

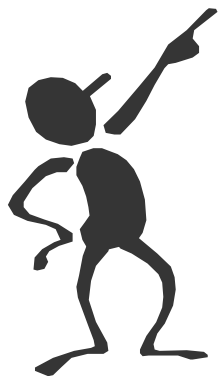


# SISTEMAS DE INFORMAÇÃO

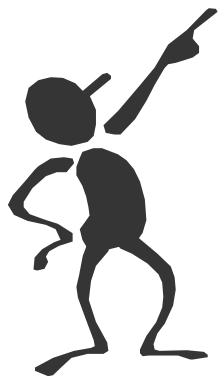
LEVANTAMENTO DE REQUISITOS

Helder Rodrigo Pinto

---



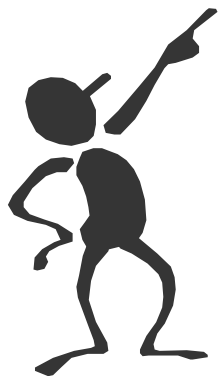
- Engenharia de Software
- Levantamento de Requisitos
- RUP
- UML
- Casos de Uso



- Engenharia de Software
- Levantamento de Requisitos
- RUP
- UML
- Casos de Uso

- Conjunto de métodos, ferramentas e procedimentos
  - Método:
    - Como fazer
  - Ferramentas:
    - Apoio automatizado
  - Procedimentos:
    - elo entre os métodos e ferramentas

- Representação abstracta de uma realidade
  - Desenho técnico
  - Maquete
  - Diagramas
- Representação escrita ou gráfica
- Procura eliminar a ambiguidade e redundância



- Engenharia de Software
- Levantamento de Requisitos
- RUP
- UML
- Casos de Uso

- Técnicas para garantir que os objectivos do sistema sejam cumpridos, de forma consistente.
- Conjunto estruturado de objectivos/actividades usadas para derivar, validar e manter o projecto.
- Essas actividades do processo incluem o levantamento/briefing, análise, negociação e validação.

- Leitura de documentos
- Observação
- Entrevistas
- Reuniões
- Questionários
- Engenharia Reversa (reengenharia) das aplicações existentes



- O que acontece quando os Requisitos estão errados?
  - o sistema pode ser entregue com atraso e com custo acima do esperado inicialmente;
  - o cliente e utilizador final não ficam satisfeitos com o sistema;
  - a utilização do sistema pode ser incerta, com erros de sistema regulares, provocando interrupções durante a operação normal;
  - se o sistema continua em uso, os custos de manutenção e evolução do sistema são normalmente muito altos.

- O que o sistema *deve* fazer
  - **Requisitos Evidentes:** explicitados pelo utilizador
  - **Requisitos Ocultos:** efectuados pelo sistema sem conhecimento explícito do utilizador

- Como o sistema deve realizar os requisitos funcionais e com que qualidade
  - **Requisitos Obrigatórios**
    - Requisitos incluídos de certeza
  - **Requisitos Desejáveis**
    - A inclusão depende de outros requisitos, custo, risco ou recurso
  - **Requisitos Temporários**
    - Permanecem apenas em algumas versões do programa
  - **Requisitos de qualidade**
    - Têm a ver com a usabilidade, confiabilidade, eficiência, configurabilidade, segurança, implementação, interface, etc

- Consequências de não restringir a complexidade
  - Quanto mais complexo é um sistema mais sujeito está a ruptura
  - Projectos terminados para além dos prazos e custos previstos
  - Projectos que não cumprem na íntegra os requisitos iniciais

- Supondo um briefing com um cliente que pretende a criação de uma aplicação mobile de gestão de *tickets* de atendimento num Centro de Saúde.
  - Requisitos Funcionais?
  - Requisitos Não Funcionais?

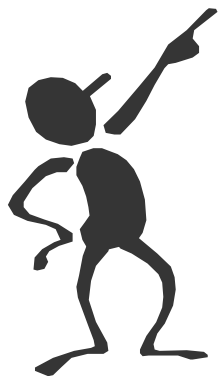
- Levantamento de requisitos funcionais:
  - Cada utilizador deve poder fazer registo, login e logout.
  - É permitido recuperar a password.
  - O Utilizador pode fazer o pedido de um *ticket* para três tipos de atendimento: consulta marcada, urgência, enfermagem.
  - É possível verificar, para cada serviço, a que horas iniciou o atendimento, em que lugar vai, quantos faltam para o seu, media de tempo para atendimento do utente seguinte e data de término do respetivo atendimento.

