(Des)ligad@s: adaptação de Atividades Desligadas ao Ensino Remoto de Emergência em tempos de pandemia

Joana Paulo Pardal
Instituto de Educação, Universidade de Lisboa
joana.paulo.pardal@edu.ulisboa.pt

José Cruz
Agrupamento de Escolas de São João da Talha
jcruz@aesjt.pt

José Campos
LASIGE, Faculdade de Ciências, Universidade de Lisboa
jcmcampos@ciencias.ulisboa.pt

João Piedade
UIDEF, Instituto de Educação da Universidade de Lisboa
jmpiedade@ie.ulisboa.pt

Resumo: Neste artigo relatam-se as adaptações à intervenção pedagógica realizada com meia turma da Escola Secundária de São João da Talha, na disciplina de "Programação e Sistemas de Informação" do curso profissional de "Técnico de Gestão e Programação de Sistemas Informáticos", realizada remotamente no 3.º período do ano letivo 2019/20, durante o confinamento a que a pandemia de COVID-19 obrigou. A intervenção decorreu na primeira metade do módulo 4 "Estruturas de Dados Estáticas" ao longo de 9 aulas divididas em 30 minutos síncronos e 60 minutos assíncronos. Exploraram-se várias pistas pedagógicas relativas à aprendizagem inicial da programação, incluindo atividades desligadas e a criptografia. Também se procurou articular corretamente a experiência prévia dos alunos, com atividades desligadas que obrigavam a sistematizar esse conhecimento, e a representação esquemática da memória durante a execução dos programas, para melhorar a compreensão da estrutura de dados estática dos vetores (*arrays*) e os algoritmos de criação, manipulação, pesquisa e ordenação necessários à sua utilização. Em vez de se medirem as melhorias apenas pela análise de código desenvolvido, propõe-se a utilização de exercícios de avaliação de competências de Pensamento Computacional relacionados, minimizando o impacto da escolha da linguagem de programação na possibilidade de expressão do conhecimento dos alunos.

Palavras-Chave: Aprendizagem baseada em Problemas, Atividades Desligadas, Jogo de Fuga Educativo Criptográfico, Estruturas de Dados Estáticas, Introdução à Programação, Pensamento Computacional.

***Abstract**: In this article we report the pedagogical project developed with a half a class from the "São João da Talha's High School", in the subject of "Programming and Information Systems" that took place on the 3rd term of the school year of 2019/20 during the lockdown forced upon us by the COVID-19 pandemic. The topic of the capstone project was the first half of the 4th module on "Static Data Structures" with 9 classes divided into 30-minute synchronous and 60-minute asynchronous. Several pedagogical clues were followed related to the initial learning of programming, including unplugged activities and cryptography. Also, there was proper articulation between the previous experience of the students and a proposed schematic representation of memory state during program execution. On the one hand, there is common sense and unplugged activities that make the students acknowledge what they already know. On the other hand, the memory representation allows us to better understand the arrays static data structures and the related algorithms (creation, manipulation, search and sort). Instead of measuring the improvements only through code analysis, related computational thinking exercises were used to minimize the translation problems that arrive when using programming languages instead of the native one.*

***Keywords:** Problem-based Learning, Unplugged Activities, Cryptographic Escape Room, Static Data Structures, Introduction to Programming (101), Computational Thinking.*

Introdução

A intervenção pedagógica que aqui se relata foi desenvolvida na disciplina de Programação e Sistemas de Informação, no módulo 4 referente a "Estruturas de Dados Estáticas".

A disciplina organiza-se em 3 grandes tópicos: Programação e Algoritmia; Programação com Objetos; e Sistemas de Informação e Bases de Dados. Os primeiros módulos referem-se ao primeiro tópico abordando (1) algoritmos; (2) estruturas de controlo; (3) subprogramas; (**4**, 5 e 6); estruturas de dados estáticas, compostas e dinâmicas; e (7) ficheiros (§ Figura 1).

A intervenção foi preparada tendo em conta esta sequência, consolidando os temas já abordados e ilustrando a necessidade dos temas seguintes.

No módulo da intervenção, estudam-se cadeias de caracteres (*Strings*) e vetores de tipos simples, bem como as operações principais que se realizam com e sobre estas estruturas: declaração, inicialização, manipulação; inserção, pesquisa, remoção; e ordenação (§ Figura 2). Também são estudadas algumas tarefas típicas: procura do maior (ou menor) elemento; escrita de todos ou dos maiores (ou menores) elementos que um dado valor; cálculo da média dos valores; contagem e soma dos valores superiores ou inferiores à média; procura de subsequências; união e interseção; e rotação de elementos.

