# DESENVOLVIMENTO DE SISTEMAS EMBEBIDOS

***(MESTRADO EM INFORMÁTICA)***

*- SESSÃO 5: Proposta de metodologia -*

JOÃO MIGUEL FERNANDES
*Email: miguel@di.uminho.pt*
*URL: http://www.di.uminho.pt/~miguel*



## Sumário

1. **Enquadramento**
2. **Metodologia MIDAS**
3. **A Análise na Metodologia MIDAS**





## 1. Enquadramento (1/2)

- **Objectivos deste módulo**
  - **Apresentar uma proposta de metodologia (MIDAS) para desenvolvimento de sistemas embebidos.**
  - **Discutir as características mais importantes (modelo do processo, notação, métodos) dessa metodologia.**
- **Audiência alvo**
  - **licenciados (com ou sem formação na área das TSI) com responsabilidades e experiência comprovada (desejável!) na análise, concepção e implementação de sistemas baseados em software**





## 1. Enquadramento (2/2)

- **Bibliografia recomendada**
  - **Fernandes J.M. (2000). "*MIDAS: Metodologia Orientada ao Objecto para Desenvolvimento de Sistemas Embebidos*". Tese de Doutoramento, DI, UM.**
  - **Jacobson I., Booch G., Rumbaugh J. (1999). "*The Unified Software Development Process*". Object Technology. Addison-Wesley. ISBN 0-201-57169-2.**
  - **Selic B., Gullekson G., Ward P.T. (1994). "*Real-Time Object-Oriented Modeling*". John Wiley & Sons. ISBN 0-471-59917-4.**
  - **Calvez J.P. (1993). "*Embedded Real-Time Systems: A Specification and Design Methodology*". Software Engineering Practice. John Wiley & Sons. ISBN 0-471-93563-8.**





## 2. Metodologia MIDAS (1/16)

- **Caracterização**
  - **Abordagem operacional.**
  - **Especificação unificada, gráfica e multi-vista.**
  - **Modelação orientada ao objecto.**





## 2. Metodologia MIDAS (2/16)

- **Caracterização**
  - **A abordagem operacional usa modelos de especificação que podem ser vistos como programas escritos numa linguagem de alto nível.**
  - **Esses programas são compilados e executados, a fim de se detectarem, durante o desenvolvimento do sistema, falhas e inconsistências.**
  - **A metodologia usa uma abordagem baseada numa representação unificada e neutra.**
  - **A representação do sistema vai sendo progressivamente refinada, até se obter um modelo válido do sistema.**
  - **Nessa altura, podem então escolher-se as tecnologias, nas quais os vários componentes vão ser implementados.**

## 2. Metodologia MIDAS (3/16)

- **Caracterização**
  - **As notações para especificar sistemas reactivos devem ter um forte cariz visual e intuitivo, para facilitar a transição da análise para a implementação e a comunicação entre clientes e projectistas.**
  - **O uso de representações gráficas mostra-se útil, sempre que as especificações não estiverem sobrecarregadas com demasiados pormenores.**
  - **É preferível o uso de modelos multi-vistas, cada um dos quais abordando uma dada perspectiva de modelação do sistema.**
  - **Assim, adoptar-se-á a linguagem UML, que preenche os requisitos atrás indicados (unificada, multi-vista e gráfica)**
  - **Os mecanismos de modelação que UML disponibiliza adaptam-se a um processo incremental e iterativo que se baseie em casos de uso e num modelo de objectos.**





## 2. Metodologia MIDAS (4/16)

- **Caracterização**
  - **Pretende adoptar-se um modelo de processo "sensato", de modo que possam ser cometidos erros em qualquer das fases que o compõem.**
  - **Um modelo que atenda esta consideração pode ser recuado no seu fluxo, permitindo que se revisitem fases anteriores, com o intuito de rectificar o que possa não estar correcto.**
  - **Adicionalmente, um modelo não deve ter um fluxo rígido, pois a flexibilidade na aplicação das diversas etapas e dos diversos passos do processo, deve ser entendida como uma característica positiva.**
  - **Nesta linha de raciocínio, a utilização do modelo em cascata (ou de qualquer uma das suas variantes) será preterida em detrimento de modelos mais iterativos, como o modelo transformacional ou em espiral.**





## 2. Metodologia MIDAS (5/16)

- **Caracterização**
  - **Na abordagem tradicional , a maior fatia do teste está concentrada no fim do projecto.**
  - **Para sistemas embebidos, é crucial validar as restrições que condicionam o seu desenvolvimento o mais cedo possível, visto que os custos de desenvolvimento e teste deste tipo de sistemas são elevados.**
  - **Para construir um sistema que tem de obedecer a requisitos funcionais e não funcionais, devem construir-se vários modelos de especificação e realizar uma implementação.**
  - **Após estarem definidos o conjunto de restrições impostas e o modelo de concepção que suporte a funcionalidade pretendida para o sistema, é provável que não seja possível encontrar uma implementação que satisfaça a totalidade das restrições.**





