

# Interface Pessoa-Máquina

Licenciatura em Engenharia Informática

---

## Ficha Prática #01

---

José Creissac Campos  
jose.campos@di.uminho.pt

(v. 1.3)

### Conteúdo

<b>1</b>	<b>Objectivos</b>	<b>1</b>
<b>2</b>	<b>KLM – Keystroke-level Model</b>	<b>1</b>
2.1	Aplicação de KLM . . . . .	1
2.2	Regras para colocação do operador M . . . . .	2
2.3	Ferramentas . . . . .	4
<b>3</b>	<b>Exercícios</b>	<b>5</b>
3.1	KLM . . . . .	5
3.2	Erros de utilização . . . . .	7
3.3	Perfis do Utilizador . . . . .	8

## 1 Objectivos

1. Aprender e praticar KLM.
2. Treinar a definição de perfis de utilizador.

## 2 KLM – Keystroke-level Model

O KLM é um método baseado no conhecimento empírico do sistema psicomotor humano que permite analisar desempenho na realização de tarefas previamente conhecidas. O método é útil para comparar o desempenho previsto de métodos de operação alternativos.

### 2.1 Aplicação de KLM

Cada método de operação (ou *tarefa*) é dividida em operações físicas ao nível do dispositivo (premir uma tecla, mover o rato, premir o botão do rato, etc.). A cada operação corresponde um operador e a cada operador está associado um tempo de execução (ver Tabela 1). Utilizam-se heurísticas para introduzir operações de “preparação mental” (pausas). Somam-se os tempos dos operadores para obter uma previsão do tempo de execução da tarefa.

Para auxiliar a resolução da ficha, é apresentado um conjunto de operadores típicos na Tabela 1 e um conjunto de regras para colocação do operador de preparação mental (**M**), na Secção 2.2.

### Exemplo

Considere que se pretende analisar o modo de apagar um ficheiro num gestor de janelas. Considere ainda as seguintes condições:

- o objectivo é apagar um ficheiro,
- o ficheiro e caixote do lixo estão visíveis,
- a mão do utilizador está colocada no teclado e deve continuar lá no final da operação,
- o utilizador é *average non-secretary typist*.

A primeira alternativa considerar é a de arrastar o ficheiro para o caixote do lixo. Esta alternativa pode ser decomposta na seguinte sequência de acções:

Tabela 1: Operadores tradicionais KLM

Code	Operation	Time
<b>K</b>	Key press and release (keyboard)	Best Typist (135 wpm)
		Good Typist (90 wpm)
		Poor Typist (40 wpm)
		Average Skilled Typist (55 wpm)
		Average Non-secretary Typist (40 wpm)
		Typing Random Letters
		Typing Complex Codes
		Worst Typist (unfamiliar with keyboard)
<b>P</b>	Point the mouse to an object on screen	1.10 seconds
<b>B</b>	Button press or release (mouse)	0.10 seconds
<b>H</b>	Hand from keyboard to mouse or vice versa	0.40 seconds
<b>M</b>	Mental preparation	1.20 seconds
<b>T(n)</b>	Type string of characters	$n \times K$ seconds
<b>W(t)</b>	User waiting for the system to respond	

1. Mover a mão para o rato — **H**
2. Apontar com o rato para o ficheiro — **P**
3. Premir o botão do rato — **B**
4. Arrastar para o caixote do lixo — **P**
5. Largar o botão do rato — **B**
6. Mover a mão para o teclado — **H**

Esta sequência dá origem à string **HPBPBH**. É agora necessário acrescentar os operadores de preparação mental (**M**).

## 2.2 Regras para colocação do operador M

O passo mais difícil de aplicação desta técnica é a colocação dos operadores de preparação mental. Nesta secção apresentamos uma estratégia para o fazer. A partir da codificação com os operadores físicos/respostas do sistema, deve-se aplicar a regra 0 abaixo e depois iterar as regras 1 a 4 até a codificação convergir para uma solução.