Figura 1

Conceitos abordados nos módulos iniciais da disciplina de Programação e Sistemas de Informação

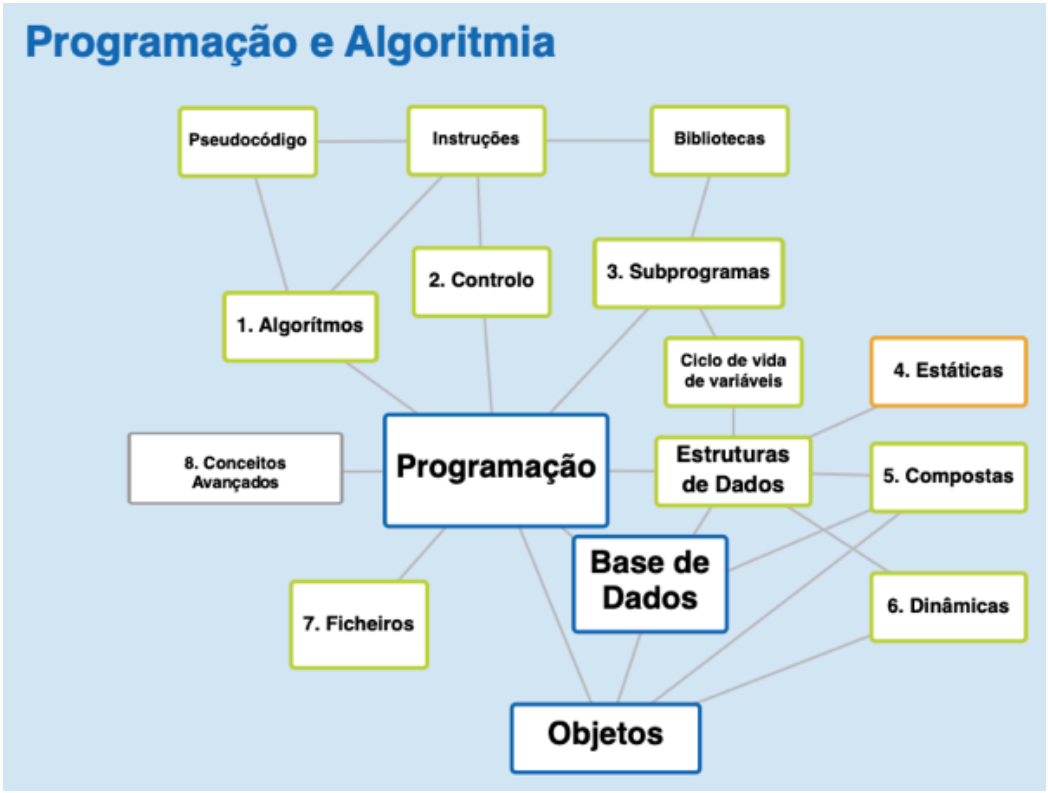


Figura 2

Conceitos abordados no módulo 4 da disciplina

Dificuldades de aprendizagem de Programação

As principais dificuldades na aprendizagem de Programação, de forma geral, incluem a complexidade associada à análise *top-down*, a falta de bons hábitos de estruturação, a má interpretação de especificações, e a falta de teste e *debug* (Ulloa, 1980; Dale, 2006).

Na aprendizagem de estruturas de dados, incluindo vetores, Vrachnos & Jimoyiannis (2017) indicam que os equívocos mais comuns estão na compreensão do conceito de variável.

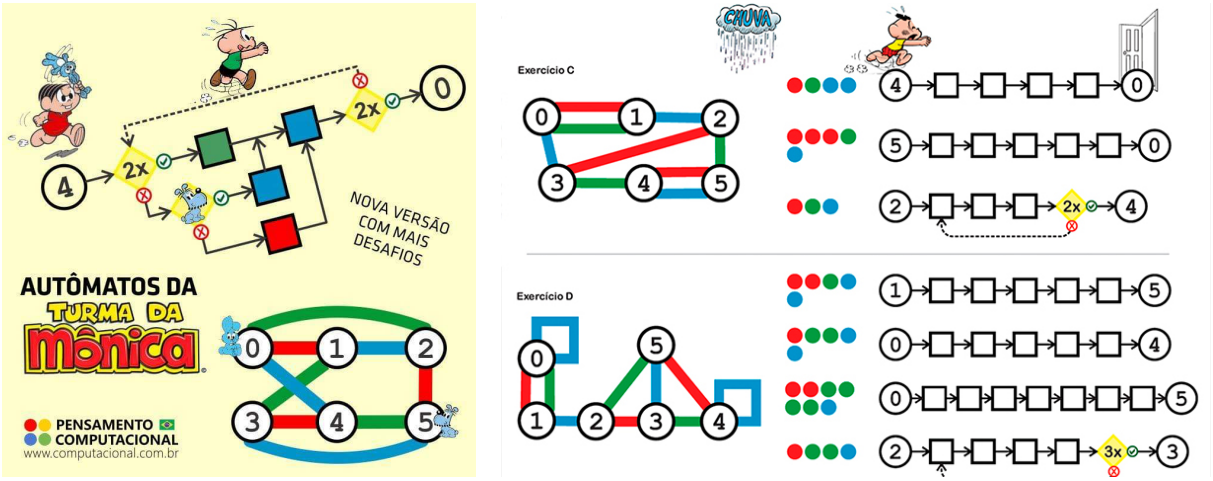
Outra dificuldade é linguística: as linguagens de programação usam inglês estruturado simplificado, mas os alunos não usam aquela língua com proficiência obrigando-os a traduzir do português (em que pensam) para inglês, antes de passarem à linguagem de programação, aumentando a complexidade cognitiva (Baldwin & Macredie, 1999). No primeiro módulo, minimizou-se esta dificuldade utilizando o ***Portugol*** (Manso et al., 2009) que utiliza Português estruturado tornando o ambiente mais amigável (Dasgupta & Hill, 2017). Coloca-se, depois, o desafio da transição para a linguagem de programação (Weintrop & Wilensky, 2019) usada nos restantes módulos.

