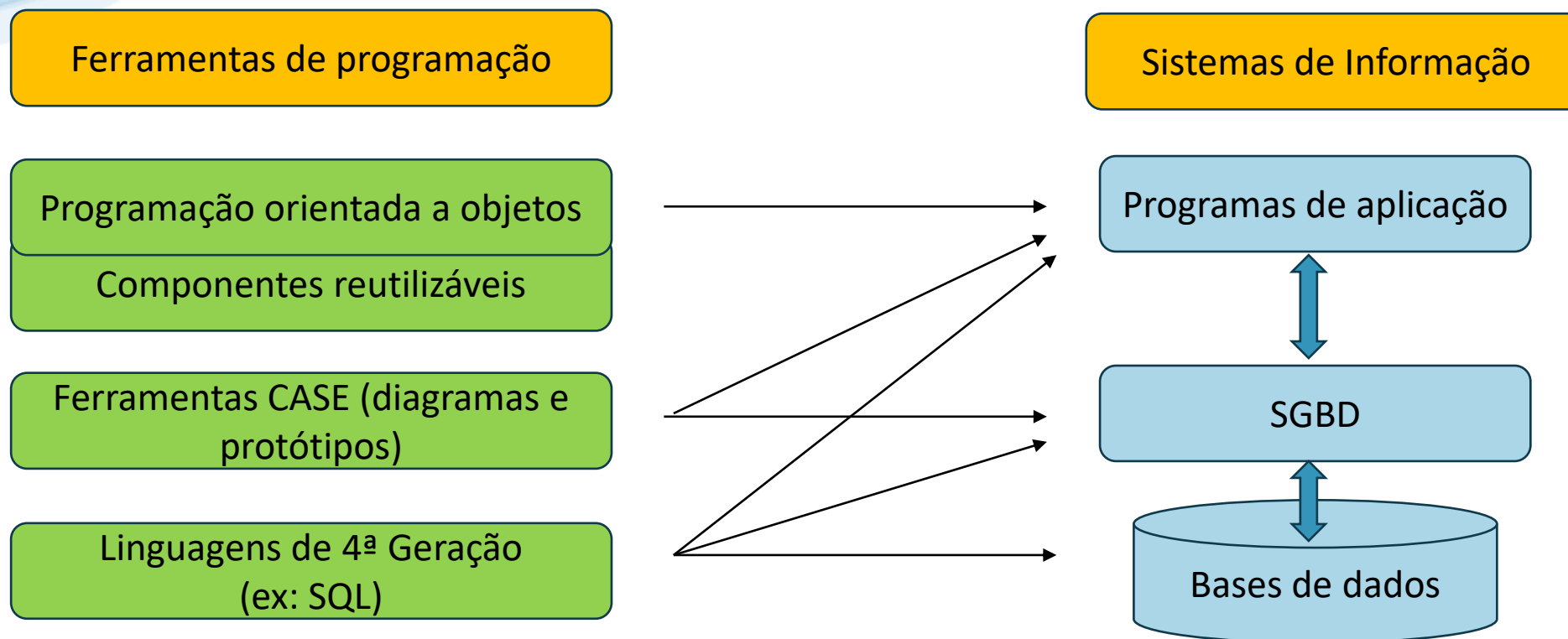
Implementação

- Passa por 2 tipos de tarefas distintas:
 - Criação do esquema físico da base de dados (com base no modelo lógico)
 - Codificação de programas de aplicação para acesso e manipulação de base de dados.
- Alguns conceitos no domínio da TI/SI que se referem a tecnologias e ferramentas mais utilizadas por analistas e programadores:
 - Programação orientada a objetos (POO)
 - Criação e reutilização de componentes
 - Linguagens de programação de 4ª geração
 - Ferramentas CASE

Implementação

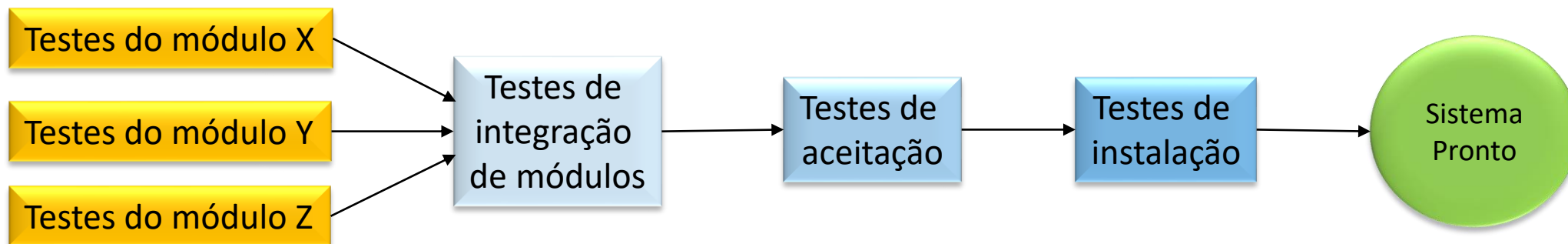
- Programação orientada a objetos (POO):
 - Um dos principais paradigmas de programação (com a linguagem C++, Java, entre outras).
 - Permite a criação e reutilização de componentes de uns sistemas para outros, de umas aplicações para outras.
- As linguagens de 4ª geração (4GL):
 - Possibilitam aos utilizadores interagirem com os sistemas de bases de dados (SGBD) obtendo de forma rápida, consultas, relatórios, etc. (por exemplo: a linguagem SQL).
- Ferramentas CASE:
 - Permitem aos analistas e programadores desenharem diagramas e esquemas das bases de dados, bem como criara aplicações de forma rápida.