**Regra 0** Inserir um **M** à frente de todos os **Ks** que não sejam parte de argumentos<sup>1</sup> e de todos os **Ps** que seleccionam comandos (por oposição a argumentos).

Depois, avaliar a colocação de cada **M**:

**Regra 1** Se o operador que se segue a um **M** for considerado totalmente previsível, remover o **M** (por exemplo, mover o rato e clicar — **MPMK** → **MPK**).

**Regra 2** Se uma sequência de **MKs** constitui uma unidade cognitiva (por exemplo, o nome de um comando), remover todos os **Ms** excepto o primeiro (por exemplo, o comando /giphy no Discord — **MKMKMKMKMKMK** → **MKMKMKMK**).

**Regra 3** Se um **K** é um terminador redundante (por exemplo, o terminador de um comando imediatamente após o terminador de um operando), remover o **M** à frente do **K** (no exemplo anterior, se o operando é preencher os quatro dígitos de um ano e executar o comando é premir `enter` — **MKMKMKMK**<sup>2</sup> → **MKMKMK**).

**Regra 4** Se um **K** termina uma string constante (por exemplo, um `tab` após o nome de um comando), apagar o **M** à frente do **K**; mas se o **K** termina uma string variável (por exemplo, um `tab` após um argumento) então manter o **M**.

### Exemplo

Partindo da string **HPBPBH** anteriormente definida, podemos aplicar as regras acima do seguinte modo:

1. **HMPBMPBH** – por aplicação da **Regra 0**.
2. **HMPBPBH** – por aplicação da regra **Regra 1** (considera-se que mover para o caixote do lixo é uma acção completamente previsível no contexto de querer apagar o ficheiro, pelo que não necessita de preparação mental).

Uma vez que nesta string não existem operadores **K**, as regras 2 a 4 não se aplicam. Como aplicar novamente a **Regra 2** não elimina qualquer **M**, a operação de apagar um ficheiro arrastando-o para o caixote do lixo pode então ser caracterizada pela string **HMPBPBH**. Realizando os cálculos com os valores da Tabela 1, concluímos que a esta operação corresponde um tempo total de 4,4s.

<sup>1</sup>A definição original das regras tinha em consideração uma linguagem de comandos, mas pode ser generalizada para interfaces gráficas. Nesse contexto, o preenchimento de campos numéricos ou de texto de um formulário são considerados *argumentos*.

<sup>2</sup>Uma vez que o ano é considerado um argumento, não houve lugar à colocação de **Ms** nos **K** que o representam, durante a aplicação da regra 0.

## 2.3 Ferramentas

Por forma a facilitar a aplicação deste tipo de técnicas, foram desenvolvidas diversas ferramentas. Desde simples calculadoras, até ferramentas capazes de determinar a sequência de operadores correspondente a uma dada operação.

Para resolver os exercícios de KLM pode recorrer a uma das seguintes alternativas:

- Calculadora com funcionalidades essenciais, disponível em: <http://www.di.uminho.pt/~jfc/KLMcalc/>
- A ferramenta Cogulator, disponível em: <http://cogulator.io/>, tem funcionalidades mais avançadas, como a geração automática das string de operadores e análise da qualidade da descrição da operação. Suporta um conjunto mais alargado de operadores (ver exemplo abaixo) e várias variações do método base.

### Exemplo

Em Cogulator, a sequência de acções definida no exemplo que tem vindo a ser usado poderia ser representada como:

1. **Hands** to mouse
2. **Mental** preparation
3. **Point** to file icon
4. **Click** file icon (100 ms)<sup>3</sup>
5. **Point** to trash
6. **Click** trash (100 ms)
7. **Hands** to keyboard

Como se pode ver na Figura 1, a este método de apagar um ficheiro corresponde o tempo de 4.3s<sup>4</sup>.

