Análise de Sistemas e Base de Dados

Aula Teórica 01

Carlos Pereira

ESAN, Universidade de Aveiro

Fevereiro 2025

Cofinanciado por:
NORTE 2030 Programa Regional do Norte
PORTUGAL 2030
Cofinanciado pela União Europeia

Sumário

Sumário

Sumário da aula

1. Introdução à Análise de Sistemas
2. Implementação da Análise (UML)
3. Diagramas de Use Cases
4. Descrição Detalhada de Use Cases

Introdução à Análise de Sistemas

Análise de Sistemas

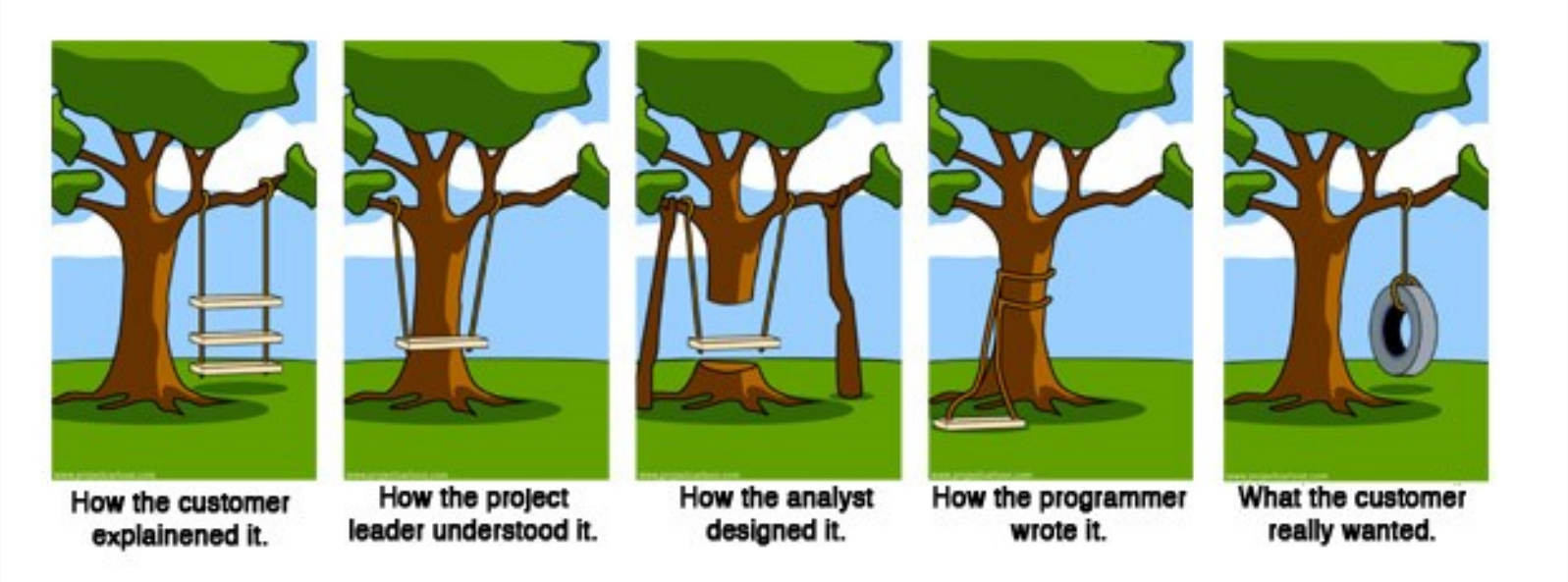
Ciclo de desenvolvimento de um sistema

- Análise
- Desenho de uma solução
- Implementação dessa solução
- Teste da solução

Motivação para a análise de sistemas

- A necessidade de conhecer um sistema - já existente, ou um sistema a desenvolver;
- A complexidade dos sistemas a analisar;
- A constatação de que o uso de abordagens ad-hoc, não sistematizadas, levam a falhanços estrondosos;
- A circunstância, mais ou menos generalizada, de que quem analisa os sistemas, quem tem de os planear, desenhar e implementar, normalmente não conhece os problemas que se pretende ajudar a resolver com esses sistemas.