Tipos de Tecnologias e ferramentas de desenvolvimento de SI



Testes e Instalação

- O Sw deve ser dividido em diferentes módulos. Por ex: um modulo pode gerir as encomendas; outro para gerir os produtos de um armazém; etc.
- Fase de testes pode ser dividida em:
 - **Testes ao nível do módulo** – em que os programadores poem à prova as funcionalidades pretendidas para essa parte do sistema;
 - **Testes de integração** – em que os módulos são integrados num todo e este é submetido a novos testes.



Testes e instalação

- Para além destes testes, devem ser feitos **testes de aceitação** - em que as aplicações são avaliadas pelos seus utilizadores finais, para ver se correspondem às funcionalidades desejadas e se existem retificações a fazer.
- Quando o sistema está pronto a operar na organização, devem ainda ser efetuados novos testes – **Testes de instalação**.
- A instalação implica:
 - Colocação em funcionamento de vários componentes do sistema, tanto ao nível de hardware como de software (SGBD, bases de dados e aplicações);
 - Configurações e parametrizações, por ex: ao nível dos servidores das bases de dados, direitos e restrições dos utilizadores, etc.
 - Distribuição de documentação de apoio
 - Formação de utilizadores (se necessário).

Manutenção

- Inclui várias tarefas:
 - Apoio aos utilizadores na resolução de problemas surgidos
 - Introdução de grandes alterações
 - Retificação de deficiências detetadas
 - Acrescento de novas funções.
- Tipos de Manutenção:
 - **Manutenção corretiva** – introdução de retificações em relação a erros detetados.
 - **Manutenção perfeita ou preventiva** – introdução de alterações para melhorar o desempenho do sistema ou evitar problemas previsíveis (por ex: aumento do volume de dados/ informação a processar);

Manutenção

- **Manutenção adaptativa** – introdução de alterações do sistema exigidas por entidades externas (como o Estado, os clientes ou fornecedores) ou impostas a partir da própria organização (por ex:
 - ✓ aquisição de novos equipamentos,
 - ✓ mudança do SGBD utilizado,
 - ✓ exigência de aceder a bases de dados via internet,
 - ✓ etc.)



Tarefa 6

Introdução ao UML

- O desenvolvimento tecnológico veio permitir que toda a informação possa ser suportada em computadores. Assim, ao nível das organizações, o sistema de informação tende a ter um suporte informático cada vez mais significativo.
- As aplicações informáticas modernas tendem a ser cada vez mais flexíveis, mas não estão preparadas para satisfazer todas as necessidades de informação dos seus potenciais utilizadores.
- Torna-se necessário podermos recorrer a uma linguagem que facilite a comunicação entre aqueles que tem de lidar com o negocio e com a Informática: atuais e potenciais utilizadores que definem as suas necessidades, gestores que avaliam se os sistemas informáticos satisfazem essas necessidades e informáticos que desenvolvam as funcionalidades pretendidas.

Introdução ao UML

- UML – Unified Modelling Language abre perspectivas para responder ao desafio de desenvolvimento de novos sistemas de informação, cada vez mais complexos, fiáveis e ajustados às necessidades dos utilizadores.
- UML pode ser traduzido por Linguagem de Modelação Unificada.
- Linguagem que utiliza uma notação padrão para especificar, construir, visualizar e documentar sistemas de informação orientados por objetos.
- Funciona como um meio de comunicação entre os diversos elementos envolvidos no processo: utilizadores, gestores e equipa de desenvolvimento.



Introdução ao UML

- Pode ser utilizada para documentar o sistema ao longo de todo o ciclo de desenvolvimento, começando com a tarefa inicial de análise de processos de negocio da organização e prolongando-se até à tarefa de manutenção evolutiva do sistema informático.

Diagrama de Casos de Uso (Use Cases)

- Os casos de uso representam o levantamento de requisitos de um sistema.
- O requisito num sistema é uma funcionalidade ou característica considerada relevante na ótica do utilizador.
- Normalmente representa o comportamento esperado do sistema, que na pratica consiste num serviço que deve ser disponibilizado a um utilizador.

Diagrama de Casos de Uso (Use Cases)

- Os requisitos podem ser classificados em três categorias:
 - **Requisitos Funcionais:** Descrevem o que um sistema faz ou é esperado que faça. Serão levantados, abrangendo a descrição de processamentos a efetuar pelo sistema, entradas (inputs) e saídas (outputs) de informação em papel ou ecrã que derivam da interação com pessoas e outros sistemas.
 - **Requisitos não funcionais:** estão relacionados com as características qualitativas do sistema, descrevendo a qualidade com que o sistema devesse fornecer os requisitos funcionais.