No entanto, como também se pode ver na figura, a ferramenta avisa que antes do operador **Point** é usual ter-se um operador **Look**. Isto deve-se ao Cogulator ter mais operadores que o KLM básico. Desse modo é possível especificar em mais detalhe as operações, em particular as operações mentais e de percepção realizadas pelos utilizadores.

---

<sup>3</sup>O Cogulator não tem o equivalente ao operador **B** (premir ou largar o botão do rato), mas podemos redefinir a duração do click para durar 0,1s.

<sup>4</sup>Os tempos usados pela ferramenta são ligeiramente diferentes dos definidos na Tabela 1.

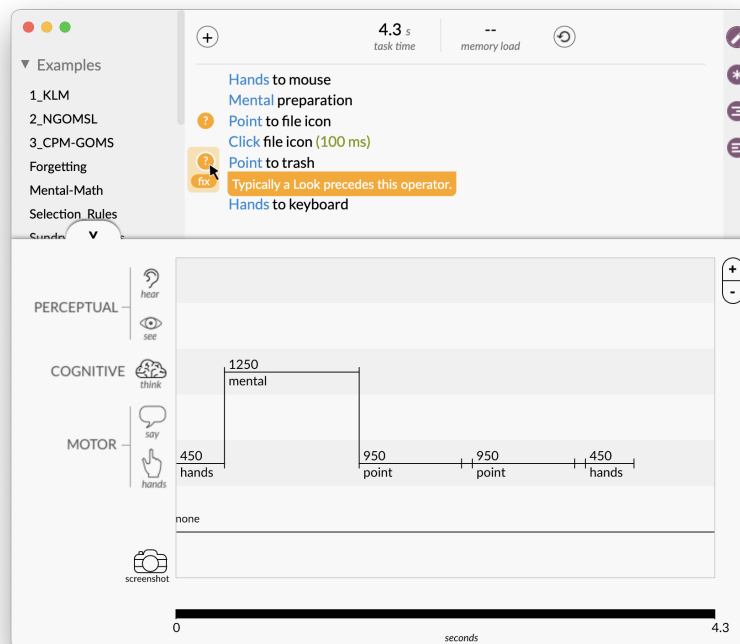


Figura 1: Cogulator

### 3 Exercícios

#### 3.1 KLM

Sugere-se a utilização da ferramenta Cogulator, para a realização dos exercícios desta secção.

1. Considere o exemplo apresentado ao longo da Secção 2. Considere ainda que existe uma barra de menus na janela da aplicação e que no menu Edit, nela existente, existe uma opção para apagar um ficheiro **previamente selecionado**.
  - (a) Determine se a opção de utilizar o menu será mais rápida que a opção de arrastar para o caixote do lixo (nas condições definidas na Secção 2.1).
  - (b) Sabendo que se pretende acrescentar um atalho (^D) à opção de apagar do menu, avalie se a sua implementação tornará a interface mais rápida (nas condições definidas).
  - (c) Com base no resultado, avalie o impacto potencial de utilizar atalhos numa interface.

**BOLETIM DE IDENTIFICAÇÃO DA ENTIDADE**

*NIF:	0	0	0	9	0	0	0	0	0	0
*Nome:										
*Morada:	(Rua, Avª., etc)									
Localidade:										
*Cód.postal:	0	0	0	0	-	0	0	0	0	
*Loc. do CP:										
*R.Finan.:	0	0	0	0	-					
*Telefone:	(pode colocar vários n.º.s)									
Fax:	(pode colocar vários n.º.s)									
*E-mail:										
N.º S.Social:	0	0	0	0	0	0	0	0	0	0
<b>Nº DE IDENTIFICAÇÃO BANCÁRIA (NIB)</b>										
*Banco:	0	0	0	0						
*Agência:	0	0	0	0						
*Conta:	0	0	0	0	0	0	0	0	0	0
*Digitos de Controle:	0	0								
<b>Nº DE IDENTIFICAÇÃO BANCÁRIA ASE (só para Escolas)</b>										
*Banco:	0	0	0	0						
*Agência:	0	0	0	0						
*Conta:	0	0	0	0	0	0	0	0	0	0
*Digitos de Controle:	0	0								