- Levantamento de requisitos não funcionais:
  - No caso de ticket para consulta marcada, a aplicação acede à base de dados e (de acordo com o Número do Utente) verifica se tem consulta, apenas gerando um ticket se tiver consulta marcada.
  - No caso de urgência gera, mas quando ultrapassa o limite de consultas de urgência estabelecido, informa que pode não ser possível o atendimento, se ultrapassar o limite final, o *ticket* não é gerado.
  - No caso de atendimento de enfermagem segue as mesmas regras que a urgência.
  - O sistema de tickets reinicia todos os dias às 23:59. A partir das 00:00 já é possível fazer um pedido de ticket para as consultas desse dia.
  - É possível pedir mais do que um ticket, mas apenas um para cada tipo de atendimento.
  - Se um utilizador fizer uso abusivo da aplicação será banido durante um determinado período de tempo.

- Levantamento de requisitos não funcionais:
  - A Base de Dados deve ser desenvolvida em MySQL.
  - A aplicação deve seguir uma arquitetura em 3 camadas (UI, BLL e DAL).
  - Deve usar-se POO.
  - A aplicação deve permitir guardar uma foto por utilizador.



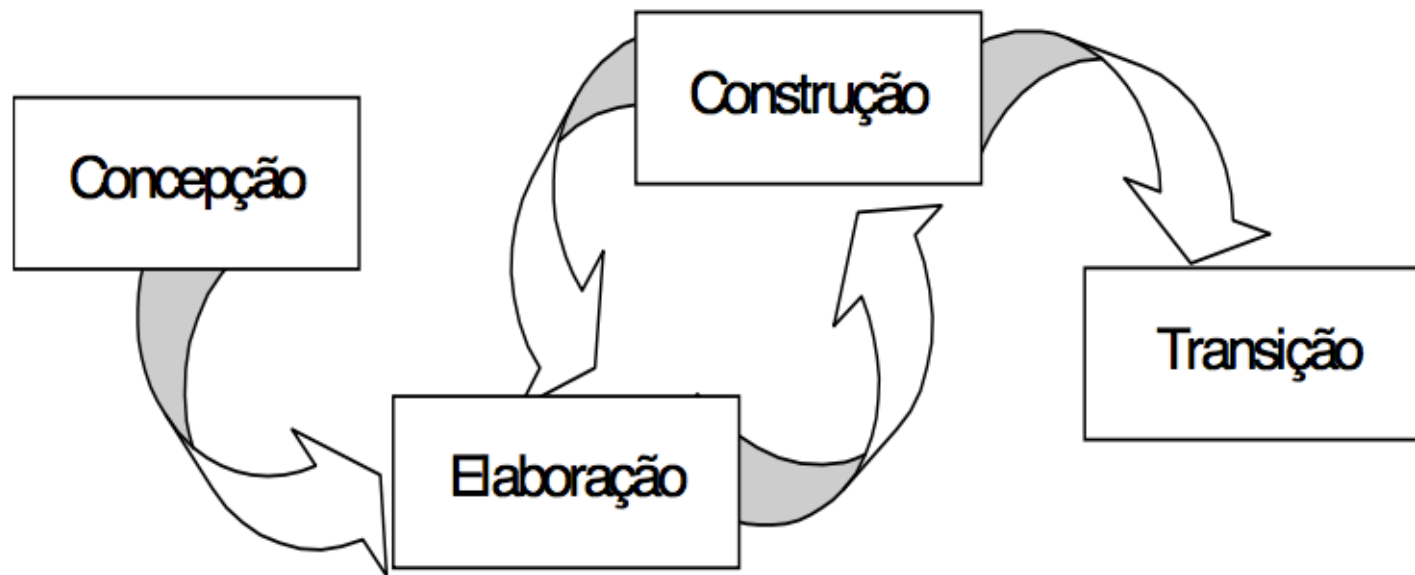
- Defina alguns dos requisitos funcionais e não funcionais de um destes Web Sites:
  - Imovirtual
  - CustoJusto
  - Fnac
  - StandVirtual
  - Facebook