Diagrama de Casos de Uso (Use Cases)

- Requisitos não funcionais abrangem medidas de desempenho como:
 - Tempos de resposta,
 - volume de dados,
 - Considerações de segurança
- **Requisitos de facilidade de utilização:** garantem que existirá uma boa ligação entre o sistema desenvolvido, utilizadores do sistema e também as tarefas que desempenham apoiados pelos sistema. Por exemplo: menus especiais para um ecrã tátil.

Diagrama de Casos de Uso (Use Cases)

- São utilizados para a apresentação de requisitos e para assegurar que tanto o utilizador final como o perito numa determinada área ou o especialista informático possuem um entendimento comum dos requisitos.
- **Objetivo:** mostrar o que um sistema deve efetuar e não como o vai fazer.
- Estes diagramas utilizam as seguintes abstrações de modelação:
 - Atores
 - Casos de uso
 - Relações (include, extend e generalização)

Diagrama de Casos de Uso - Atores

- Um ator representa uma entidade externa que interage com o sistema.
- Apesar da representação humanizada, os atores podem não ser só pessoas, mas também outros sistemas físicos ou lógicos como, por exemplo, um modulo de contabilidade.
- Em geral, um ator pode invocar vários casos de uso e um caso de uso pode ser invocado por vários atores.
- Os atores devem ser caracterizados através de uma pequena descrição, de forma a assegurar uma correta compreensão do significado do ator por todos os elementos da equipa envolvida na análise.

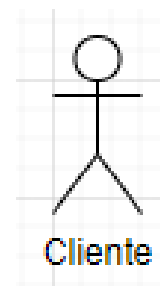
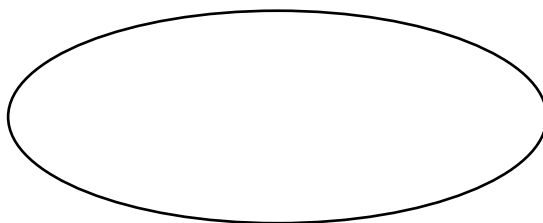