## 2. Metodologia MIDAS (6/16)

- **Caracterização**
  - **Um modelo de processo iterativo é novamente importante para o desenvolvimento de sistemas embebidos, já que não impõe a necessidade de finalizar a implementação para validar o sistema.**
  - **Além disso, a utilização de protótipos e a validação dos modelos desenvolvidos, em fases precoces do projecto, permite detectar, estudar e corrigir problemas que, noutras situações, só o poderiam ser em fases finais do projecto (ou, pior ainda, durante a utilização do sistema).**
  - **A necessidade de modelos de processo flexíveis é, na prática, sentida, porque, por vezes, não é possível aplicar as diversas fases dum dado processo pela ordem em que ele foi inicialmente idealizado.**





## 2. Metodologia MIDAS (7/16)

- **Caracterização**
  - **É fundamental perceber que a metodologia, per si, não é suficiente para assegurar êxito no desenvolvimento de sistemas.**
  - **Ela deve ser adaptada às necessidades específicas da organização e estendida com outras técnicas úteis.**
  - **A metodologia Midas foi concebida para o desenvolvimento de sistemas embebidos,**
  - **Assim, tentará, tipificar-se as características e as situações em que a metodologia Midas pode ser melhor aproveitada ou nas quais pode ser facilmente enquadrada.**





## 2. Metodologia MIDAS (8/16)

- **Caracterização**
  - **A equipa de projecto será composta por várias pessoas, uma vez que uma pessoa não consegue, sozinha e em tempo útil, desenvolver um sistema complexo.**
  - **Deste facto resulta a necessidade de algum tipo de gestão de projecto (gestão de recursos humanos, gestão financeira, calendarização de tarefas, organização do trabalho em grupos, condução de reuniões, formato da documentação, etc.), que não será tratada aqui**
  - **Também a realização dum estudo de viabilidade, antes de se iniciar a fase de análise, é de extrema importância para as fases seguintes de projecto, mas também não será encarada.**
  - **Apesar de ser possível projectar sistemas para uso próprio, a situação mais comum é o sistema ser desenvolvido para outrém.**

## 2. Metodologia MIDAS (9/16)

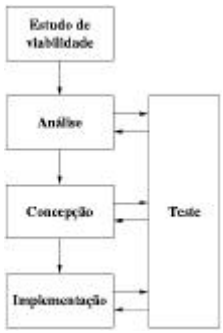
- **Caracterização**
  - **A metodologia parte do pressuposto que a equipa de projecto desenvolve um sistema que lhe foi encomendado por um cliente, entendido como uma entidade externa à equipa de projecto.**
  - **Regra geral, o cliente é especialista no domínio de aplicação, mas não é conhecedor de linguagens de especificação, modelação de sistemas, hardware, software, etc.**
  - **Por outro lado, é lícito supor que o conhecimento dos projectistas na área de aplicação é geralmente diminuto.**
  - **A metodologia pode ser usada para desenvolver de raiz sistemas, ou então em situações de engenharia reversa (reengenharia).**
  - **A primeira situação pode ser considerada um caso especial da segunda, em que se desenvolve o sistema pela primeira vez.**





## 2. Metodologia MIDAS (10/16)

- **Modelo do processo**
  - **Esta visão está em consonância com a divisão do processo, em dois níveis: macro e micro.**
  - **O macro-processo segue o modelo em cascata, tem associada uma escala temporal maior (meses ou anos) e serve como uma referência de controlo para o micro-processo.**
  - **O micro-processo segue o modelo em espiral, tem associada uma escala temporal mais reduzida (semanas, dias ou mesmo horas) e facilita uma abordagem iterativa e incremental no desenvolvimento de sistemas.**








## 2. Metodologia MIDAS (11/16)

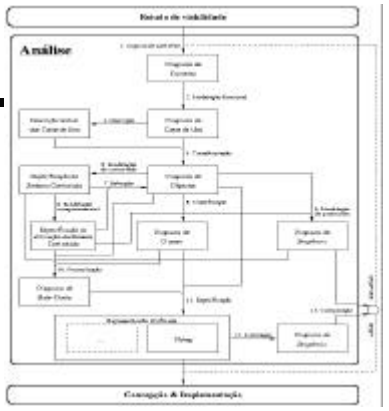
- **Modelo do processo**
  - **Garantir o cumprimento do macro-processo é uma tarefa que incumbe aos gestores do projecto**
  - **O micro-processo é da responsabilidade dos projectistas, a quem uma determinada tarefa de desenvolvimento foi atribuída.**
  - **Esta perspectiva do processo a dois níveis permite reconciliar duas forças antagónicas.**
  - **Por um lado, há um certo conflito entre projectos muito cerimoniosos e projectos sem qualquer cerimónia.**
  - **Os primeiros impedem qualquer assomo de criatividade, pois tudo tem de seguir, cegamente, o estipulado pelo método (exemplo, SSADM), ao passo que os segundos convergem facilmente para situações caóticas e insustentáveis.**