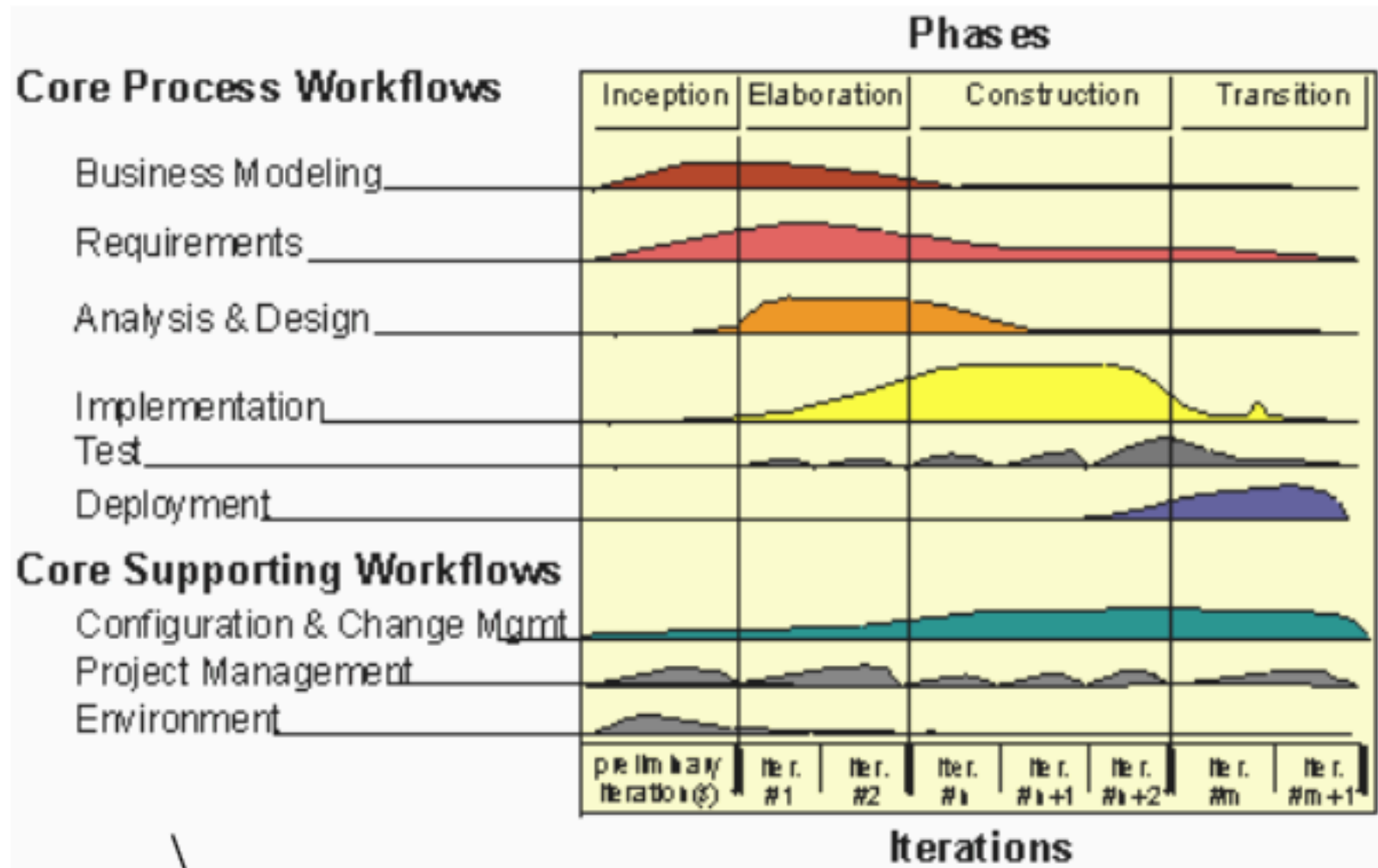


- Engenharia de Software
- Levantamento de Requisitos
- RUP
- UML
- Casos de Uso

- Rational Unified Process
- Ferramenta de desenvolvimento de software iterativo
- Não tenta definir um processo concreto, mas sim uma estrutura de processo adaptável para selecionar os elementos do processo que são apropriados para as necessidades do projecto.

## RUP - Fases





- **Iniciação/ Concepção**

- Delimitação do âmbito do projecto e estudo de viabilidade. Estimativa de custos e cronograma do projecto. Identificação de riscos potenciais.
- Identificação dos principais requisitos (casos de uso críticos) e compreensão do sistema de forma abrangente (alinhamento com estratégia do negócio)

- **Elaboração**

- Incorpora a maior parte da análise de requisitos, a análise de domínio e o projeto.
- Descrição da arquitectura de sw para todo o sistema estável e robusta para todo o sistema, tendo em consideração os seus requisitos.

- **Construção**

- Implementação (programação) e testes

- **Transição**

- Instalação do sistema e sua disponibilização para os utilizadores

- **Fases**

- Cada fase divide-se em iterações a definir em cada projecto concreto

- **Trabalhadores (*workers*)**

- são perfis a que correspondem competências para a realização de actividades

- **Actividades**

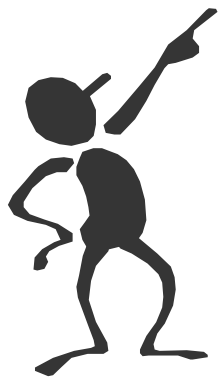
- são tarefas que podem ser entregues a trabalhadores individuais