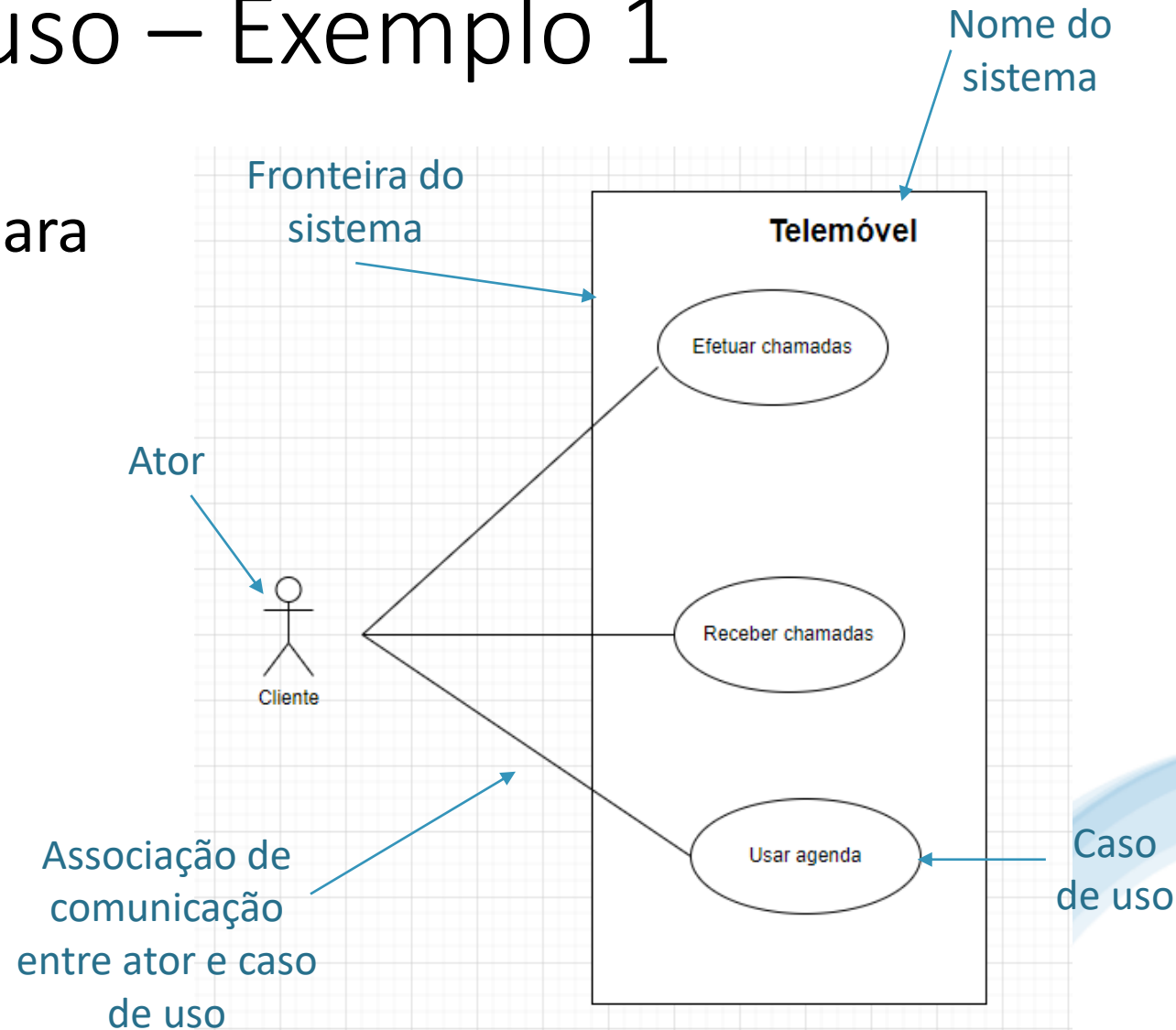
Diagrama de Casos de Uso – Casos de uso

- Simplificando...
 - Um caso de uso pode ser uma funcionalidade do sistema vista pelos utilizadores; um tipo de interação entre os atores e o sistema
- Como identificar os casos de uso?
 - Pensar em cada ator e nas interações que tem com o sistema
 - Um caso de uso agrupa interações elementares de atores com elementos da interface do sistema.



Diagramas de casos de uso – Exemplo 1

- Um utilizador usa o seu telemóvel para efetuar 3 operações:
 - efetuar chamadas,
 - receber chamadas
 - e usar a agenda.



Diagramas de casos de uso – Exemplo 2

- Exemplo um restaurante, onde se pretende representar 3 ações:
 - Pedido de Reserva da mesa
 - Servir almoço
 - Emitir fatura

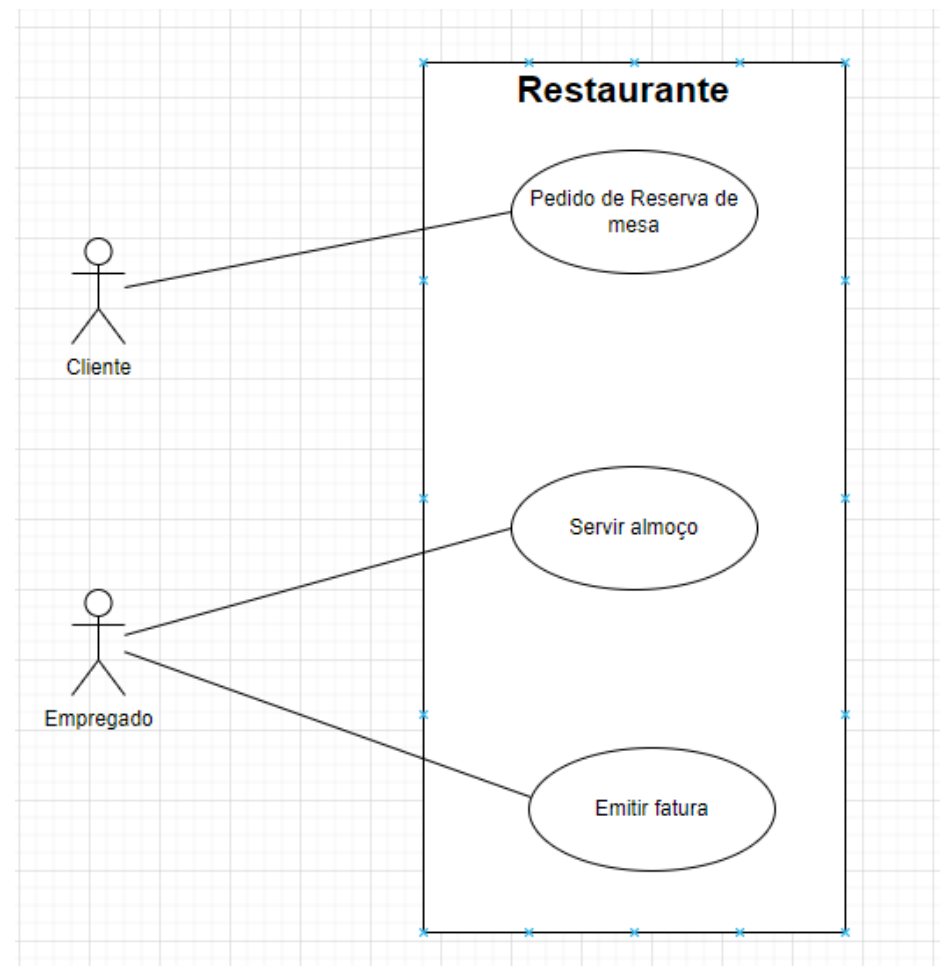


Diagrama de Casos de Uso (Use Cases) - Relacionamentos entre casos de uso

- Os casos de uso podem ser relacionados entre si. Mais frequentes são:
 - “include”
 - “extend”
 - Generalização
- Pode-se ter casos em que a execução de um caso de uso implique na execução de um outro
- Ou um caso de uso possui uma parte que se repete em outros casos de uso.
- Para evitar redundância de texto, pode-se isolar essas partes em casos de uso separados, e relacioná-los uns aos outros

Diagrama de Casos de Uso (Use Cases) - Relacionamento <include>

- Uso principal: Quando um caso de uso possui um comportamento parcial comum a vários outros casos de uso
 - Evitar repetir comportamento ->reutilização
 - A inclusão do outro caso de uso é obrigatória