## 2. Metodologia MIDAS (12/16)

- **Fase de análise**








## 2. Metodologia MIDAS (13/16)

- **Modelo do processo**
  - **1. Captura do ambiente: Obtém-se o diagrama de contexto que mostra o sistema, entendido como uma entidade única, e todos os actores, vistos como entidades externas.**
  - **2. Modelação funcional: Com base no diagrama de contexto, constrói-se o diagrama de casos de uso, que é entendido como um refinamento daquele. Para cada actor do sistema, especificado no diagrama de contexto, devem identificar-se os respectivos casos de uso a que ele recorre.**
  - **3. Descrição: Neste passo, escreve-se para cada um dos casos de uso indicados no respectivo diagrama, uma descrição textual que caracteriza os seus aspectos funcionais. Essa descrição deve indicar o tipo de informação utilizada pelo caso de uso.**





## 2. Metodologia MIDAS (14/16)

- **Modelo do processo**
  - **4. Transformação: Os casos de uso são transformados em objectos seguindo regras. Este passo constitui uma das inovações que a metodologia Midas introduziu.**
  - **5. Modelação do controlado: Para sistemas com uma forte componente de controlo, este passo permite determinar as características do sistema controlado.**
  - **6. Modelação comportamental: Tendo em conta o sistema controlado, este passo especifica a forma como o mesmo será usado. A forma de especificar essa utilização não está fixada, dependendo do sistema em causa.**
  - **7. Selecção: Este passo permite escolher os objectos a ter em consideração para desenvolvimento, de acordo com algum critério. Pode restringir-se o sistema àqueles objectos que, por alguma razão, se estimam como importantes.**

## 2. Metodologia MIDAS (15/16)

- **Modelo do processo**
  - **8. Classificação: Para os objectos criados, indica-se a respectiva classe. Através desta classificação, devem identificar-se objectos que pertençam à mesma classe e relacionar as várias classes entre si (por hierarquia).**
  - **9. Modelação de protocolos: Tendo em conta os objectos que constituem o sistema e as ligações entre eles, descrevem-se algumas interacções (mais típicas ou relevantes) entre alguns dos objectos do sistema.**
  - **10. Formalização: Por análise dos diagramas de sequência e do documento que especifica a utilização do sistema controlado, criam-se diagramas de state-charts, para qualquer objecto/classe cujo comportamento dinâmico seja relativamente complexo.**





## 2. Metodologia MIDAS (16/16)

- **Modelo do processo**
  - **11. Especificação: Neste passo, com base nos diagramas de objectos, de classes e de state-charts, procede-se à criação duma especificação, que descreve essas 3 vistas do sistema. Esta transformação utilização a linguagem Oblog como representação unificada de sistemas embebidos.**
  - **12. Simulação: Usando a especificação como entrada, este passo permite simular o sistema, sendo criados diagramas de sequência que descrevem situações de funcionamento do modelo especificado.**
  - **13. Comparação: Com base nos 2 conjuntos de diagramas de sequência obtidos nos passos 10 e 12, faz-se uma comparação dos resultados de ambos os conjuntos, de forma a verificar se o comportamento que se pretende obter corresponde ao comportamento especificado.**





## 3. Análise em MIDAS (1/40)

- **Diagrama de contexto**
  - **Apesar da notação UML não suportar directamente a descrição de diagramas de contexto, é possível adaptar os diagramas de caso de uso e de objectos, para esse fim.**
  - **Assim, usam-se actores para especificar as entidades externas que comunicam com o sistema e um objecto para representar todo o sistema.**
  - **Pode também olhar-se o diagrama de contexto como sendo o diagrama de casos de uso de mais alto nível, que é constituído por um único caso de uso, representativo de toda a funcionalidade disponibilizada pelo sistema.**





## 3. Análise em MIDAS (2/40)

- **Diagrama de contexto**
  - **A estratégia para criar o diagrama de contexto consiste nos 3 passos seguintes:**
    - **Construir uma lista de actores, entradas e saídas.**
    - **Desenhar o diagrama de contexto, escolhendo o tipo adequado para cada uma das ligações entre actores e o sistema.**
    - **Especificar pormenorizadamente as ligações numa tabela.**
  - **Identificar os actores dum sistema pode revelar-se ingrato, pois por vezes é complicado justificar se uma dada funcionalidade está dentro ou fora do sistema.**





## 3. Análise em MIDAS (3/40)

- **Diagrama de contexto**
  - **Normalmente, é sempre necessária alguma iteração na identificação dos actores, até se atingir uma solução estável.**
  - **Algumas recomendações para identificar os actores:**
    - **O enunciado do problema refere-se explicitamente ao objecto, (ex: quando o sistema tem de interactuar com dispositivos).**
    - **O objecto já está disponível no ambiente da aplicação, ou seja, não será desenvolvido como parte do sistema.**
    - **O objecto envia ou recebe informação do sistema.**
    - **O objecto é uma pessoa que usa o sistema.**
  - **Quando nenhuma destas hipóteses se aplica a um dado objecto, então esse objecto deve ser desenvolvido no âmbito do sistema em causa e nele incorporado.**