- **Artefactos**

- são *inputs* e *outputs* de actividades

- ***Workflows***

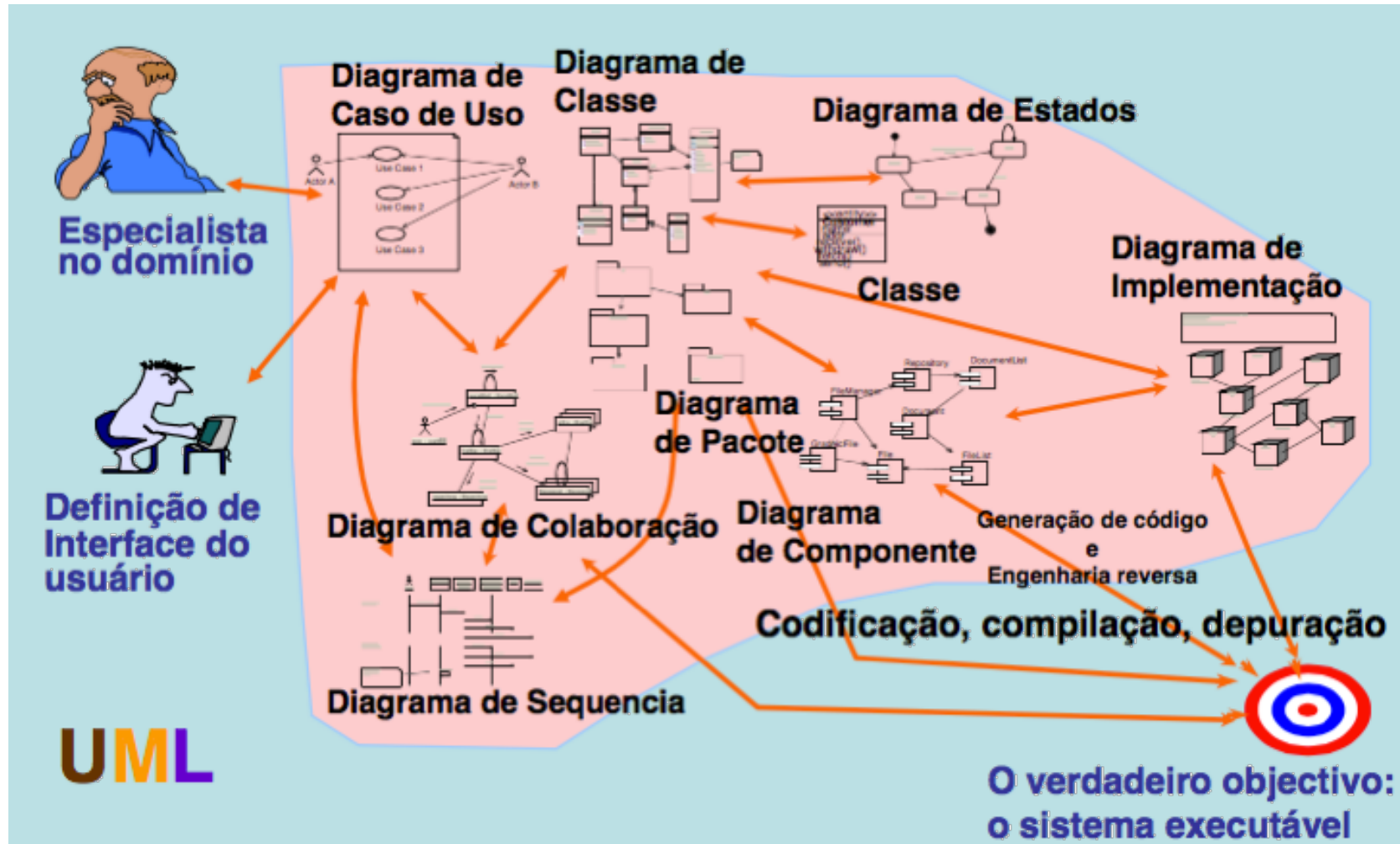
- Agrupam actividades relacionadas



- Engenharia de Software
- Levantamento de Requisitos
- RUP
- UML
- Casos de Uso

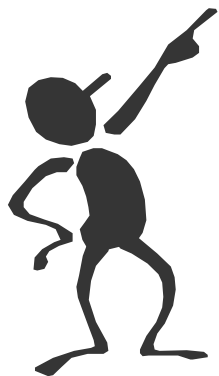


- Unified Modelling Language
- Utiliza notação padrão para especificar, construir, visualizar e documentar sistemas de informação orientados a objectos
- Meio de comunicação entre os diversos elementos envolvidos no processo:
  - Utilizadores
  - Gestores
  - Equipa de desenvolvimento