Atividades Desligadas e seu potencial pedagógico

Com o aumento da importância do ensino da Programação muitas escolas ficam detidas em questões financeiras, nomeadamente na aquisição de computadores. À semelhança dos manipulativos de Matemática (Carvalho et al., 2016), criaram-se soluções económicas que permitem o seu estudo: por exemplo, na Nova Zelândia criaram um conjunto de **atividades *desligadas*** nas quais os alunos utilizam os conceitos do Pensamento Computacional (Wing, 2006) para resolver problemas, sem computadores (Bell et al., 2014), promovendo a oralidade e constituindo uma boa maneira de ensinar e avaliar os raciocínios feitos pelos alunos (Bell & Vahrenhold, 2018). No Brasil, Brackmann (2017) desenvolveu um conjunto de atividades em Português, com personagens de banda desenhada conhecidos (§ Figura 3). Podem, também, ser utilizadas para explicitar o conhecimento previamente adquirido sobre ordenação (Chen et al., 2007), escrevendo os algoritmos na língua materna, mantendo o nível cognitivo mais baixo (Roussel et al., 2017). Além disso, podem ser usadas para medir as capacidades de pensamento computacional dos alunos (Rodriguez et al., 2017).

Figura 3

Atividades desligadas em Português do Brasil: autómatos da Turma da Mónica (capa e exercícios)



Casos Notáveis no Ensino de Programação

Para planificar as aulas, recorreu-se à própria experiência de utilização de atividades desligadas no ensino de programação no 2.º ciclo (Pardal & Arruda, 2020), 1.º ciclo (Pardal & Chambino, 2019) e Pré-Escolar (Pardal & Almeida, 2019); e identificaram-se boas práticas inspiradoras. O curso mais interessante é o CS50 da Universidade de Harvard (Malan, 2021). Este *MOOC* tem um currículo semelhante ao da disciplina. Dirige-se a alunos de todos os cursos explicando os temas a partir da intuição dos alunos e de exemplos de aplicação na vida real (§ Figura 4).

A narrativa utilizada para articular os temas é muito parecida à que se tinha inicialmente planeado, tendo confirmado a orientação do trabalho em desenvolvimento (Malan, 2010).

Figura 4

Atividades desligadas de procura linear e ordenação nas aulas de CS50 de David J. Malan

Fonte: Facebook CS50 Harvard, 2020, fotografias de Arturo J. Real

Ensino Remoto de Emergência: obrigatório ligar!

A introdução aos vários temas seria feita com atividades desligadas diminuindo as distrações inerentes ao uso de computador e a complexidade das linguagens de programação. No entanto, com o confinamento geral que nos surpreendeu a todos, foi necessário adaptar o planeamento original transportando-o para o *online*.

As metodologias e estratégias didáticas utilizadas procuraram envolver o aluno no processo de aprendizagem: utilizou-se ***Aprendizagem baseada em Problemas*** (Kay, et al., 2000), ***Sala de Aula Invertida*** (Kim, 2021), ***Atividades Desligadas*** (Feaster et al., 2011) de ***Pensamento Computacional*** (Rodriguez et al., 2017) e ***Jogos de Fuga Educativos Criptográficos*** (Ho, 2018).

Nas atividades desligadas, em vez de "comandar" algum colega, os alunos deram instruções ao professor que exagerando a sua execução, forçou os alunos a clarificá-las.

O *Moodle* foi complementado pelo *Google Classroom*. As aulas síncronas decorreram pelo *Google Meet*. O editor de código passou de *Visual Studio Express 2015* (instalado nos laboratórios da escola) para ***paiza.io*** (IDE *online*) e deste para ***Repl.it*** (outro IDE *online*).