## 3. Análise em MIDAS (4/40)

- **Diagrama de contexto**
  - **Outra forma alternativa ou complementar de encontrar os actores do sistema consiste em colectar as respostas a uma série de questões:**
    - **Quem usa o sistema?**
    - **Quem instala o sistema?**
    - **Quem arranca o sistema?**
    - **Quem mantém o sistema?**
    - **Quem desliga o sistema?**
    - **Que outros sistemas usam o sistema?**
    - **Quem recebe informação do sistema?**
    - **Quem disponibiliza informação ao sistema?**
    - **Algo sucede automaticamente no momento de arranque?**

## 3. Análise em MIDAS (5/40)

- **Comunicação inter-objectos**
  - **As interligações de comunicação descritas são aplicáveis a todos os diagramas em que existam ligações entre objectos (diagramas de classes, objectos, sequência e colaboração).**
  - **Regra geral, as metodologias para desenvolvimento de software usam apenas, para comunicação, mensagens.**
  - **A mensagem é um mecanismo de comunicação ponto a ponto, pelo que não é possível a difusão de mensagens.**
  - **A representação completa do comportamento dum sistema, à luz dum meta-modelo que inclui apenas o mecanismo de passagem de mensagens, mostra-se uma solução demasiado vinculada à forma como os componentes de software comunicam entre si .**





## 3. Análise em MIDAS (6/40)

- **Comunicação inter-objectos**
  - **Assim, um meta-modelo baseado apenas em mensagens limita significativamente uma característica desejada para os modelos, que consiste na independência destes com a solução final (hardware ou software).**
  - **Alguns exemplos concretos elucidam melhor este assunto.**
  - **Entre os objectos candidatos para implementação em hardware encontram-se aqueles que lidam com sinais de elevada largura de banda (visão e áudio).**
  - **Modelar as ligações a esses objectos através de protocolos usando apenas mensagens, além de bastante inadequado, poderia confundir os projectistas sobre qual a melhor implementação para esses objectos.**





## 3. Análise em MIDAS (7/40)

- **Comunicação inter-objectos**
  - **Também há situações em que a ocorrência de eventos é relevante para vários componentes, pelo que convém, por questões de expansibilidade, que o objecto, responsável por assinalar o evento, não necessite de conhecer a quem esse evento é relevante.**
  - **Tal situação não é facilmente modelada com mensagens, pois, recorrendo unicamente a esse mecanismo, será inevitável serem explicitamente enviadas mensagens a todos os objectos sensíveis a esse evento.**
  - **O objecto responsável pela sinalização do evento terá, portanto, a ingrata tarefa de conhecer a que objectos e quando deve dar conhecimento da ocorrência do evento.**





## 3. Análise em MIDAS (8/40)

- **Comunicação inter-objectos**
  - **Finalmente, certos objectos podem ser responsáveis por disponibilizar continuamente certos valores, ao longo do tempo, sem necessidade desses objectos conhecerem quem usa, e quando, essa informação.**
  - **Como exemplos típicos, considere-se um relógio que disponibiliza continuamente a hora do dia, um termómetro que proporciona a temperatura, ou um sensor que fornece em contínuo o valor do nível de concentração de gás num dado ambiente fechado (quarto).**
  - **Embora seja sempre possível disponibilizar a informação mediante o estabelecimento dum protocolo envolvendo mensagens, tal solução não é a mais natural, podendo até ser entendida como forçada.**





## 3. Análise em MIDAS (9/40)

- **Comunicação inter-objectos**
  - **Em sistemas reactivos, o tratamento dos eventos e da informação depende tipicamente do estado do sistema, que se encontra distribuído pelos objectos que o compõem.**
  - **Este facto requer que cada objecto tenha acesso aos eventos e às informações que lhe são relevantes, de acordo com as necessidades e o estado desse objecto.**
  - **Muitas vezes, o controlo dinâmico e a transformação da informação são os aspectos mais importantes de concepção, sendo irrelevante o controlo sobre o acesso à informação.**
  - **Assim, o apertado controlo sobre o acesso que é imposto pela comunicação por mensagens apresenta-se demasiado restritivo para sistemas embebidos, motivo pelo qual são propostos alguns novos tipos de ligações inter-objectos.**





## 3. Análise em MIDAS (10/40)

- **Comunicação inter-objectos**
  - **Em Midas, estão disponíveis 4 tipos de interligações :**
    - **Interacção,**
    - **Evento,**
    - **Fluxo discreto de informação,**
    - **Fluxo contínuo de informação.**

| Tipo de interligação | Símbolo gráfico | Estereótipo |
|---|---|---|
| Interacção |  | «Interaction» |
| Evento |  | «Event» |
| Fluxo discreto de informação |  | «discrete info» |
| Fluxo contínuo de informação |  | «continuous info» |

## 3. Análise em MIDAS (11/40)

- **Comunicação inter-objectos**
  - **Interacção: Retrata o habitual mecanismo de passagem de mensagens em software, sendo a sua semântica similar à chamada duma função.**
  - **Trata-se dum mecanismo síncrono de comunicação entre 2 objectos, em que o emissor requisita uma dada operação ao receptor, ficando em suspenso até que este responda, através do envio dum resultado.**
  - **Evento: É usado, por um objecto, para indicar as mudanças do seu próprio estado que sejam relevantes a outros objectos.**
  - **Não há a noção de sincronismo, dado que a natureza e o tempo de resposta não são controlados pelo objecto emissor.**
  - **Trata-se dum mecanismo de comunicação fundamental para sistemas reactivos, usado com frequência na especificação dos diagramas de estados dos objectos.**