Motivação para a análise de sistemas



Análise de Sistemas

O que é a análise de sistemas

- É um processo de identificação das necessidades de um sistema:
  - Requisitos funcionais;
  - Outro tipo de requisitos;
- Não é um processo de identificar a forma como o sistema será implementado;
- O resultado da análise deve ser capaz de comunicar de um modo completo e consistente o que é necessário a um sistema.

O desafio da análise

- Problema central: conhecimento do domínio do problema:
  - O analista tem de conhecer o domínio do problema e as responsabilidades do sistema que se pretende desenvolver no contexto desse domínio;
  - Só se pode resolver o que se conhece. Não é possível resolver um problema que se desconheça assim como as suas implicações;
  - O conhecimento do domínio do problema permite ao analista encontrar as melhores soluções, em termos de sistema a desenvolver, que muitas vezes mesmo os especialistas no domínio do problema não considerariam.

Domínio do problema:

- Quem quer que pretenda conceber um sistema para auxiliar o desempenho de determinadas tarefas tem de conhecer tudo o que está relacionado com esse desempenho:
  - Como é implementado usando os métodos "tradicionais" / existentes;
  - Quais são os problemas principais que lhe estão associados;
  - Quais são os aspetos onde se conhecem formas melhores de funcionamento e quais são essas formas (procura das soluções "ótimas" e viáveis);
  - O que se pretende, para um novo sistema, é que conserve, na medida do possível, o que há de melhor no funcionamento do sistema que o precede e que evite, desejavelmente, os problemas que evidencie.

Domínio do problema:

- Quem analisa, na maioria das vezes, tem de ter acesso a especialistas no domínio do problema, uma vez que desconhece, na maior parte dos casos, o domínio do problema;
- Mesmo que o analista conheça bem o problema, pode ter de partilhar esse conhecimento com os restantes analistas;
- Muitas vezes, na conclusão do processo de análise, o analista transformou-se num especialista do domínio do problema.

Desafio: comunicação como outro problema

- Universos com discurso diferente que têm de ser conciliados: analista e especialista no domínio do problema
- Esta comunicação é crucial:
  - O analista tem de obter o conhecimento do domínio do problema a partir da interacção com os especialistas;
  - O analista tem de comunicar ao especialista do domínio do problema o conhecimento que adquiriu, para validação;
  - Tem, igualmente, de lhe transmitir os requisitos do sistema, tal como ele os interpretou.

Resultados da análise de sistemas

- Formalizam as necessidades de quem "encomenda" ou necessita do sistema;
- Estabelecem uma lista de coisas para fazer, em termos do desenvolvimento do sistema;
- Traduzir-se-á, no nosso caso, num conjunto de modelos;
- Esses modelos serão a base do desenvolvimento do sistema e da verificação da adequação da respectiva implementação.

O que é um modelo?

- Um modelo representa abstractamente uma componente/vertente de um sistema;
- Captura os aspetos importantes daquilo que representa, enquanto omite aqueles menos relevantes;
- Esse modelo é, ele próprio, representado numa determinada forma;
- Os modelos que vamos desenvolver são representados usando a linguagem UML.

Para que serve um modelo?

- Para capturar e representar de modo preciso os requisitos de um sistema, de modo a que todos os interessados possam compreendê-los e concordar com eles;
- Para capturar e representar outras decisões sobre o sistema;
- Para melhor perceber o sistema a desenvolver, em toda a sua complexidade;
- Para gerir e "conquistar" a própria complexidade do sistema, uma vez que nos pode dar visões "parciais" desse sistema;
- Para explorar múltiplas soluções sem o custo do respectivo desenvolvimento.

Para que serve um modelo?

- Recapitulando, são em particular relevantes estes objetivos:
  - Visualizar o sistema como é ou como queremos que seja;
  - Especificar estrutura e comportamento;
  - Documentar decisões tomadas;
  - Comunicar todos estes aspetos a todos os interessados.
- Um modelo pode ser o produto final do processo de análise, mas pode também representar um passo intermédio;
- O modelo está em constante evolução, durante o processo de análise.