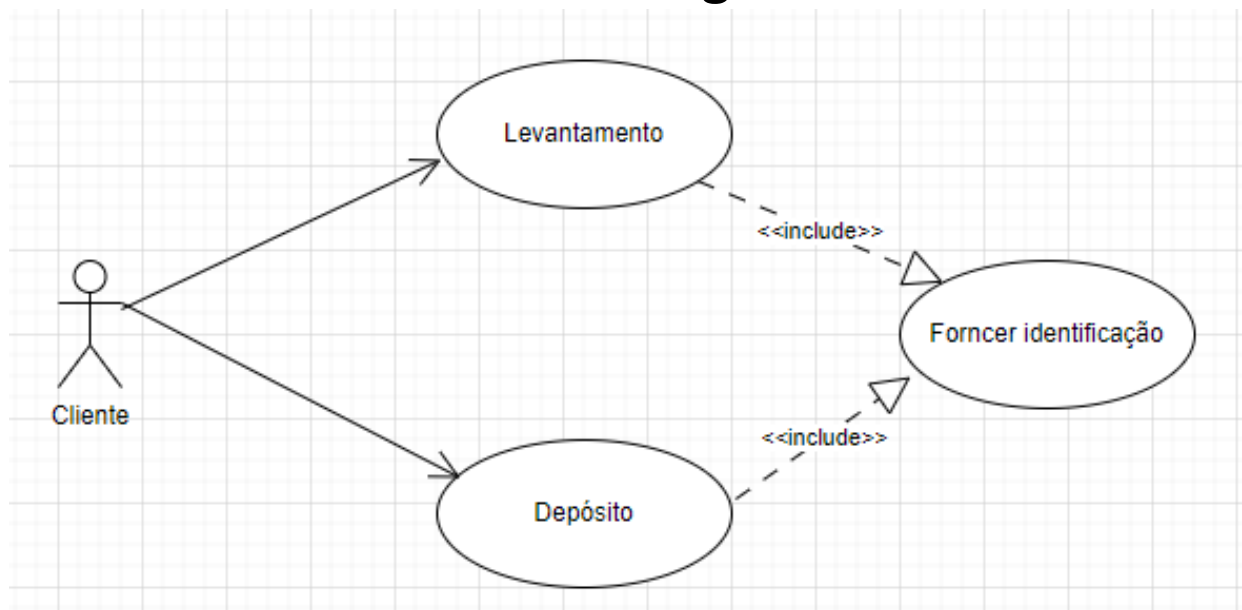


Diagrama de Casos de Uso (Use Cases) - Relacionamento <extend>

- Um caso de uso estende outro se ele adiciona comportamento ao caso de uso base.
 - Quando um fluxo alternativo é complexo e merece maior detalhe, pode-se escrever na forma de uma extensão ao caso de uso base.

Ou

- Quando uma modificação é necessária num caso de uso e não se quer mexer no caso de uso base

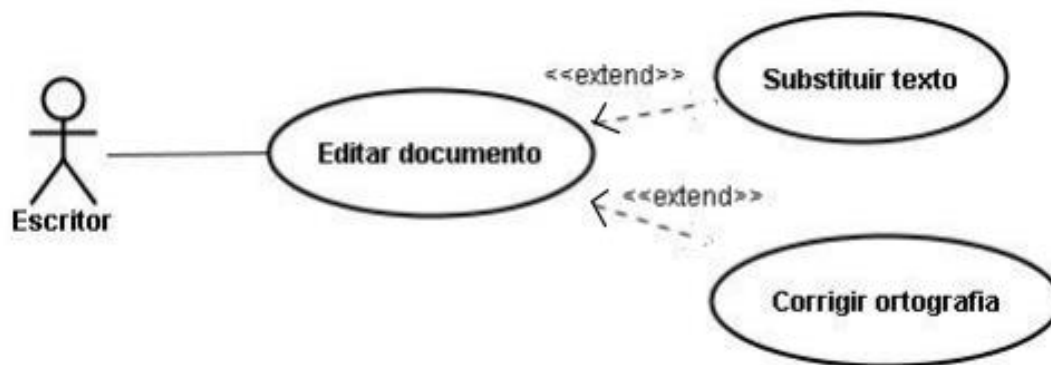


Diagrama de Casos de Uso (Use Cases) - Relacionamento Generalização

- Associação entre Casos de Uso onde dois ou mais deles possuem características semelhantes
- Nesse caso um Caso de Uso será o GERAL
- O Caso de Uso geral descreve as características compartilhadas
- As especializações definem características específicas