## 3. Análise em MIDAS (12/40)

- **Comunicação inter-objectos**
  - **Fluxo discreto de informação: Exprime a transferência de dados entre objectos, duma forma discreta.**
  - **Neste mecanismo não existe a noção de sincronismo, dado que o receptor da informação pode ler qualquer valor anteriormente gerado sempre que dele necessitar.**
  - **Não há qualquer tipo de controlo por parte do emissor sobre quando a leitura é feita, pois aquele apenas controla o valor da informação, pois é o responsável pela sua actualização.**
  - **Fluxo contínuo de informação: Este mecanismo de ligação é muito idêntico ao anterior, mas pressupõe a transferência de dados entre objectos, dum modo contínuo. Directamente, o receptor da informação tem apenas disponível para uso o último valor de informação (i.e. o valor actual).**





## 3. Análise em MIDAS (13/40)

- **Comunicação inter-objectos**
  - **Os modos assíncronos de comunicação implicam, na prática, que o objecto responsável pela informação, após a disponibilizar, continua a sua execução, independentemente da forma como essa informação é tratada pelo receptor.**
  - **Em comunicações síncronas, sucede o contrário, ou seja, o emissor da informação fica à espera do tratamento dado pelo receptor para poder prosseguir a sua execução.**
  - **O modo assíncrono de comunicação permite ao emissor continuar a executar logo após a mensagem ter sido despachada.**





## 3. Análise em MIDAS (14/40)

- **Comunicação inter-objectos**
  - **A principal vantagem associada ao modo assíncrono é o período de tempo, durante o qual o objecto emissor está a tratar da comunicação, ser curto.**
  - **No modo síncrono de comunicação, o emissor fica bloqueado até receber uma resposta à mensagem inicial. Durante esse período, não pode responder a outros estímulos que, eventualmente, lhe sejam enviados.**
  - **A vantagem desta forma de comunicação consiste no nível de controlo relativamente à ordem das actividades que é implicitamente disponibilizado.**





## 3. Análise em MIDAS (15/40)

- **Diagramas de casos de uso**
  - **Para identificar os casos de uso dum sistema, a "melhor" estratégia consiste em tentar encontrar, primeiramente, os actores desse sistema e, depois, para cada um daqueles, colectar uma lista de possíveis casos de uso.**
  - **Cada fluxo completo de eventos, iniciado por um actor, representa um caso de uso do sistema em análise.**
  - **Os actores do sistema já foram obrigatoriamente identificados no diagrama de contexto, pelo que resta a tarefa de colectar uma lista de casos de uso.**
  - **É preferível ver a construção dos diagramas de contexto e de casos de uso como um processo iterativo.**





## 3. Análise em MIDAS (16/40)

- **Diagramas de casos de uso**
  - **Os casos de uso podem ser decompostos em outros casos de uso, repetindo-se o processo até se chegar aos cenários que mostram sequências de interacção entre objectos.**
  - **Assim, outra possibilidade para construção dos diagramas de casos de uso consiste na identificação dos cenários do sistema, a partir dos quais é possível, por agregação, inferir os casos de uso.**
  - **A semântica associada aos casos de uso levanta alguns problemas,**
  - **Um desses problemas refere-se à possibilidade de haver casos de uso sem qualquer ligação, implícita ou explícita, a actores.**

## 3. Análise em MIDAS (17/40)

- **Diagramas de casos de uso**
  - **Tal não é permitido em UML e esse caso de uso contraria o propósito dos casos de uso, como mecanismo para captar as funcionalidades que o sistema disponibiliza aos seus utentes.**
  - **No entanto, essa possibilidade é útil para modelar tarefas que têm de ser executadas periódica e automaticamente e sem qualquer intervenção externa.**
  - **Um sistema que tenha que fazer alguma operação automática (estatísticas, log ou reciclagem de memória) deve ele próprio ser o responsável por iniciar internamente essa operação.**
  - **Sugere-se a utilização dum estereótipo <<internal>> associado aos casos de uso que representam essas tarefas iniciadas internamente.**





## 3. Análise em MIDAS (18/40)

- **Diagramas de casos de uso**
  - **Assim que o diagrama de casos de uso estiver desenhado, deve produzir-se uma descrição textual (ou gráfica) sobre cada um dos casos de uso, para servir como referência às fases seguintes do projecto.**
  - **Há vários formatos para descrever os casos de uso: texto informal, texto estruturado em passos numerados (com pré e pós-condições), pseudo-código ou diagramas de actividade.**
  - **Com base nos casos de uso do sistema, é possível seguir, no projecto, uma abordagem guiada ao risco.**
  - **Para tal, deve construir-se uma lista com todos os casos de uso e atribuir-se a cada um deles um nível de importância, usando uma escala apropriada.**





