

# Bake-off #2: Seleção em Uls Densas

- **Disponível:** 10 de Março de 2025
- **Entrega:** até às 23h59, dois dias úteis antes do sétimo laboratório (via Fenix)
- **Desafio:** diminuir o tempo de escolha e seleção numa interface densa
- **Resultado esperado:** interface funcional que minimize o tempo de seleção de um alvo numa grelha de 80 alvos
- **Avaliação:** 0-20 valores; 10 valores pela avaliação sumativa (avaliado nos Labs 5 e 6), e 10 valores pelo tempo de seleção médio dos alvos e respetiva taxa de sucesso no *bake-off*

## 1. Desafio

O objetivo do segundo *bake-off* é diminuir o tempo de seleção de alvos numa interface densa (80 alvos). É disponibilizado um [código-fonte](#) em [p5.js](#) que:

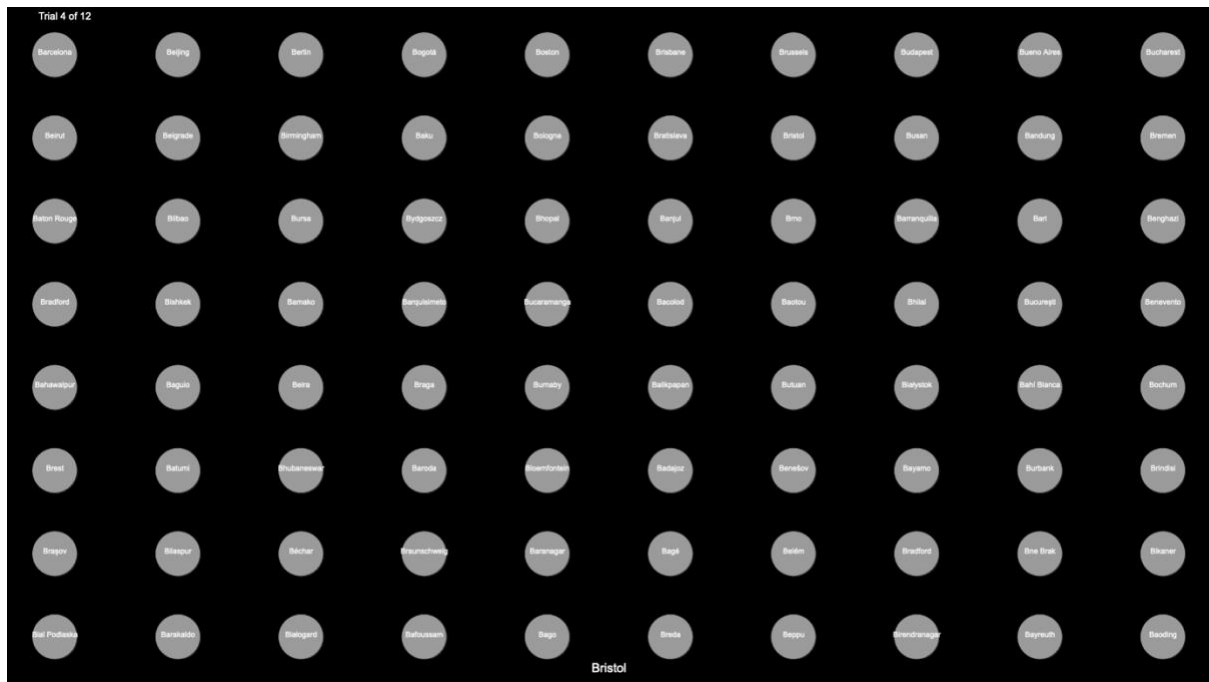
1. Mostra uma grelha de 8x10 alvos aos vossos utilizadores (Figura 1);
2. Identifica cada alvo na grelha com uma label que é associada de forma aleatória no início da execução do programa;
3. Indica qual a legenda do alvo a selecionar;
4. Quantifica o desempenho do utilizador com base na taxa de sucesso (accuracy, 0-100%), tempo total da tarefa (segundos), tempo médio por alvo (segundos), e tempo médio por alvo com penalização (segundos) se a taxa de sucesso do vosso utilizador não for 100%;
5. Guarda estas métricas de desempenho na plataforma [Firebase](#).

Para “vencerem” este *bake-off* têm de alterar o código-fonte de maneira que os vossos utilizadores selecionem os alvos o mais rapidamente possível (atenção à penalização por erros de seleção).

## 2. Funcionamento e Regras

O *bake-off* é um desafio de desenho aberto. É crucial que iniciem um processo iterativo de geração e teste de ideias desde o primeiro dia. **A vossa solução tem de obedecer às seguintes regras:**

1. Não pode assumir conhecimento sobre o alvo a selecionar nem aceder à variável com esta legenda. No entanto, podem assumir conhecimento sobre os alvos já pedidos e selecionados;
2. Não pode modificar o código-fonte que calcula as métricas de desempenho descritas em 1.4, nem o código referente à Firebase em 1.5;
3. Não pode manipular a área nem o conteúdo com a legenda do alvo a selecionar na parte inferior do ecrã;
4. Não pode requerer o uso de hardware adicional para input para além de um rato convencional com dois botões, e de hardware adicional para output para além de auriculares;
5. Não é permitido o input de texto, seja por teclado físico ou virtual;
6. Não pode alterar os ficheiros na pasta “legendas”;



**Figura 1.** A grelha de 8x10 alvos gerada pelo [código-fonte](#) que vos é dado.

7. Têm de garantir que todos os elementos da vossa UI (alvos, etc.) têm o mesmo tamanho em qualquer ecrã. Para isso, vejam com atenção o método `windowResized()` onde calculamos os pixels-per-inch (PPI) e pixels-per-cm (PPCM) do ecrã dos vossos utilizadores.