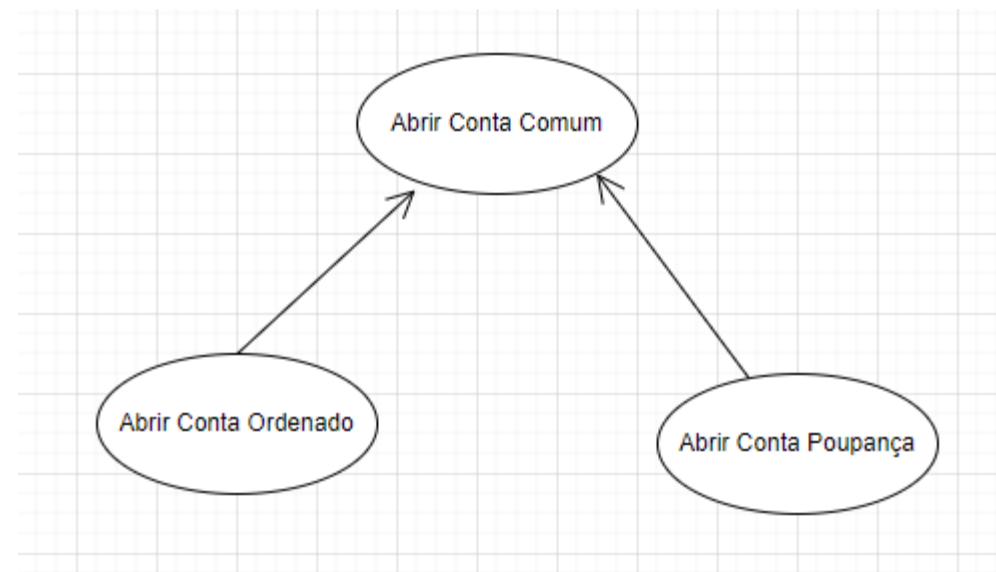


Diagrama de Casos de Uso (Use Cases) – Exemplo 3

- Pretende-se desenvolver um sistema de informação de gestão para um grupo de pizzarias, PhonePizza, que permita aos clientes efetuar encomendas na loja e através da Internet.
- Na loja, o cliente dirige-se ao empregado de balcão que introduzirá no sistema a encomenda pretendida.
- Caso a encomenda seja efetuada através da Internet, o cliente terá que se identificar, através do seu nome de utilizador e palavra-passe (controlo de acesso).
- O cliente pode então registar os artigos que pretende encomendar, podendo usufruir de um desconto no item, caso este esteja em promoção.
- O sistema deve ainda permitir que o Gestor da Pizzaria efetue as reservas de mesa, verificando se este tem autorização para o efetuar.

Diagrama de Casos de Uso (Use Cases) – Exemplo 3

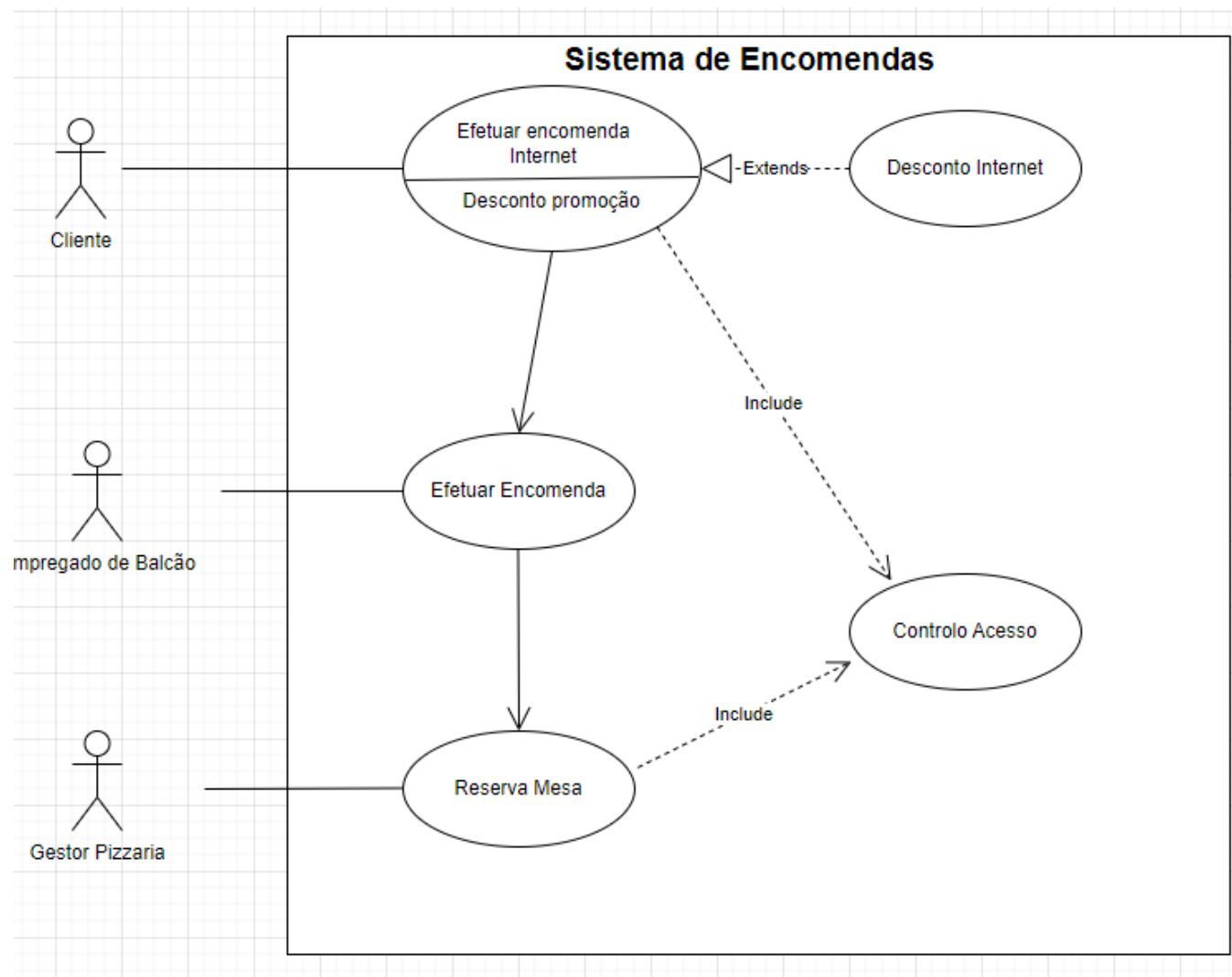
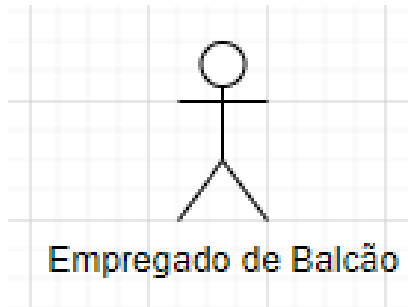
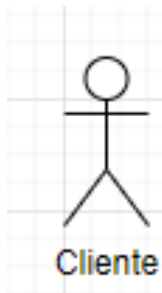