## 3. Análise em MIDAS (19/40)

- **Diagramas de casos de uso**
  - **Pode, assim, planear-se a implementação do sistema com base no nível de importância atribuído a cada caso de uso.**
  - **Os casos de uso indispensáveis são implementados em primeiro lugar, os importantes a seguir, os úteis depois e, se houver tempo, implementam-se os casos de uso supérfluos.**
  - **Assim, quando se ultrapassa o prazo estabelecido inicialmente para um projecto, é mesmo assim possível apresentar um sistema com as funcionalidades mais importantes a funcionar (e supostamente testadas).**
  - **As funcionalidades não incluídas podem ser introduzidas na versão seguinte do sistema, sem que daí resultem grandes inconvenientes para os utilizadores.**





## 3. Análise em MIDAS (20/40)

- **Diagramas de objectos**
  - **Assim que o comportamento exteriormente visível do sistema estiver definido, devem identificar-se os objectos e as classes que permitem descrever o sistema em desenvolvimento.**
  - **Ainda há quem não distinga claramente entre os objectos e as classes.**
  - **A confusão entre objectos e classes está profundamente relacionada com a natureza abstracta do software.**
  - **Por exemplo, tanto a planta duma casa como a própria casa podem ser modeladas por objectos, mas também se pode entender a planta como a "classe" da casa.**
  - **Não há mal algum com esta visão enquanto os dois tipos de objectos estiverem separados.**





## 3. Análise em MIDAS (21/40)

- **Diagramas de objectos**
  - **Os problemas surgem quando se misturam no mesmo modelo os objectos e as classes do sistema a construir.**
  - **Este é ainda um dos problemas mais comuns que se verificam, quando se usam conceitos orientados ao objecto.**
  - **Uma classe pode ser vista como um padrão que permite criar objectos para a aplicação em causa e, provavelmente, para outras aplicações futuras, enquanto que os objectos representam os elementos que constituem realmente a aplicação.**
  - **Esta perspectiva permite concluir que é preferível não incorporar no mesmo modelo as classes e os objectos.**





## 3. Análise em MIDAS (22/40)

- **Diagramas de objectos**
  - **O espaço de análise pode dividir-se em três dimensões ortogonais: informação, comportamento e apresentação.**
  - **A dimensão informação descreve os dados do sistema, o que permite especificar o seu estado interno.**
  - **A dimensão comportamento indica quando e como o estado do sistema é alterado.**
  - **A dimensão apresentação providencia os pormenores para apresentar o sistema ao exterior.**

## 3. Análise em MIDAS (23/40)

- **Diagramas de objectos**
  - **O modelo de análise constrói-se através da especificação de objectos neste espaço tri-dimensional.**
  - **Uma hipótese, que não se recomenda, é usar objectos que expressam apenas uma única dimensão.**
  - **É isso que acontece nos métodos estruturados, em que as funções são colocadas no eixo comportamento e os dados no eixo informação.**
  - **Outra hipótese, que também não se recomenda, consiste na utilização de objectos, que podem ser colocados em qualquer posição do espaço.**
  - **Esta perspectiva vê todos os objectos como iguais, mas não permite classificar um objecto segundo o papel que desempenha no sistema.**





## 3. Análise em MIDAS (24/40)

- **Diagramas de objectos**
  - **Em Midas, adoptou-se a perspectiva que associa, a cada objecto uma dada categoria, permitindo obter uma estrutura que pode, mais facilmente, adaptar-se às mudanças.**
  - **Cada uma das categorias está fortemente relacionada com uma das dimensões, sem contudo implicar a inexistência de componentes nas outras 2 dimensões.**
  - **Os objectos podem classificar-se em três categorias:**
    - **Objectos-interface.**
    - **Objectos-entidade.**
    - **Objectos-função.**





## 3. Análise em MIDAS (25/40)

- **Diagramas de objectos**
  - **Um objecto-interface modela comportamento e informação que dependem do interface do sistema, i.e. do diálogo (da comunicação) do sistema com os actores que com ele interagem.**
  - **Tudo o que respeita a interface do sistema deve ser colocado em objectos-interface.**
  - **O objecto-interface deve estar de tal forma encapsulado que, se houver alguma mudança no modo de comunicação, só o objecto-interface é modificado, ficando inalterados todos os restantes objectos.**
  - **Os objectos-interface, por permitirem isolar as partes de interface das partes funcionais, tornam mais directa a reutilização destas (sob a forma de objectos-função).**





## 3. Análise em MIDAS (26/40)

- **Diagramas de objectos**
  - **Um objecto-entidade modela principalmente informação, cuja existência deve ser prolongada (não se incluem, portanto, dados com um carácter temporário).**
  - **Para além dos atributos que caracterizam o objecto-entidade, todo o comportamento associado à manipulação dessa informação deve ser incluído nesse objecto-entidade.**
  - **Como exemplo dum objecto-entidade considere-se uma conta bancária com os respectivos atributos e operações.**