Conceitos de Modelação

- Domínio do problema: a fração da realidade que se pretende modelar;
- Sistema: aquilo se vai analisar;
- Modelo: uma abstração de um sistema;
  - É uma representação simplificada, auto consistente e completa do Domínio do Problema;
- Vista: uma projeção do sistema segundo um aspecto particular;
  - Um modelo contêm, normalmente, múltiplas vistas.

Implementação da Análise

UML - Uma definição breve

- A UML é uma linguagem de modelação visual, baseada em diagramas, que se tornou standard em 1997;
- Permite a criação de modelos para descrever sistemas dos mais diversos tipos, embora na origem da linguagem tenham estado os sistemas que incluem software;
- Cada modelo é constituído por um conjunto de diagramas, correspondentes a perspectivas ou pontos de vista específicos do modelo;
- Existem múltiplos diagramas standard em UML, por norma divididos em 2 tipos:
  - estáticos: ex.: diagramas de use case ou diagramas de classes;
  - dinâmicos: ex.: os diagramas de actividades ou de sequência.

Implementação da Análise

Que tipo de análise vamos fazer?

- Análise de um sistema do ponto de vista das funcionalidades dos seus utilizadores:
  - Utilizadores dos sistemas (atores);
  - Outros interessados nos sistemas (stakeholders);
  - Funcionalidades oferecidas aos utilizadores (use cases);
  - Descrição de aspetos mais relevantes da funcionalidade usando outros tipos de diagramas (diagramas de atividade).

Atores

- Um ator representa tudo o que interage com o sistema:
  - Pessoas, outros sistemas de software, dispositivos de hardware, etc;
  - Cada ator define um papel específico e coerente, em termos de interação com o sistema;
  - Uma única pessoa física pode ser representada, em termos da sua interação com o sistema, por mais de um ator (pessoa que desempenha mais um papel na sua interação com o sistema). Ex: um funcionário de uma empresa pode também ser um cliente;
  - Do mesmo modo, distintas pessoas físicas podem ser representadas por um único ator (ex: todos os funcionários são representados pelo mesmo ator).

Outros interessados (stakeholders)

- Nesta categoria enquadram-se todos os intervenientes que têm interesse na organização ou no sistema a desenvolver:
  - administração da organização;
  - entidades externas com interesse na existência/bom funcionamento do sistema;
  - clientes, etc...
- São relevantes, porque o sistema a desenvolver, ou os procedimentos a implementar, têm de proteger os interesses de cada um destes stakeholders;
- Só são relevantes e só interessam se for claro também o interesse associado, que o sistema terá de proteger.

Use Cases

- Um Use Case especifica o comportamento de um sistema ou parte de um sistema:
  - consiste numa descrição de um conjunto de sequências de ações, incluindo variantes, que o sistema desempenha para atingir um resultado observável com valor para um dado ator;
- Deste modo, os use cases permitem descrever as funcionalidades de um sistema, do ponto de vista do utilizador (e de acordo com a perspetiva usada para a análise).

Use Cases

- Um use case tem associados três aspetos essenciais:
  - Um ator;
  - Uma sequência de ações que especifica a interação entre o ator e o sistema;
  - Um ou mais interessados (stakeholders) no use case e o use case tem de proteger os interesses de todos estes interessados.

Descrição de use cases

- Quando iniciamos a análise de um sistema, a primeira aproximação em termos de use cases passa, normalmente:
  - Identificação dos use cases "principais";
  - Representação em diagramas de use case;
  - Descrição necessariamente abreviada de cada um desses use cases, usando notação exemplificada no slide seguinte.
- A descrição pode, depois, ser revista e detalhada com um maior nível de pormenor, numa fase posterior;
- A análise (e as restantes fases de desenvolvimento de um sistema) são feitas de modo iterativo.