É crucial que iniciem um **processo iterativo** de ideação-prototipagem-teste desde o primeiro dia. Terão de **iterar pelo menos duas vezes** o protótipo que vos é dado no código-fonte, usando como base: as heurísticas, princípios e fatores humanos descritos nas aulas teóricas e na bibliografia; e nos testes realizados com os vossos utilizadores. Cada bloco da Figura 2 corresponde a uma semana de aulas.

### 3. Recomendações e Recursos

Confirmem com o docente do laboratório se tiverem dúvidas sobre se uma das vossas decisões de desenho quebra alguma das regras descritas em “2. Funcionamento e Tarefas”.

Lembrem-se, o vosso objetivo de desenho é minimizar o tempo de seleção. Vejam com atenção as aulas sobre “Fatores Humanos”, os capítulos 2 (“Nós, os Humanos”) e 9.3 (“Avaliação preditiva”) do livro da cadeira, e recursos online como <https://lawsofux.com/>.

Para os testes estatísticos recomendamos que vejam com atenção a aula teórica sobre “Análise Quantitativa” ou recursos online como as páginas do *Statology*:

- Two sample t-test: <https://www.statology.org/two-sample-t-test/>
- Paired samples t-test: <https://www.statology.org/paired-samples-t-test/>

Para evitarmos problemas de acesso e *hosting* no dia do *bake-off*, têm que desenvolver os vossos protótipos no seguinte editor web: <https://editor.p5js.org/>.

#### 3.1 Tutoriais, exemplos, e referência da linguagem

- Learn: <https://p5js.org/tutorials/>
- Tutoriais vídeo *Coding Train*: <https://www.youtube.com/playlist?list=PLRqwx-V7Uu6Zy51Q-x9tMWlv9cueOFTFA>
- Referência da linguagem: <https://p5js.org/reference/>
- *ShiffBot* (Chrome): <https://shiffbot.withgoogle.com/>
- Exemplos: <https://p5js.org/examples/>
- Bibliotecas: <https://p5js.org/libraries/>

## 4. Planeamento

Podem encontrar o planeamento laboratorial recomendado para este *bake-off* na [página da cadeira](#) e no diagrama seguinte:



Figura 2. Planeamento semanal recomendado para o segundo *bake-off*.

## 5. Dia de *Bake-off* (Lab 7)

O segundo *bake-off* ocorre **exclusivamente no sétimo laboratório**. Cada aluno **testará vários projetos de vários outros turnos**.

É da responsabilidade de cada grupo preparar a solução e o *link* de acesso à aplicação p5.js. Aos alunos pede-se que **não interajam com os autores** dos projetos durante o *bake-off*, que concluem as tarefas **sem distrações e com o máximo de concentração possível**, e que **usem um computador com rato** por uma questão de consistência e justiça dos resultados.

**Comportamentos desonestos (menos éticos) resultam na desqualificação da competição (cotação de 0.0v).** Tempos médios de seleção dois desvios padrões acima ou abaixo da média serão descartados. Alunos com 3 ou mais avaliações descartadas serão penalizados com 0.5v. A mesma penalização será aplicada aos alunos que não completem todas as avaliações atribuídas dentro do tempo de aula.

Reportem algum projeto que **quebre as regras** definidas em “2. Funcionamento e Regras” ao docente do laboratório. A participação no *bake-off* pressupõe o uso de um computador com um **ecrã >= 13"**.

## 6. Submissão

Apenas um membro do grupo terá de realizar a entrega. Esta pressupõe um documento ZIP com o seguinte formato IPM236LXX\_GXX.zip e contendo apenas o conteúdo do vosso projeto p5 (*File > Download*).

## 7. Avaliação

1. **10v, Avaliação sumativa e testes A/B**, demonstrado durante os Labs 5 e 6:
  - O novo protótipo foi testado com **pelo menos 10 utilizadores** (N>=10) – 2v;
  - **Representação A/B eficaz** entre os tempos médios obtidos com o protótipo da semana anterior e o da semana atual (*bar chart*, etc.) – 3v;

- Correto uso e interpretação do **teste estatístico de inferência** para os tempos médios obtidos com o protótipo da semana anterior e o da semana atual (*t-test*, intervalo de confiança, etc.) – 5v.
2. **10v, Tempo médio de seleção (com penalização):** esta componente será calculada através dos resultados dos testes com utilizadores durante o bake-off (Semana 6). A métrica é calculada automaticamente pelo código-fonte fornecido e submetida para uma base de dados (Firebase). O tempo médio de seleção (com penalização) será associado à seguinte nota:
- >4.083s: **0v**
  - ≤4.083s e >3.447s: **2v**
  - ≤3.447s e >2.812s: **4v**
  - ≤2.812s e >2.673s: **6v**
  - ≤2.673s e >2.557s: **8v**
  - ≤2.557s: **10v**

Caso **não submetam o projeto p5.js no Fenix** serão avaliados apenas na primeira componente (7.1 – máx. 10v). Alunos que **não compareçam à sessão do *bake-off*** (Lab 7) terão cotação de 0v na segunda componente (7.2), com exceção de casos com falta justificada (por ex. declaração médica).

Finalmente, grupos que **quebrarem as regras** definidas acima em “2. Funcionamento e Tarefas” terão cotação de 0v na segunda componente da avaliação (7.2).