Este, por incorporar as instruções e testes automáticos, facilitando a consulta aos alunos que assistiam às aulas em telemóveis. Também permite testar o código antes de o submeter para avaliação antecipando algum *feedback*.

As Tabelas 1 e 2 apresentam as atividades inicialmente previstas e as que foram, de facto, desenvolvidas em cada aula, identificando as atividades síncronas e assíncronas.

Foram divididas em dois blocos para acomodar uma questão de calendário.

Desenvolveram-se questionários para os alunos experimentarem as atividades desligadas: pensar nos conceitos antes de os programar. Criou-se uma aplicação para usar os algoritmos de ordenação trocando a posição de elementos dois a dois.

Utilizaram-se atividades da ***Khan Academy*** de pesquisa linear e binária.

Conseguiu-se, assim, desenvolver boa parte das atividades previstas, mas, a componente social, de grupo, não se conseguiu (ainda) recriar, ficando a faltar um certo desafio competitivo.

Tabela 1

Calendarização inicial das aulas de PSI a serem lecionadas durante a intervenção

	Lição	Atividades a Desenvolver	Foco
3.º Período	M4.01 M4.02	Avaliação diagnóstica – compreensão de procura e ordenação (Bebras, nível 1); Atividades desligadas de procura e ordenação – escrita dos algoritmos em português;	
	M4.03 M4.04	Atividades desligadas de criptografia simples – cifras de César, ROT13, de transposição, PigPen, Vigenère, e grelhas; Conversão de binário para `ASCII` (revisão de Arquitetura de Computadores, Módulo 1).	Unplugged
	M4.05 M4.06	Leitura de (bom) código de procura e ordenação; Identificação de algoritmos.	Read
	M4.07 M4.08	Leitura de (bom) código; Representação do estado da memória ao longo da execução dos programas:	
	M4.09 M4.10	declaração de vetores; procura de valores em vetores; ordenação de vetores. Análise Crítica de Complexidade e Eficiência.	Draw
	M4.11 M4.12	Escrita de código que implemente os métodos criptográficos abordados em M4.03 e 04; Análise Crítica de Complexidade e Eficiência.	Write
	M4.13 M4.14	Depuração de Código (próprio e de colegas); Resolução de erros comuns – dificuldades mais frequentes.	Debug
	M4.15 M4.16	`String` como cadeia de caracteres (`char[]`) e o caracter `NULL` (`\0`); Representações alternativas de `String`: tamanho da cadeia de caracteres.	`String`
	M4.17 M4.18	Cadeias de outros tipos de dados (float, long, double); Representação em memória (8, 16, 32 e 64 bits).	Bits & Bytes
	M4.19 M4.20	Vetores alinhadas: representação de informação Vetores de `Strings`, (paralelismo com colunas de Bases de Dados, intuição de estruturas).	`String[]`
	M4.21 M4.22	*Escape Room* Criptográfico Problema de aplicação do conhecimento adquirido; Autoavaliação; Avaliação da intervenção.	Assessment

Nota: As aulas eram de 50 minutos, mas eram dadas em blocos de 100 minutos. Algumas das aulas seriam dadas em dois blocos sucessivos, no mesmo dia, de 200 minutos (1-4 e 7-10).

Tabela 2

Calendarização final das aulas de PSI que foram lecionadas a distância durante a intervenção

	Lição	Atividades a Desenvolver		Foco
		Síncronas (30min)	**Assíncronas** (60min)	
Módulo IIV: Estruturas de Dados Estáticas	M4.1	Kahoot! Avaliação Diagnóstica Cifras Posicionais e de Substituição	Atividade Cifras Posicionais e de Substituição. Resolução de "*Uma Aventura no Palácio das Janelas Verdes*".	***Unplugged***
	M4.2	Revisão das Cifras Algoritmos de Substituição (demonstração)	Implementação da impressão de uma String Implementação da Cifra de César (+1).	Read
	M4.3	String como Lista de Caracteres Impressão de Caracteres da Lista (demonstração)	Implementação da pesquisa e impressão de uma lista utilizando ciclos `for` e o índice da posição.	Write
	M4.4	Pesquisa Linear em vetores Procura em PowerPoint a simular cartas	Implementação de pesquisa linear utilizando ciclos `for`, índices e comparadores.	Draw
	M4.5	Cifras de Substituição Vetores de Strings	Implementação da Cifra de César genérica, substituição de letras com vetor auxiliar.	Write
	M4.6	Sorteio de Chaves inteiras Armazenamento de inteiros em vetores	Reescrever o código do mini projeto do M3 utilizando vetores de inteiros para guardar as variáveis	Improve
	M4.7	*Bubble Sort* desligado Explicitação do algoritmo utilizado intuitivamente	Implementação do algoritmo do *Bubble Sort*. Chamada da função de *Bubble Sort* para ordenar a chave	Write
	M4.8	Implementação colaborativa do algoritmo (correção da tarefa da aula anterior)	Análise de erros das soluções apresentadas (verificação). Passo de otimização de parar quando já não há mudanças.	Debug & Optimize
	M4.9	*Exit Room*: aplicação de todo o conhecimento adquirido na resolução de um jogo	Avaliação da Aprendizagem desenvolvida, da qualidade da intervenção, e da adequação dos métodos.	Assessment