Descrição Abreviada de Use Cases

- Cada use case tem de ter:
  - Um nome (escolhido de modo a que "sugira" o conteúdo do use case);
  - Uma descrição: numa fase inicial do processo, basta um parágrafo que descreva, sinteticamente, a função associada ao use case. Numa fase posterior, terá de haver um maior nível de detalhe.
- Ex: Nome: Fazer encomenda:
- Descrição: Permite ao utilizador fazer a encomenda de um ou mais produtos. O utilizador indicará os produtos e a respetiva quantidade, identificar-se-á perante o sistema, indicará o método de pagamento e concluirá a encomenda.

Implementação da Análise

Identificar atores

- Perguntas que se podem fazer:
  - Quem usa o sistema?
  - Que outros sistemas o utilizam?
  - Quem obtém informação a partir do sistema?
  - Quem fornece informação ao sistema?
  - Há alguma coisa que aconteça automaticamente num tempo bem definido?

Identificar outros interessados

- Perguntas que se podem fazer:
  - Quem tem interesse no sistema, de modo direto ou indireto?
  - Quem é afetado, direta ou indiretamente, pelo funcionamento do sistema?
  - Quem tem, para além dos atores concretos, interesse no resultado de uma determinada funcionalidade, expressa sob a forma de um use case.

Implementação da Análise

Identificar use cases

- Os use cases, naturalmente, descrevem as coisas que os atores pretendem que o sistema faça por eles
  - Que funções quererá cada ator do sistema?
  - O sistema armazena informação? Que atores consultam, criam, modificam ou apagam informação?
  - O sistema necessita de informar um ator sobre alterações do seu estado?
  - Há eventos externos dos quais o sistema necessita de ter conhecimento? Que ator informa o sistema disso?
- Principal Objectivo: tentar determinar todas as funcionalidades que o sistema tem de oferecer e que atores as utilizam!

Implementação da Análise

Identificar use cases

- Duas estratégias principais:
- Análise das operações de criação, consulta e alteração de informação pelas quais o ator é responsável;
- Análise das tarefas do ator e de procurarmos colocar-nos na sua "pele", para o desempenho dessas tarefas.

Diagramas de Use Cases

Uso de diagramas

- Os diagramas conferem um suporte gráfico à análise, que torna mais fácil a compreensão da visão do sistema, tal como o analista o identificou;
- Os diagramas devem ser organizados em vistas, em que cada vista apresenta uma parte do sistema global;
- Sugerem-se o uso de múltiplas vistas, nos modelos a desenvolver:
  - Por ator;
  - Por área funcional.