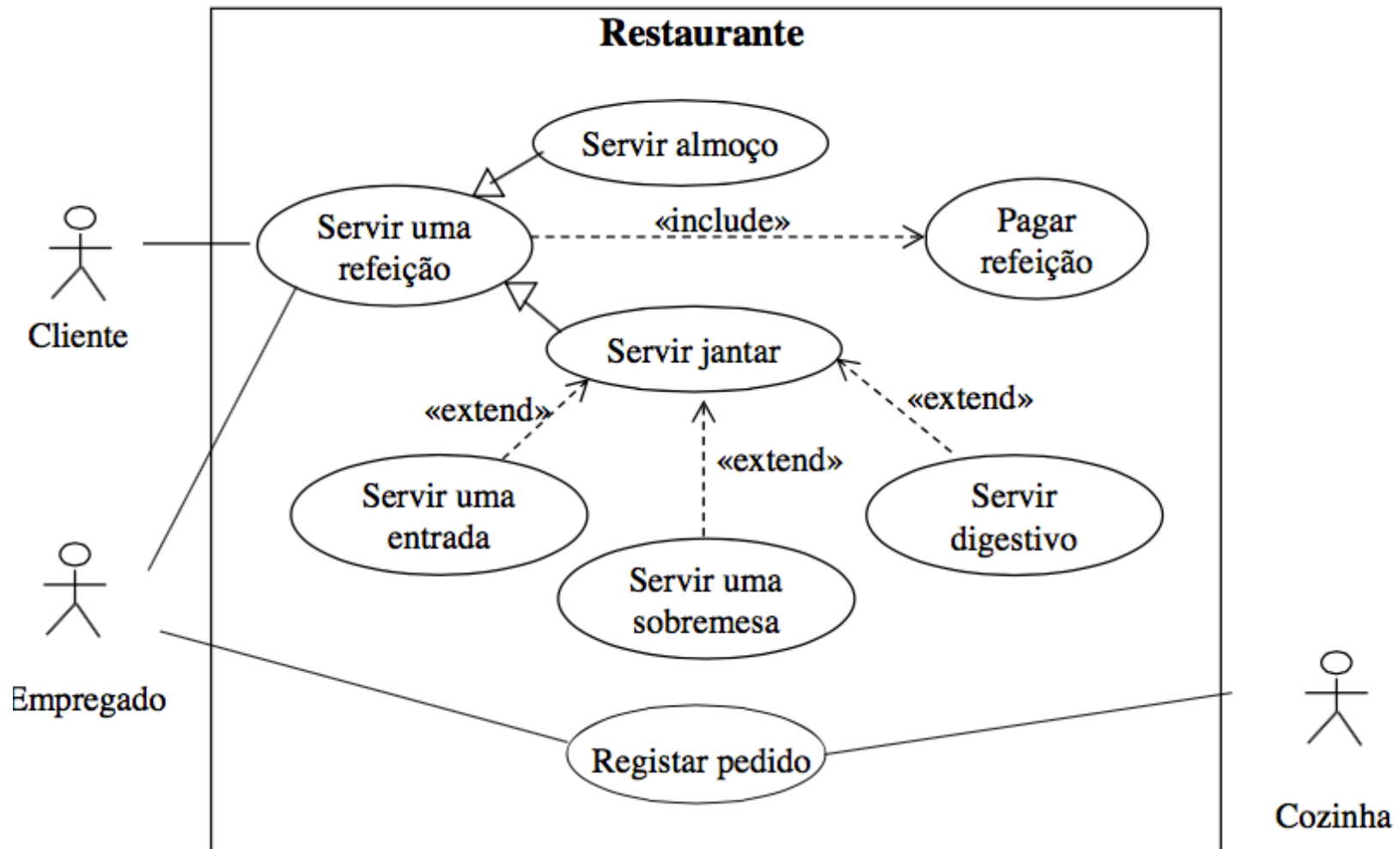
- **Diagramas de *Use Cases***
- **Diagrama de Classes**
- Diagrama de Objectos
- **Diagrama de Sequência**
- Diagrama de Colaboração
- Diagrama de Transição de Estado
- Diagrama de Actividades Diagramas de Arquitectura
  - Diagramas de componentes
  - Diagramas de instalação (deployment)

- Diagrama de *Use Cases*
  - Serve para identificar as fronteiras do sistema e descrever os serviços e funcionalidades que devem ser disponibilizados a cada um dos utilizadores
- Diagrama de Classes
  - Através do qual se descreve a estrutura de informação e relação dos objectos que é utilizada no sistema
- Diagrama de Sequência
  - Mostram como os objectos do sistema interagem para fornecer a funcionalidade de um *use case*. São diagramas de interacção.



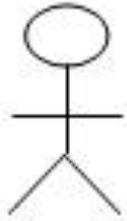
- Engenharia de Software
- Levantamento de Requisitos
- RUP
- UML
- Casos de Uso

- Sequência de acções que um ou mais **actores** realizam num sistema para produzir um resultado observável com valor para os actores.
- É uma funcionalidade do sistema vista pelos utilizadores (sem considerar o comportamento interno do sistema).