Nota: A divisão entre as aulas 5 e 6 corresponde à interrupção dos feriados de Junho de 2020: 10 de Junho, dia de Portugal, de Camões e das Comunidades Portuguesas, e dia 11 de Junho, Corpo de Deus, que nesse ano ocorreram na mesma semana, em dias seguidos.

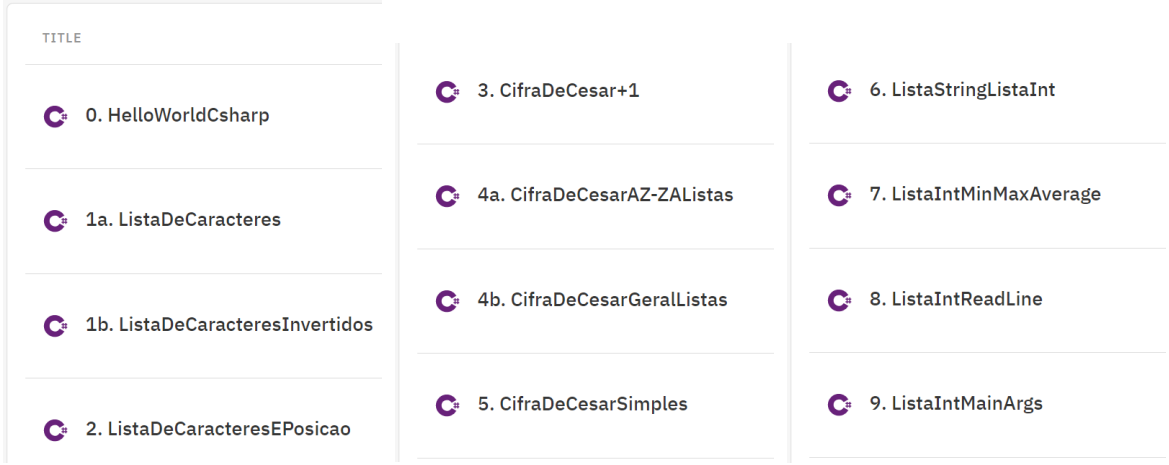
Metodologia

Os conhecimentos dos alunos foram avaliados no início da intervenção com um teste de escolha múltipla, baseado num desenvolvido e validado por Costa & Miranda (2017) para avaliação das competências iniciais de programação, no ***Kahoot!***. No final, repetiram-se as perguntas para aferir a evolução dos alunos, mas agora no ***Quizizz*** (§ Tabela 4).

Nos tempos assíncronos, os alunos deveriam responder a desafios de programação no ***Repl.it***, onde manipulavam *Strings* e vetores de inteiros (§ Figura 5).

Figura 5

Lista das tarefas disponibilizadas aos alunos no Repl.it.



Nota: As tarefas dividem-se em Cifras de César e manipulação de vetores de inteiros.

Resultados e Discussão

A Tabela 3 apresenta as perguntas feitas aos alunos no pré-teste, a média por pergunta, e as respostas do pós-teste, também acompanhadas da média por pergunta.

A Tabela 4 apresenta as médias obtidas nos vários momentos. No primeiro questionário alcançaram uma média de 39% de acerto. No final da aula foram incentivados a repetir o exercício para estudarem as questões referentes aos módulos anteriores. No tempo de trabalho autónomo, a maior parte dos alunos repetiu o questionário e melhorou os seus resultados.

No questionário final, os alunos responderam às mesmas perguntas, tendo a maioria dos alunos melhorado o seu desempenho, embora não tanto como nos treinos.

Observando as respostas dadas por cada aluno, vemos que apesar de, na maior parte dos casos, os alunos melhorarem em média, isso é feito sem manter os acertos iniciais: os alunos acertaram um número maior de perguntas, mas erraram perguntas que tinham acertado anteriormente.

Tabela 3

Respostas dos alunos para cada pergunta no pré-teste e no pós-teste.