Diagrama Centrado no Ator

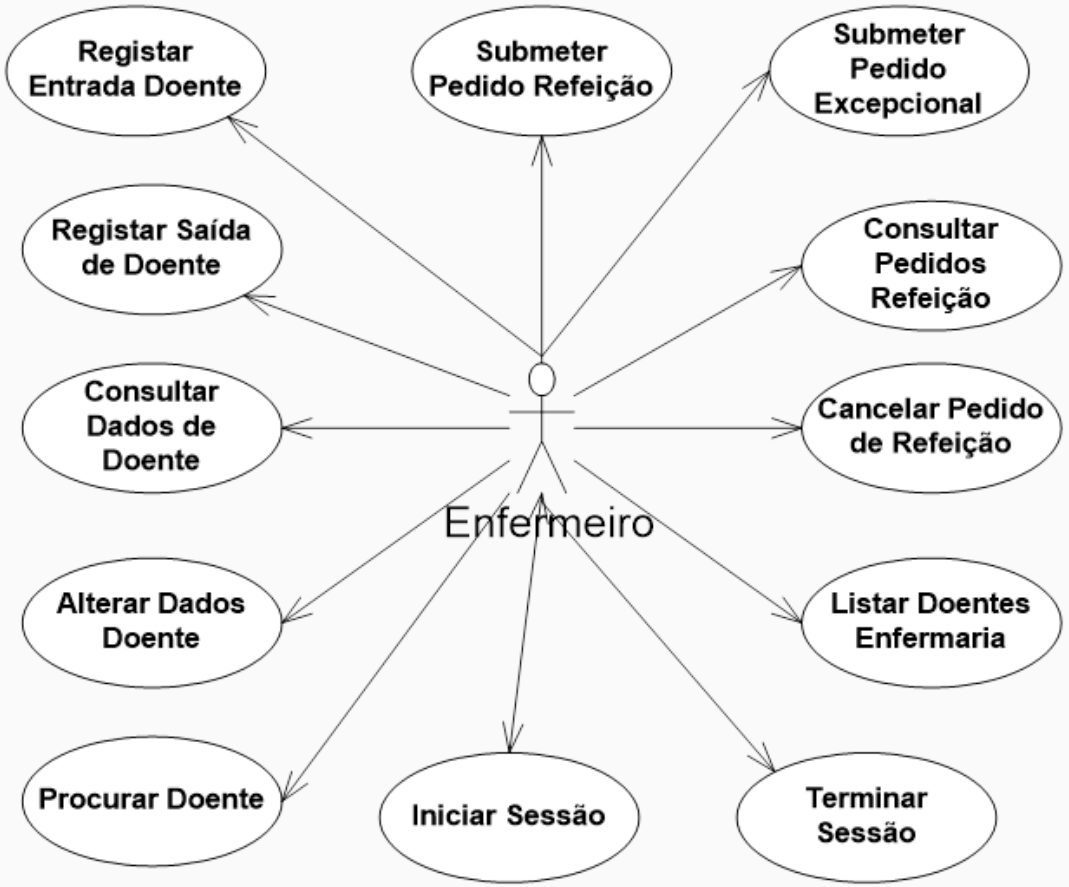
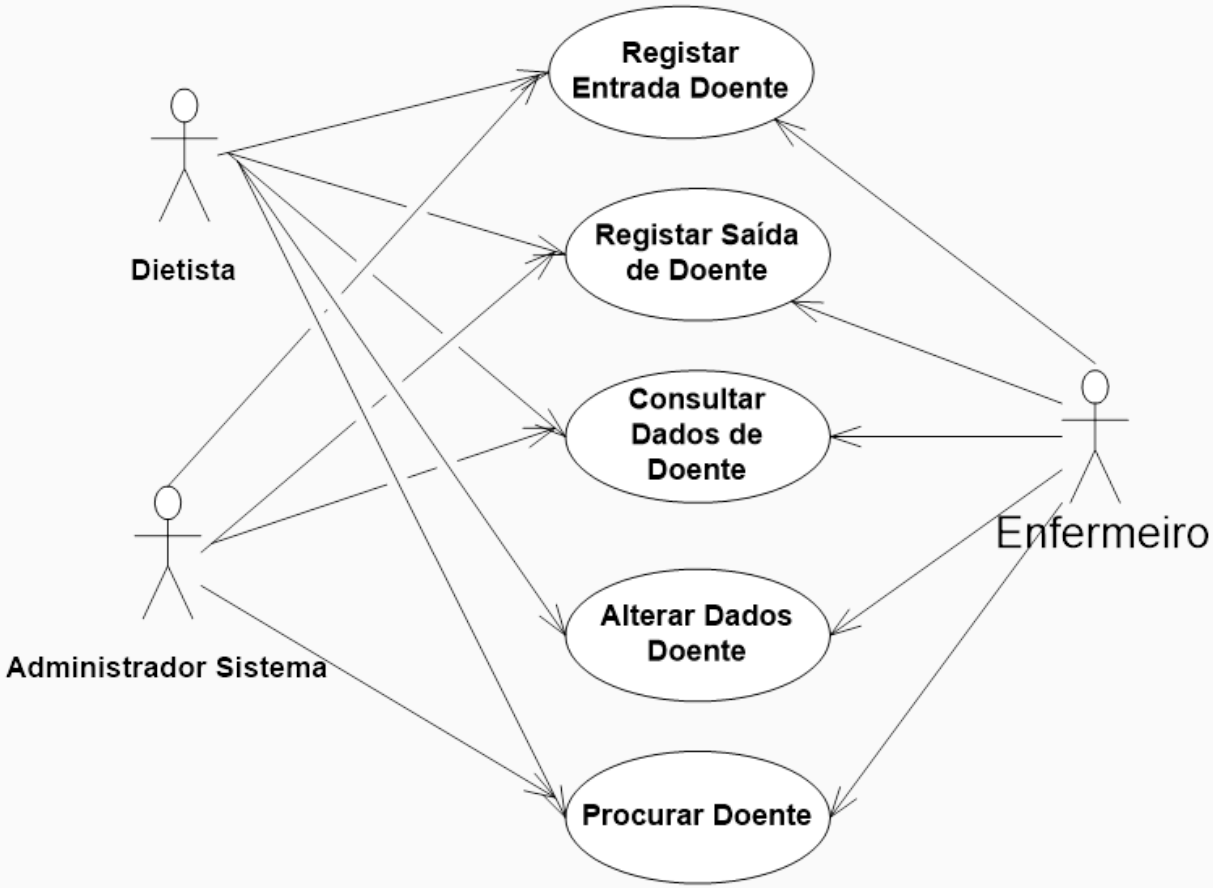