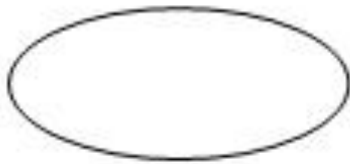


- Permite:
  - Especificar o **contexto** do sistema com quem interage (**actores**) e com que finalidade (**casos de utilização**)
    - Casos de utilização são funcionalidades do sistema vistas pelos utilizadores (**o que** faz e não **como** faz)
  - Modelar os **requisitos funcionais** do sistema
  - Pode referir-se a um sistema de software, um sistema de negócio ou organização, um equipamento, uma classe, etc.
  - É elaborado por analistas e especialistas de domínio nas fases iniciais do desenvolvimento (análise) e refinado nas fases seguintes (projecto) numa perspectiva de desenvolvimento iterativo e incremental.





- Actor



- *Use Case*



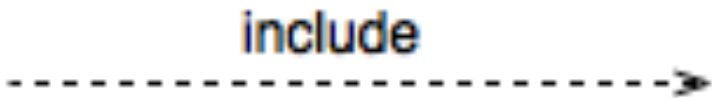
- Sistema



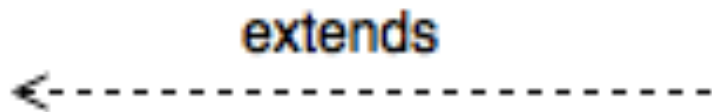
- Associação



- Generalização/Herança



- *Extend*



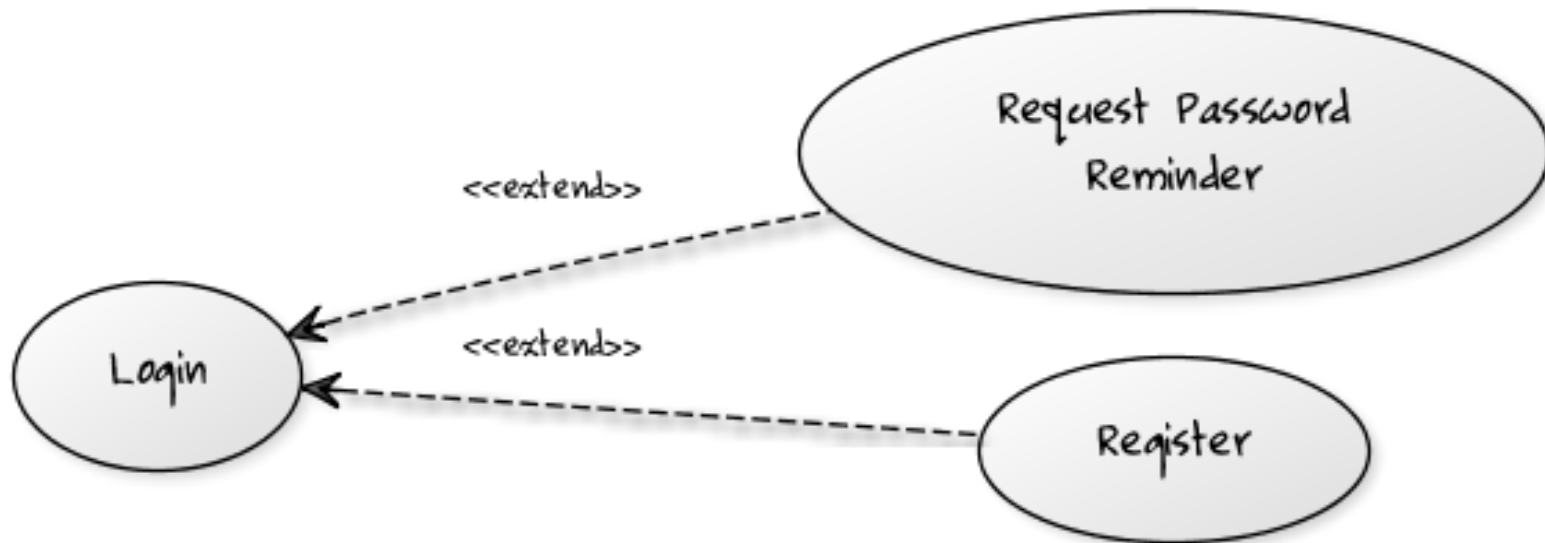
- *Include*

- É um papel que alguma entidade do ambiente envolvente ao sistema desempenha quando interage com o sistema.

- Usufri das mesmas propriedades que uma relação pai/filho, onde o *use case* filho herda o comportamento do pai

- **Inclusão** <<include>> (seta a tracejado)
  - O caso base incorpora o comportamento do outro
  - Corresponde a uma relação de delegação
  - Evita repetir os mesmos fluxos de acção
- **Extensão** <<extend>> (seta a tracejado)
  - O caso destino pode ser estendido com o comportamento de outro(s).
  - Permite representar subfluxos de acções alternativos





- **Primários:** representam os *objectivos* dos actores.
  - Que informações o sistema deve produzir
  - O sistema deve realizar alguma acção que ocorre regularmente no tempo
  - Para cada requisito funcional, existe um (ou mais) caso(s) de uso para atendê-lo
- **Secundários:** aqueles que não trazem benefício directo para os actores, mas que são necessários para que sistema funcione
  - Manutenção de ficheiros.
  - Manutenção de utilizadores.
  - Manutenção de informações provenientes de outros sistemas.