Pergunta	Respostas A B G I J L P Q R T	%
1. Quando quero declarar um valor que NÃO MUDA durante a execução do programa, uso uma ... **constante**	☑ ☒ ☑ ☒ ☒ ☒ ☒ ☐ ☒ ☒	20%
	☑ ☒ ☑ ☑ ☑ ☑ ☑ ☒ ☐ ☒	↗ 60%
2. Quando quero declarar um valor que MUDA ao longo da execução do programa, uso uma ... **variável**	☒ ☒ ☑ ☑ ☑ ☑ ☒ ☑ ☑ ☒	60%
	☑ ☑ ☑ ☑ ☑ ☑ ☒ ☒ ☐ ☒	→ 60%
3. Quando separo uma parte do código para evitar duplicação de código, uso uma... **função**	☒ ☑ ☑ ☒ ☒ ☑ ☑ ☒ ☒ ☑	50%
	☑ ☒ ☒ ☒ ☑ ☑ ☒ ☒ ☐ ☒	↘ 30%
4. A condição "if ... then ... else ..." é uma estrutura de... **decisão**	☒ ☒ ☒ ☒ ☒ ☑ ☑ ☒ ☒ ☑	30%
	☒ ☑ ☑ ☒ ☑ ☒ ☑ ☒ ☐ ☒	↗ 40%
5. Quando quero decidir entre duas possibilidades, uso a estrutura... **if (...) then (...) else (...)**	☒ ☒ ☑ ☒ ☒ ☑ ☑ ☒ ☑ ☑	50%
	☑ ☒ ☒ ☒ ☑ ☒ ☑ ☑ ☐ ☑	→ 50%
6. Quando quero decidir entre várias possibilidades (escolha múltipla), uso a estrutura... **switch (...) { case ...: .... break;}**	☒ ☒ ☑ ☒ ☒ ☒ ☑ ☒ ☒ ☒	20%
	☑ ☒ ☑ ☑ ☒ ☒ ☑ ☒ ☐ ☒	↗ 30%
7. Quando quero receber informação do utilizador, uso... **System.Console.ReadLine**	☒ ☒ ☑ ☒ ☒ ☒ ☑ ☒ ☑ ☒	30%
	☒ ☒ ☑ ☑ ☑ ☑ ☑ ☑ ☐ ☒	↗ 60%
8. Quando quero apresentar informação ao utilizador, uso... **System.Console.WriteLine**	☒ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☒	80%
	☒ ☑ ☑ ☑ ☑ ☑ ☑ ☑ ☐ ☒	↘ 70%
9. Quando quero repetir uma instrução um número certo de vezes, uso... **for (...; ...; ...) { ... }**	☑ ☒ ☑ ☒ ☑ ☒ ☑ ☑ ☑ ☒	60%
	☒ ☒ ☑ ☒ ☒ ☒ ☑ ☒ ☐ ☑	↘ 30%
10. Quando quero repetir uma instrução enquanto se verificar uma condição, uso... **while (...) { ... }**	☑ ☑ ☒ ☒ ☑ ☑ ☒ ☒ ☒ ☒	40%
	☒ ☑ ☒ ☒ ☑ ☒ ☑ ☑ ☐ ☑	↗ 50%
11. Quando quero repetir uma instrução até que uma condição deixe de se verificar, uso... **do { ... } while (...)**	☒ ☑ ☑ ☑ ☑ ☒ ☑ ☒ ☑ ☒	60%
12. Quais as estruturas de decisão que posso utilizar? **if (...) then {...} else {...}** | **switch (...) {case ...: .... break;}**	☐ ☒ ☑ ☐ ☐ ☒ ⊡ ☒ ⊡ ☒	10%
	☒ ☑ ☒ ☒ ⊡ ☑ ☒ ☒ ☐ ☑	↗ 30%
13. As estruturas de repetição que posso usar são... **while (...) {...}** | **for (...; ...; ...) {...}**	☒ ☑ ☑ ☐ ⊡ ⊡ ☒ ☒ ⊡ ☒	20%
	☒ ☒ ☑ ☒ ☒ ☑ ☑ ☒ ☐ ☒	↗ 30%
14. Qual a ordem correta das seguintes linhas de código? **System.Random rand = new Random(); int n = rand.Next(1,50); if (n > 10) System.Console.Write(" {0}", n);**	☒ ☑ ☒ ☒ ☒ ☒ ☑ ☒ ☒ ☒	20%

*Nota: As respostas podem estar: **correta**, **parcialmente correta**, **errada**, **não respondida**. A pergunta 11 não foi repetida no pós-teste porque os alunos não tinham usado a estrutura de controlo "do... while..."; e a pergunta 14 porque não havia este tipo de perguntas na nova ferramenta.*

Tabela 4

Médias dos alunos no pré-teste, nos testes intermédios (de treino e estudo) e no pós-teste.