Figura 2: Exemplo de formulário

2. Considere a Figura 2, retirada dos *slides* da primeira aula. Trata-se de uma interface para ser utilizada num *call center*, pelo que a velocidade de execução é um aspecto muito relevante. Considere ainda que está a ser coconsiderada uma interface alternativa, em que os números são preenchidos em campos de *input* textuais (e posteriormente validados pelo sistema). Estima-se que, neste caso, a validação dos números pelo sistema possa demorar 500ms.

- Realize uma análise comparativa das duas interfaces utilizando KLM e assumindo um utilizador "*average non-secretary typist*". Para simplificar a análise, considere apenas o preenchimento de NIF e nome.
- Sabendo que o formulário poderá ter que ser preenchido por clientes, consegue identificar algum potencial problema na introdução do NIB que a passagem para campos de texto irá aumentar? Qual?
- Esse problema existirá caso quem preenche o formulário seja o funcionário do *call center*? Justifique.

3. ★Relembre o sistema que concebeu e desenvolveu durante o trabalho prático

de Desenvolvemento de Sistemas de Software<sup>5</sup>. Analise a interface existente e escolha uma das funcionalidades relativas a jogar. Realize uma análise KLM dessa funcionalidade.

4. Considere o exemplo da inserção de publicações ilustrado nos vídeos fornecidos. Faça uma análise KLM das duas alternativas ilustradas (inserção publicação a publicação vs. inserção de até cinco publicações de cada vez). Para tal, considere:

- a inserção de duas publicações numa lista que já ultrapassa o tamanho do écran;
- uma *scroll constant* **S** de 2,3s (relembre que o botão adicionar está no fim da página);
- os seguintes tamanhos para as strings com que os campos do formulário de cada publicação vão ser preenchidos<sup>6</sup> (uma vez que serão iguais nas duas análises, podem até ser removidas da análise):
  - referência – **T(10)**
  - ano – **T(4)**
  - URL – **T(30)**
  - item bibliográfico – **T(90)**
- que o primeiro campo (referência) fica automaticamente seleccionado e que se pode avançar entre os campos de *input* com Tab;
- que o sistema demora em média 5s a responder – **W(5)**.

Avalie ainda qual seria o impacto de não ter programado a navegação por Tabs e/ou a selecção automática do primeiro campo de texto.

### 3.2 Erros de utilização

1. Em grupo de dois a quatro elementos:

- (a) cada elemento deverá indicar dois ou três erros de utilização que lhe tenham acontecido ou de que tenha conhecimento;
- (b) para cada erro, procurem agora em grupo identificar se se tratou de um *Slip*, um *Lapse*, ou um *Mistake*;
- (c) para cada tipo de erro (*Slip/Lapse/Mistake*) que tenham encontrado, indiquem o exemplos que vos parece mais ilustrativo.

<sup>5</sup>Caso esteja a trabalhar em grupo, seleccionem um dos trabalhos do grupo. Se não tem um sistema funcional, considere o sistema fornecido com esta ficha.

<sup>6</sup>Nos vídeos está a ser utilizado *copy&paste*, pode considerar essa alternativa se preferir.



### 3.3 Perfis do Utilizador

Considere que pretende desenvolver uma aplicação para um bar. Esta deverá permitir gerir desde o atendimento aos clientes nas mesas, até à gestão dos pedidos no balcão.

1. Recorrendo à sua experiência pessoal, identifique quais os tipos de utilizador que deverão ser considerados na aplicação.
2. Trabalhando em grupo, escolham um dos elementos como representativo de um dos tipos de utilizador. Os restantes elementos do grupo deverão então desenvolver o perfil desse tipo (utilize a checklist da página 24 do Módulo 3).
3. Escolha um dos restantes tipos e desenvolva uma *Persona* que caracterize esse tipo de utilizador.