- Parte-se do princípio que todos os *Use Cases* são originados por actores, contudo existem alguns que podem ser despoletados pelo sistema
- Exemplo:
  - Cópia de segurança para uma unidade de backup
  - Enviar promoções mensais a uma lista de clientes

- Também é comum encontrem-se as funcionalidades de um sistema descritas em texto livre para complementar os diagramas gráficos.

- Pré-condição
  - Indica o que deve existir inicialmente para que o cenário descrito seja seguido com sucesso
- Pós-condição
  - É demonstrado o que irá acontecer depois do cenário ser concluído

- Cenário Principal
  - Onde não surgem problemas
- Cenário Secundário
  - Onde são descritos caminhos alternativos, quando se pensa no que poderá correr mal

- Efectuar uma encomenda pela internet:
  - O cliente após ter validado o seu acesso, selecciona a opção Encomendar, sendo mostrado o catálogo dos produtos. Para adicionar um produto tem apenas de introduzir o código do mesmo para que os seus dados sejam visualizados. Ao mesmo tempo é calculado o valor da encomenda.
  - Através da confirmação do cliente, o sistema passa para a função de pagamento. Depois da introdução e confirmação dos dados do cartão de crédito é atribuído um código à encomenda, que posteriormente será entregue na morada indicada.

- Use Case
  - Efectuar uma encomenda pela internet:
- Pré-condição
  - O cliente é um utilizador válido no sistema
- Descrição (Cenário Principal)
  1. O cliente selecciona a opção encomendar
  2. É mostrado o catálogo de produtos
  3. O cliente adiciona produtos à encomenda
  4. O sistema mostra info do produto
  5. Cada vez que é adicionado um produto o sistema calcula o valor da encomenda
  6. O cliente confirma a encomenda
  7. O sistema pede os dados do cartão de crédito
  8. O sistema confirma os dados do pagamento e atribui código à encomenda
- Pós-condição
  - A encomenda será entregue na morada do cliente

- Caminhos Alternativos (Cenário Secundário)

3.1 Se o código que o cliente inseriu não é inválido, o sistema deve avisar com uma mensagem

8.1 Se o cartão for inválido, o sistema informa o cliente e volta para o ponto 7.

- Use Case
  - Efectuar uma encomenda pela internet:
- Pré-condição
  - O cliente é um utilizador válido no sistema
- Descrição (Cenário Principal)
  1. **Include: Controlo de Acesso**
  2. O cliente selecciona a opção encomendar
  3. É mostrado o catálogo de produtos
  4. O cliente adiciona produtos à encomenda
  5. O sistema mostra info do produto
  6. Cada vez que é adicionado um produto, o sistema calcula o valor da encomenda
- **6.1 Extend: Calcular Desconto**
  7. O cliente confirma a encomenda
  8. O sistema pede os dados do cartão de crédito
  9. O sistema confirma os dados do pagamento e atribui código à encomenda
- Pós-condição
  - A encomenda será entregue na morada do cliente