Alunos / Atividade	A	B	G	I	J	L	P	Q	R	T	
Kahoot! (não respondidas)	25% (1)	33% (0)	83% (0)	17% (2)	33% (2)	50% (0)	58% (0)	25% (3)	42% (0)	25% (0)	Pré-teste*
Repetições I	✗	57%	7%	✗	✗	64%	71%	36%	✗	36%	Treino
Repetições II	86%	✗	79%	✗	✗	86%	79%	64%	✗	100%	
Quizizz	↗ 42%	↗ 42%	↘ 67%	↗ 42%	↗ 67%	↗ 58%	↗ 75%	↗ 33%	✗	↗ 33%	Pós-Teste*

*Nota: (*) As médias foram calculadas com as 12 perguntas comuns. O pré-teste foi realizado no **Kahoot!**. Já o pós-teste foi realizado no **Quizizz** para ultrapassar uma limitações da aplicação: à época o **Kahoot!** não apresentava as perguntas aos participantes, o que favorecia os alunos com dois dispositivos, que usavam um para ver as perguntas projetadas e as alternativas, e o outro para responder. O **Quizizz** mostra as perguntas e as alternativas de resposta ao mesmo tempo. Entretanto o **Kahoot!** já tem esta opção.*

Considerações finais

Neste artigo, apresentaram-se as alterações necessárias para responder aos desafios do ensino remoto de emergência trazidos pelo confinamento da Primavera de 2020. Converteram-se as aulas de 50 minutos presenciais (lecionadas em blocos de 100 ou 200 minutos), em aulas a distância (divididas em 30 minutos síncronos e 60 assíncronos).

Apesar das circunstâncias, os alunos foram capazes não só de aprender novos conteúdos, mas também de melhorar os conteúdos dos módulos anteriores.

O uso de atividades desligadas, mesmo a distância, trouxe benefícios à aprendizagem, reconhecidos pelos próprios alunos como momentos "fora da caixa" de aprendizagem significativa.

Para se poder avaliar melhor a possibilidade de utilizar as atividades desligadas e os seus métodos em ambientes de educação a distância será preciso: uma amostra maior (a turma era pequena, tendo apenas 10 alunos na disciplina); um prazo mais alargado de utilização das ferramentas; e considerar o ensino a distância não como uma resposta de emergência, mas como método principal, desde o início do planeamento.

Também será interessante explorar novas ferramentas de comunicação como o ***gather.town*** que permite o trabalho colaborativo e a supervisão de novas formas (Latulipe, 2021).

Referências Bibliográficas

Baldwin, L. P., & Macredie, R. D. (1999). Beginners and Programming: insights from Second Language Learning and Teaching. *Education and Information Technologies, 4*(2), pp. 167-179. doi.org/10.1023/A:1009652001566

Bell, T., & Vahrenhold, J. (2018). CS Unplugged - How is it used, and does it work? Em H.-J. Böckenhauer, D. Komm, & W. Unger, *Adventures Between Lower Bounds and Higher Altitudes* (Vol. 11011, pp. 497-521). Springer, Cham. doi.org/10.1007/978-3-319-98355-4_29

Bell, T., Newton, H., Duncan, C., & Jarman, S. (2014). Adoption of Computer Science in NZ schools. hdl.handle.net/10092/10626

Brackmann, C. (2017). *Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica.* Porto Alegre, RS, Brasil: Universidade Federal do Rio Grande do Sul (UFRGS). hdl.handle.net/10183/172208

Carvalho, A., Santos, C., Silva, J., & Teixeira, R. (2016). Materiais Estruturados para a Educacão Matemática Pré-Escolar. *Jornal das Primeiras Matemáticas,* 7, 27–76. hdl.handle.net/10400.3/4214

Chen, T.-Y., Lewandowski, G., McCartney, R., Sanders, K., & Simon, B. (2007). Commonsense computing: using student sorting abilities to improve instruction. *Proceedings of the 38th technical symposium on Computer Science Education* (pp. 276-280). ACM SIGCSE. doi.org/10.1145/1227310.1227408

Costa, J. M., & Miranda, G. L. (2017). Desenvolvimento e validação de uma prova de avaliação das competências iniciais de programação. *Revista Ibérica de Sistemas e Tecnologias de Informação, 25*, 66–81. doi.org/10.17013/risti.25.66-81

Dale, N. B. (2006). Most difficult topics in CS1: results of an online survey of educators. *InRoads, 38*(2), pp. 49–53. doi.org/10.1145/1138403.1138432

Dasgupta, S., & Hill, B. M. (2017). Learning to code in localized programming languages. *Proceedings of the 4th Conference on Learning@Scale* (pp. 33-39). ACM. doi.org/10.1145/3051457.3051464