Diagrama de Casos de Uso (Use Cases) – Exemplo 3 - Resolução

- Identificação dos atores:



- Os atores Cliente e Gestor da Pizzaria são as pessoas que interagem com o sistema.

Diagrama de Casos de Uso (Use Cases) – Exemplo 3 - Resolução

- Descrição dos atores do exerc. anterior:
- Cliente – uma pessoa que encomenda produtos da PhonePizza pela Internet e nas pizzarias.
- Empregado de Balcão – empregado que recebe as encomendas ao balcão da pizzeria
- Gestor da Pizzaria – empregado que está encarregue de efetuar as reservas de mesa numa pizzaria.



Diagrama de Casos de Uso (Use Cases) – Exemplo 3 - Resolução

Ator	Casos de uso
Cliente	<ul style="list-style-type: none">• Efetuar encomenda Internet• Controlo de acesso
Empregado de Balcão	<ul style="list-style-type: none">• Efetuar encomenda• Controlo de acesso
Gestor da Pizzaria	<ul style="list-style-type: none">• Reservar Mesa• Controlo de acesso

Diagrama de Casos de Uso (Use Cases) – Exemplo 3 – Resolução - Comunicação

- A comunicação entre um ator e os casos de uso pode ser representada por uma simples linha reta ou uma seta cujas pontas indicam a direção da comunicação.
- Linha reta simples – os atores podem estar colocados em qualquer ponto do diagrama, com o pressuposto que existirá alguma comunicação de emissão ou receção
- Seta unidirecional – a seta indica o sentido preferencial da comunicação. Normalmente, neste caso é habitual a colocação dos atores emissores à esquerda da fronteira do sistema, e dos atores recetores à direita.

Diagrama de Casos de Uso (Use Cases) – Exemplo 3 – Resolução - Tempo

- Na identificação dos casos de uso parte-se do princípio que todos são originados pelos atores.
- Contudo, em alguns sistemas existem casos de uso despoletados, automaticamente, de acordo com um processo temporal cíclico, onde num determinado momento no tempo o caso de uso é executado.
- Por ex: No caso de estudo, poderia existir a necessidade de efetuar uma copia periódica dos dados das encomendas o envio mensal das promoções aos clientes registados.

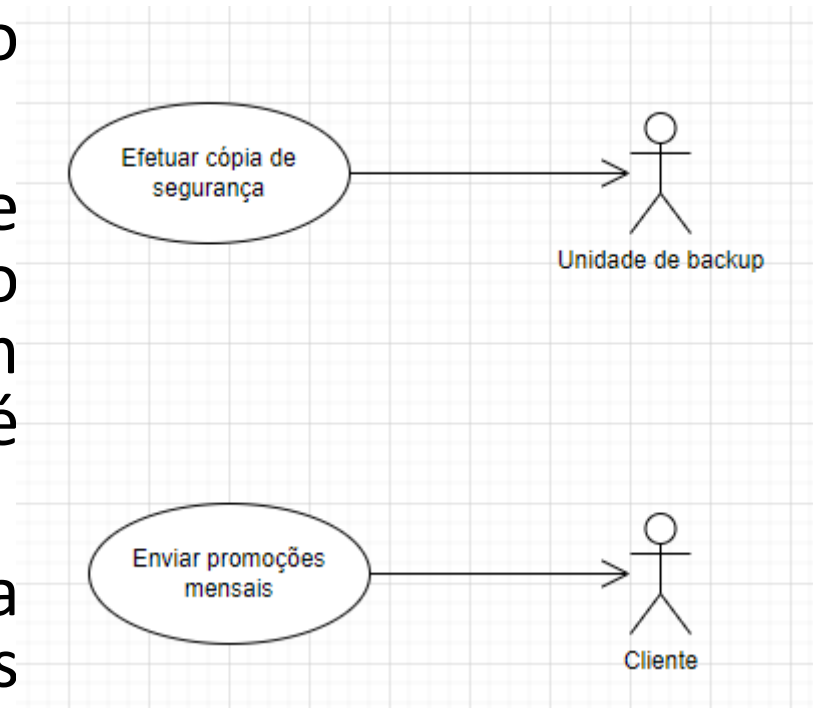


Diagrama de Casos de Uso (Use Cases) – Exemplo 3 – Resolução – Relações

- No caso de uso **Efetuar encomenda Internet** estamos a considerar um pressuposto na pré-condição que o cliente era um utilizador válido no sistema, ou seja, já tinha passado pelo controlo de acesso.
- Caso não existisse esse pressuposto, a relação <<include>> também teria que ser incluída na descrição.