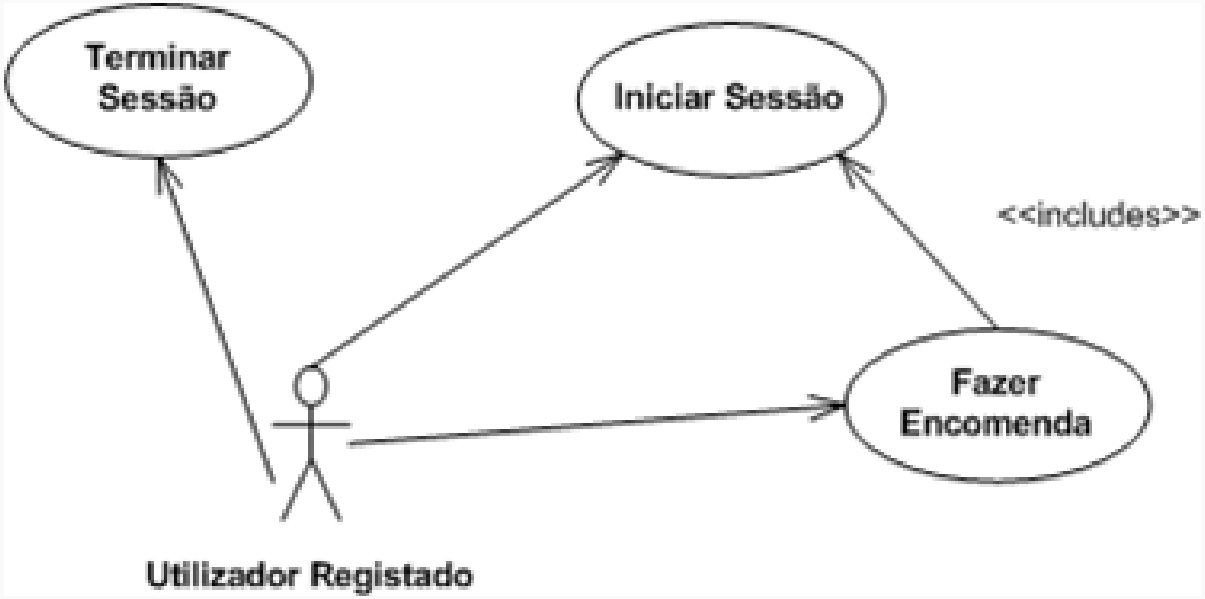
Diagrama Centrado numa Área Funcional



- Podemos ter uma vista por cada área ou, se isso resultar em vistas complexas, decompor uma única área em múltiplas vistas.