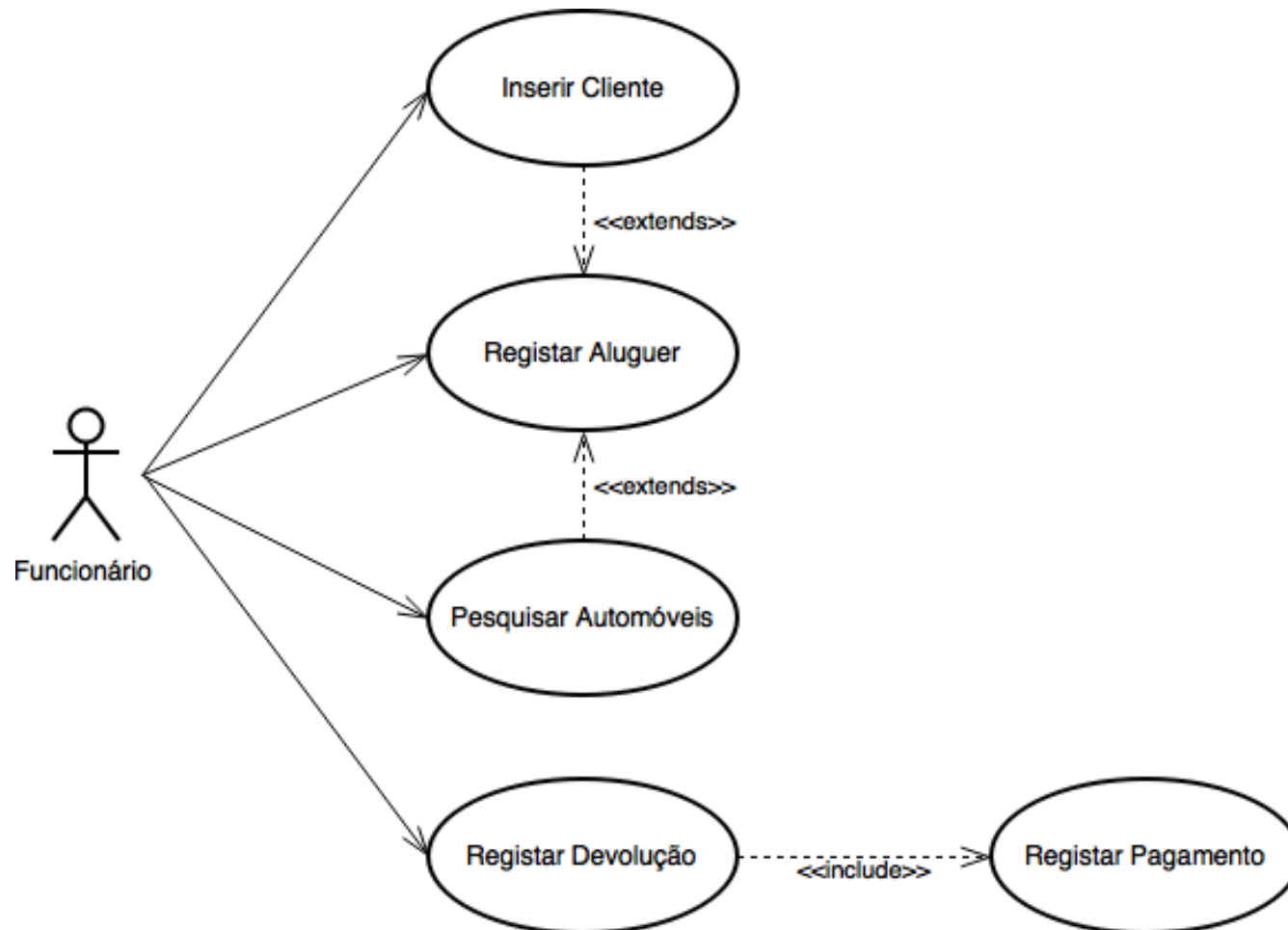
- Use Case
  - Calcular Desconto
- Pré-condição
  - O produto está com desconto internet
- Descrição (Cenário Principal)
  1. O sistema retorna o valor do desconto
  2. Mostra o desconto na encomenda
  3. Calcula o total subtraindo o preço do desconto



- Para que serve um diagrama de Use Case?
- O que é um actor?
- Qual a diferença entre a relação include e extend?
- Qual a notação para generalização?

- Pretende-se desenvolver um software de suporte a diversas actividades de uma empresa de aluguer de automóveis.
- O software é apenas utilizado pelos funcionários em loja.
- Este software deve permitir registar contratos de aluguer, entregas e devoluções de automóveis, com cálculo automático do preço a pagar pelos seus clientes.

- Diagrama de Casos de Uso para o Funcionário



### **Biblioteca**

Da entrevista com o responsável da biblioteca de uma universidade resultou a seguinte descrição para um novo sistema informático:

“A atividade da biblioteca centra-se principalmente no empréstimo de publicações para alunos da universidade. O empréstimo é registado pelos funcionários da biblioteca, que também consultam diariamente os empréstimos cujos prazos foram ultrapassados. Todo este processo é efetuado manualmente, sendo muito ineficiente. Espera-se que o novo sistema resolva esta situação”.

Os alunos necessitam pesquisar os livros existentes na biblioteca. Caso um livro esteja emprestado é mostrada a data esperada de entrega do mesmo”.



## Parque de Estacionamento

Considere os seguintes requisitos de um sistema informático para a gestão de um parque de estacionamento.

- a) O controlo é efetuado com base na matrícula do veículo.
- b) Na entrada do parque existirá um funcionário que introduz as matrículas no sistema, ficando de imediato registado a data e hora de início do estacionamento. O sistema tem que verificar se a matrícula existe.
- c) Se a matrícula não for reconhecida pelo sistema, então o funcionário registará um novo veículo no sistema.
- d) Na saída, um funcionário introduz novamente a matrícula, sendo que o sistema calcula o custo do estacionamento.
- e) O Gestor do Parque precisa de consultar diariamente uma listagem dos estacionamento. Em algumas situações, o gestor poderá desempenhar as funções de atendimento, no entanto, apenas o gestor poderá obter as listagens.