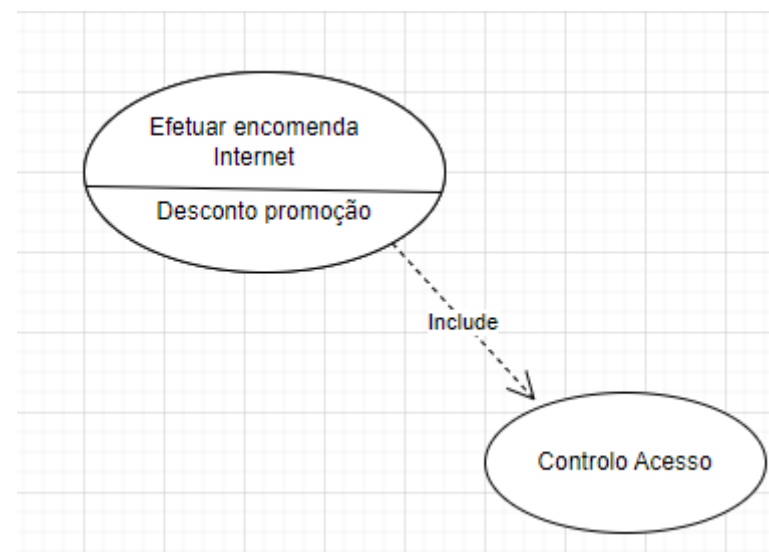


Diagrama de Casos de Uso (Use Cases) – Exemplo 3 – Resolução – Relações

- A relação <<extends>> ocorre quando existe um comportamento opcional que deve ser incluído num caso de uso. Este comportamento é definido num 2º caso de uso e invocado pelo caso de uso base através de um mecanismo de ponto de extensão.
- O mecanismo de pontos de extensão permite definir no caso de uso base onde o comportamento será incorporado, sem alterar a sua descrição. Também garante que o seu comportamento não seja alterado caso o “Desconto Internet” deixe de existir.

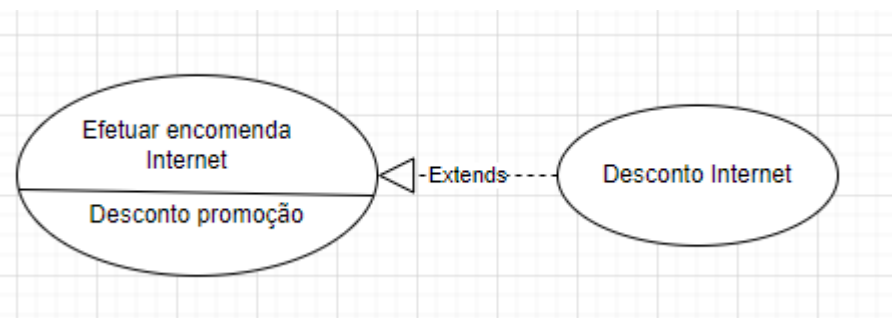
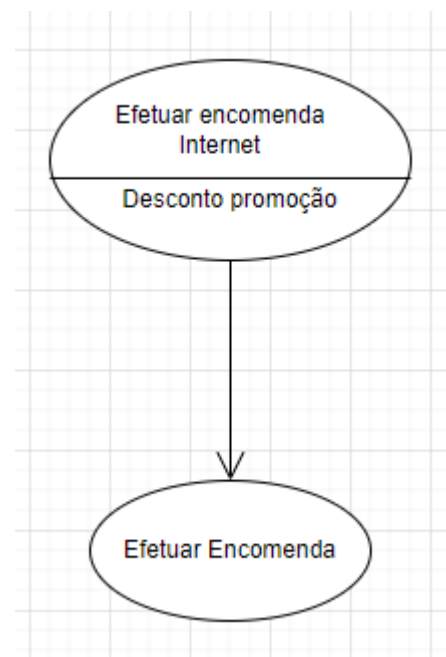


Diagrama de Casos de Uso (Use Cases) – Exemplo 3 – Resolução – Relações

- Na descrição do caso de uso “Efetuar Encomenda Internet” é definido um ponto de extensão “Desconto Promoção”.
- Desta forma o cliente ao adicionar um novo produto à sua encomenda obterá um desconto, caso o produto esteja em promoção.
- A relação de generalização é utilizada quando existe um caso de uso que é particular de um outro caso de uso. Por ex: O comportamento do caso de uso Efetuar Encomenda Internet é semelhante ao caso de uso Efetuar Encomenda, existindo apenas pequenas variações específicas do meio onde é efetuada a encomenda.





Tarefa 7