Feaster, Y., Segars, L., Wahba, a. K., & Hallstrom, J. O. (2011). Teaching CS Unplugged in the High School (with limited success). *Proceedings of the 16th annual joint conference on Innovation and Technology in Computer Science Education*, (pp. 248-252). doi.org/10.1145/1999747.1999817

Ho, A. M. (2018). Unlocking Ideas: Using Escape Room Puzzles in a Cryptography Classroom. *Problems, Resources, and Issues in Mathematics Undergraduate Studies, 28*(9), 835–847. doi.org/10.1080/10511970.2018.1453568

Kay, J., Barg, M., Fekete, A., Greening, T., Hollands, O., Kingston, J. H., & Crawford, K. (2000). Problem-based learning for foundation Computer Science courses. *Computer Science Education, 10*(2), 109–128. doi.org/10.1076/0899-3408(200008)10:2;1-C;FT109

Kim, J. A. (2021). Flipped Learning and Unplugged Activities for Data Structure and Algorithm Class. *Software Engineering in IoT, Big Data, Cloud and Mobile Computing* (pp. 93–02). Cham: Springer. doi.org/10.1007/978-3-030-64773-5_8

Latulipe, C. (2021). A CS1 Team-Based Learning Space in Gather.Town. *52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21)* (p. 1245). New York, NY, USA: Association for Computing Machinery. doi.org/10.1145/3408877.3439587

Malan, D. J. (2010). Reinventing CS50. *Proceedings of the 41st ACM technical symposium on Computer Science Education* (pp. 152-156). Milwaukee, Wisconsin, USA: ACM SIGCSE. doi.org/10.1145/1734263.1734316

Malan, D. J. (2021). *This is CS50*. Harvard University: cs50.harvard.edu

Manso, A., Oliveira, L., & Marques, C. G. (2009). Ambiente de aprendizagem de algoritmos – Portugol IDE. *Conferência Internacional de TIC na Educação*, *6.* orion.ipt.pt/~manso/papers/2009/Portugol%20Challenges2009.pdf

Pardal, J., & Almeida, C. (2019). C0d≡2CR8: Programar para Criar no Ensino Pré-Escolar. *V Seminário Internacional de Práticas Pedagógicas Inovadoras (SIPPI).* Curitiba, Paraná: Editora Positivo. www.slideshare.net/joana.paulo.pardal/sippi-2019-c0de2cr8-prescolar

Pardal, J., & Arruda, A. (2020). C0d≡2CR8: Programar para criar no 2.º ciclo. *VI Seminário Internacional de Práticas Pedagógicas Inovadoras (SIPPI).* Curitiba, Paraná: Editora Positivo. www.slideshare.net/joana.paulo.pardal/sippi-2020-c0de2cr8-2-ciclo

Pardal, J., & Chambino, A. R. (2019). C0d≡2CR8: Programar para criar no 1.º ciclo. *V Seminário Internacional de Práticas Pedagógicas Inovadoras (SIPPI).* Curitiva, Paraná: Editora Positivo. www.slideshare.net/joana.paulo.pardal/sippi-2019-c0de2cr8-1-ciclo

Picado, J., & Martins, P. M. (2014). Matemática Recreativa - Cinco tributos a Martin Gardner. *Boletim da Sociedade Portuguesa de Matemática, 71*, pp. 97–111. revistas.rcaap.pt/boletimspm/article/view/6806/5047

Rodriguez, B., Kennicutt, S., Rader, C., & Camp, T. (2017). Assessing Computational Thinking in CS Unplugged activities. *Proceedings of the Technical Symposium on Computer Science Education* (pp. 501-506). ACM SIGCSE. doi.org/10.1145/3017680.3017779

Roussel, S., Joulia, D., Tricot, A., & Sweller, J. (2017). Learning subject content through a foreign language should not ignore human cognitive architecture: A cognitive load theory approach. *Learning and Instruction, 52*, 69-79. doi.org/10.1016/j.learninstruc.2017.04.007

Ulloa, M. (1980). Teaching and Learning Computer Programming: A Survey of Student Problems, Teaching Methods, and Automated Instructional Tools. *ACM SIGCSE Bulletin, 12*(2), pp. 48–64. doi.org/10.1145/989253.989263

Vrachnos, E., & Jimoyiannis, A. (2017). Secondary education students' difficulties in algorithmic problems with arrays: An analysis using the SOLO taxonomy. *Themes in Science and Technology Education, 10*(1), 31-52. earthlab.uoi.gr/ojs/theste/index.php/theste/article/view/238

Weintrop, D., & Wilensky, U. (2019). Transitioning from introductory block-based and text-based environments to professional programming languages in high school computer science classrooms. *Computers & Education, 142*(103646), 1-17. doi.org/10.1016/j.compedu.2019.103646

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. doi.org/10.1145/1118178.1118215