## 3. Análise em MIDAS (27/40)

- **Diagramas de objectos**
  - **Um objecto-função modela comportamento que não pode ser associado, duma forma natural, a nenhum outro objecto.**
  - **Exemplo: a funcionalidade que opera sobre vários objectos e que devolve o resultado a um objecto-interface.**
  - **Considere-se um objecto que calcula o total dos saldos dum conjunto de contas bancárias.**
  - **Esta funcionalidade poderia ser atribuída a um dos objectos-entidade (conta bancária) ou ao objecto-interface encarregado de apresentar o resultado final, mas não é responsabilidade de qualquer um desses objectos.**
  - **Daí que a solução mais apropriada, em exemplos similares, seja a introdução dum objecto-função responsável por disponibilizar essa funcionalidade.**





## 3. Análise em MIDAS (28/40)

- **Diagramas de objectos**
  - **As categorias de objectos adoptadas tornam o modelo mais estável, pois as mudanças a realizar estão mais localizadas.**
  - **As alterações mais frequentes a que um sistema está sujeito são a sua apresentação e o seu comportamento, sendo a informação a componente mais estável dos sistemas.**
  - **Mudanças à apresentação afectam apenas objectos-interface, enquanto que modificações ao comportamento podem perturbar qualquer tipo de objecto:**
    - **Se a funcionalidade está associada a uma informação do sistema, então o objecto-entidade que a representa é afectado.**
    - **Se a funcionalidade a modificar está relacionada com a forma de apresentação, então o respectivo objecto-interface é alterado.**
    - **Alterações a funcionalidades que envolvem vários objectos são, em princípio, locais a um objecto-função.**

## 3. Análise em MIDAS (29/40)

- **Diagramas de objectos**
  - **Os objectos são obtidos a partir dos casos de uso, em que cada um destes é dividido em objectos das 3 categorias.**
  - **O diagrama de objectos representa uma arquitectura ideal do sistema, já que não são considerados quaisquer factores relacionados com a plataforma de implementação.**
  - **A transformação do diagrama de casos de uso para o diagrama de objectos é um dos passos mais importantes e críticos da metodologia Midas.**
  - **Esta transição consiste em distribuir o comportamento especificado pelos casos de uso, por objectos que serão os constituintes do diagrama de objectos.**
  - **Serão dadas algumas recomendações para o modo de proceder nesta tarefa.**





## 3. Análise em MIDAS (30/40)

- **Diagramas de objectos**
  - **A estratégia passa pela distribuição, a vários objectos, do comportamento especificado pelos casos de uso, indicando que parte do comportamento dum caso de uso corresponde a cada objecto introduzido no diagrama dos objectos.**
  - **Um mesmo objecto pode ser comum a diversos casos de uso.**
  - **Uma vez que os objectos são obtidos a partir casos de uso, as referências numéricas dos casos de uso devem ser transpostas para os objectos.**
  - **A referência dum objecto constrói-se com a referência do caso de uso, acrescentando um sufixo 'i', 'e' ou 'f', conforme se trate dum objecto-interface, entidade ou função.**
  - **Facilita-se assim a observação das relações entre os casos de uso e os respectivos objectos (continuidade dos modelos).**





## 3. Análise em MIDAS (31/40)

- **Diagramas de objectos**
  - **A estratégia (4-step rule set) é então a seguinte:**
    - **Passo 1: Transformar cada caso de uso em 3 objectos (1 função, 1 entidade e 1 interface). Cada objecto recebe a mesma referência que o caso de uso que lhe dá origem, acrescentando um sufixo que distingue a categoria a que pertence.**
    - **Passo 2: Com base nas descrições textuais, para cada caso de uso, decide-se quais as categorias que devem ser mantidas para representar o caso de uso. A decisão deve ser feita considerando todo o sistema (abordagem holística e não cada caso de uso per si, como sucede em algumas abordagens mais reducionistas.**
    - **Passo 3: Os objectos "sobreviventes"' devem ser agregados, sempre que houver sobreposição, numa representação unificada desses objectos.**
    - **Passo 4: Os objectos e os agregados obtidos devem ser ligados para indicar as associações entre esses objectos.**





## 3. Análise em MIDAS (32/40)

- **Diagramas de objectos**
  - **Quando o número de objectos a colocar num diagrama é maior que um dado valor, a respectiva legibilidade degrada-se.**
  - **Recorde-se a famosa regra de Miller "7±2" que indica que a mente humana não consegue, em simultâneo, captar adequadamente mais do que 7 conceitos relacionados (com uma margem, para cima ou para baixo, de 2).**
  - **Esta inaptidão, para manipular várias coisas simultaneamente, consiste numa limitação intrínseca à maioria dos seres humanos que não pode eliminar-se através de aprendizagem.**
  - **Para colmatar esta falha, pode e deve procurar-se algum mecanismo de modelação que permita formar novos objectos, a partir dum conjunto de objectos de mais baixo nível.**





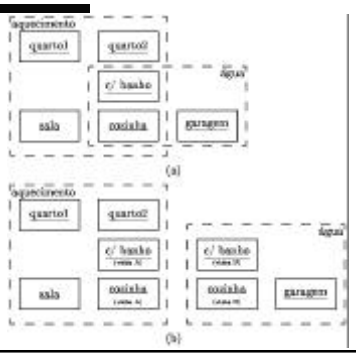
## 3. Análise em MIDAS (33/40)