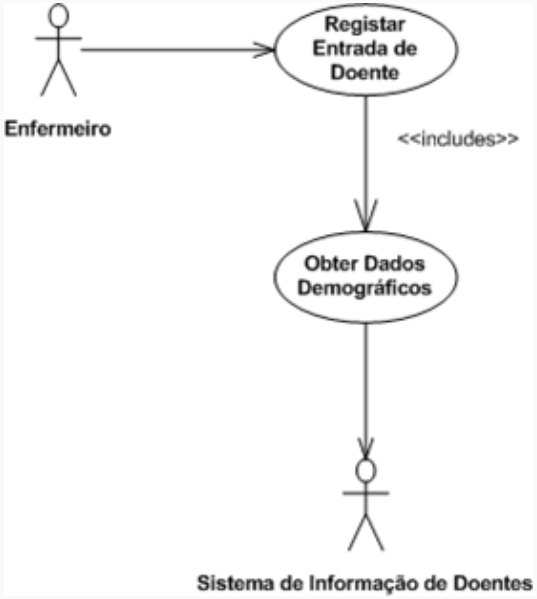
Informação num diagrama de use cases

- A direção da ligação indica quem inicia a interação:



Informação num diagrama de use cases

- Neste caso, temos um ator humano e outro que é um sistema:



Organização de Use Cases

- Podemos definir relações entre use cases:
- Inclusão:
  - O use case base incorpora obrigatoriamente outro use case durante o seu procedimento.
  - Designado por "include" no diagrama.
  - Exemplo: Checkout –include–> Login
- Extensão:
  - O use case base pode facultativamente incorporar o comportamento de outro use case.
  - O use case que estende não é por norma um use case por si só.
  - Designado por "extend" no diagrama.
  - Exemplo: Consultar Produtos <–extend– Imprimir

Descrição Detalhada de Use Cases

Descrição Detalhada de Use Cases

Caracterização de um Use Case

- Há um conjunto de informações que devem ser utilizados para caracterizar o use case:
  - Nome, descrição do use case, atores, outros interessados, prioridade, finalidade, pré e pós-condições, trigger
- Fluxo básico de acontecimentos:
  - Sequência de interações principal (cenário principal) - o objetivo é atingido com sucesso

Descrição Detalhada de Use Cases

Template a utilizar

Use Case	Nome do Use Case
Descrição Breve do Use Case	Uma descrição textual, sintética, do use case
Atores	Entidades externas ao sistema que participam na realização do Use Case
Outros Interessados	Uma lista dos outros interessados no use case (stakeholders) e do respectivo interesse
Prioridade	Prioridade para a implementação deste Use Case (Escala deverá ser definida previamente)
Finalidade	Objetivo do Use Case. Deverá representar um valor acrescentado para os atores
Pré-condições	Requisitos a verificar para que o Use Case se possa iniciar; têm de poder ser testados.
Pós-condições / garantias	O que o sistema assegura no caso de ter ocorrido o cenário de sucesso
Trigger	O evento que inicia o use case

Descrição Detalhada de Use Cases

Aspectos a considerar

- Outros interessados: indispensável referir o interesse; no percurso básico e/ou nos alternativos, mostrar como se protege o interesse
- Pré-condições: têm de poder ser testadas pelo sistema, antes do use case se iniciar;
- Pós-condições: são o que se garante que acontece, depois de terminado o use case com sucesso, num de dois aspetos:
  - Mudanças de estado no sistema;
  - Mudanças ao nível da informação mantida no sistema.

Descrição Detalhada de Use Cases

Exemplo:

Use Case	Registar entrada de doente
Descrição Breve do Use Case	Permite ao ator especificar informação relativa ao doente, nomeadamente localização e serviço responsável.
Atores	Enfermeiro, Administrador de Sistema
Outros Interessados	Médico – Tem interesse no registo dos doentes de modo a planificar o seu dia-a-dia.
Prioridade	Alta
Finalidade	Registo de toda a informação relevante relativa ao doente, incluindo localização e serviço responsável pela entrega das refeições.
Pré-condições	O ator deverá ter iniciado sessão no sistema.
Pós-condições	A informação relativa ao doente é submetida no sistema.
Trigger	O utilizador selecciona a opção 'Registar entrada de doente' da interface do sistema.