- **Diagramas de objectos**
  - **Assim, são necessários mecanismos que facilitem a criação de objectos compostos (criando relações entre objecto/sub-objectos, agregado/partes ou sistema/subsistemas), para permitir especificações do mais abstracto para o mais concreto.**
  - **Em sistemas orientados ao objecto, a solução passa pela introdução dum objecto agregado ou composto que constitui uma abstracção, que esconde, a um dado nível, os seus sub-objectos e as respectivas relações.**
  - **O uso de agregação em que há partilha de partes por vários agregados deve evitar-se porque não facilita o relacionamento entre a realidade e o modelo construído.**
  - **Para colmatar este problema, devem dividir-se os objectos que pertencem a vários compostos segundo vistas ortogonais.**





## 3. Análise em MIDAS (34/40)



## 3. Análise em MIDAS (35/40)

- **Diagramas de objectos**
  - **Por vezes, é necessário especificar múltiplos objectos da mesma classe, como por exemplo, quando se tem de construir uma estrutura repetitiva (rede neural artificial ou array de processadores).**
  - **Por este motivo, a notação UML foi estendida para suportar este mecanismo de especificação.**


  - **Assim, com um único símbolo, designado por objecto repetido, representam-se vários objectos organizados de forma regular.**
  - **Trata-se dum mecanismo de abstracção, em que o objecto é replicado várias vezes.**





## 3. Análise em MIDAS (36/40)

- **Diagramas de classes**
  - **O diagrama de classes é um referencial para uma categoria de aplicações que a partir dele se podem construir.**
  - **O modelo de classes é uma generalização de alto nível dos sistemas.**
  - **Dito por outras palavras, quando se define a forma como as classes se relacionam entre si, está a indicar-se quais os possíveis sistemas que podem ser construídos a partir dessas classes.**
  - **Neste sentido, o diagrama de classes funciona como uma "camisa de forças" para o diagrama de objectos.**





## 3. Análise em MIDAS (37/40)

- **Diagramas de classes**
  - **Midas vê o diagrama de classes como um repositório de especificações pré-definidas de objectos.**
  - **A estrutura de classes pode ser composta por várias árvores não ligadas entre si. As subclasses estão semanticamente relacionadas com as suas superclasses.**
  - **Esta perspectiva não inviabiliza a reutilização das implementações, que ocorre como uma consequência natural das relações semânticas entre classes.**
  - **Apesar do mecanismo de herança não impor nenhuma perspectiva, adoptou-se a prática de ver as superclasses como mais abstractas que as subclasses.**
  - **Isto permite que a hierarquia de classes seja uma hierarquia de abstracções e que o mecanismo de herança seja um esquema de classificação.**





## 3. Análise em MIDAS (38/40)

- **Diagramas de state-charts**
  - **Até ao momento, definiu-se a forma como os sistemas devem ser decompostos nos seus objectos constituintes e o modo como estes se inter-relacionam.**
  - **a análise orientada ao objecto inclui também a especificação do comportamento dinâmico dos sistemas, através da utilização de diagramas de state-charts.**
  - **O comportamento dum sistema refere-se à operação interna deste ao longo do tempo.**
  - **De acordo com o tipo de comportamento que os objectos exibem, estes podem ser classificados segundo as duas seguintes categorias, relevantes para sistemas embebidos):**
    - **simples/reactivo.**
    - **passivo/activo.**





## 3. Análise em MIDAS (39/40)

- **Diagramas de state-charts**
  - **Qualquer objecto cujo comportamento não dependa da história associada, diz-se que tem um comportamento simples e não tem, portanto, associada qualquer noção de estado.**
  - **Alguns objectos-informação tem o seu comportamento enquadrado neste tipo.**
  - **Por exemplo, um objecto que disponibiliza aos seus clientes o cálculo da raiz quadrada dum número, devolve sempre o mesmo resultado, independentemente da história.**
  - **Se um objecto tem um comportamento reactivo, então este pode ser descrito por um modelo à luz dum meta-modelo baseado em estados, nomeadamente por um state-chart.**
  - **Desta forma, em qualquer instante, o objecto está num dado estado que determina a forma como aquele reage às mensagens a que pode estar sujeito.**





## 3. Análise em MIDAS (40/40)

- **Diagramas de state-charts**
  - **Um objecto activo está continuamente a executar e é geralmente autónomo (i.e. exibe algum comportamento, sem a necessidade do comando dum outro objecto).**
  - **Os objectos passivos não tomam qualquer iniciativa de iniciar comunicação e só actuam a pedido de outros objectos.**
  - **Os objectos activos servem de base para o controlo de todo o sistema, enquanto que os objectos passivos, sempre que solicitados, fornecem serviços àqueles.**
  - **Tipicamente os objectos-função são activos e os objectos-informação são passivos.**
  - **Em UML, um objecto activo ou uma classe activa são representados pelos seus símbolos habituais (um rectângulo), mas com o bordo mais carregado.**