



ÁRLLON CHAVES LIMA

ORIENTAÇÃO: PROF. DR. MARCOS MONTEIRO DINIZ

COORIENTAÇÃO: PROF^a. DRA MARIANNE KOGUT ELIASQUEVICI

METODOLOGIA

70S

O ENSINO E A APRENDIZAGEM MAIS SIGNIFICATIVOS DE ALGORITMOS

ELABORAÇÃO E AUTORIA:

Árllon Chaves Lima

ORIENTAÇÃO:

Prof. Dr. Marcos Monteiro Diniz

COORIENTAÇÃO:

Profa. Dra. Marianne Kogut Eliasquevici

IDENTIDADE VISUAL:

Andreza Jackson de Vasconcelos

DIAGRAMAÇÃO:

Adriane Jackson de Vasconcelos

REVISÃO TEXTUAL:

Betty Vibranovsky

APOIO:

Universidade Federal do Pará (UFPA)

Núcleo de Inovação em Tecnologias Aplicadas a Ensino e Extensão (NITAE²)

Programa de Pós-Graduação Criatividade e Inovação em Metodologias de Ensino Superior (PPGCIMES)

Ano: 2020. Edição 1

Um convite a vocês professores(as)

Quem de vocês, professores(as) da área de Tecnologia e Informação (TI), nunca passou por uma situação em que, ao final do semestre letivo de disciplinas voltadas para introdução à programação, tal como Algoritmos, um quantitativo expressivo de alunos ficou reprovado ou passou sem ter conseguido aprender o conteúdo de forma significativa? Pois é, saibam que vocês não estão sozinhos(as).

Para ajudar a mudar essa situação, convidamos vocês, professores(as) de disciplinas voltadas à construção de algoritmos, a ler este material, que descreve a Metodologia 7Cs, concebida durante a pesquisa de mestrado do discente Árlon Chaves Lima, sob orientação do professor Dr. Marcos Monteiro Diniz e coorientação da professora Dra. Marianne Kogut Eliasquevici, no Programa de Pós-Graduação Criatividade e Inovação em Metodologias de Ensino Superior (PPGCIMES)¹.

O objetivo deste material é estimular o uso de uma nova metodologia que poderá tornar a aprendizagem dos seus alunos mais significativa. A partir da leitura, pretendemos incentivar a prática de construção de algoritmos, visando ao desenvolvimento de habilidades e competências importantes para um bom rendimento de seus alunos na disciplina Algoritmos ou similar.

Ao final, caso experimentem a Metodologia 7Cs em aula, pedimos que relatem os resultados em nosso grupo no Facebook, o que nos ajudará a aprimorá-la e compartilhá-la, permitindo-nos alcançar, dessa forma, outros(as) professores(as) que estejam em busca de minimizar as dificuldades presentes na disciplina Algoritmos e inovar em sala de aula.

Ansiosos(as) para explorar a Metodologia 7Cs?

Sejam bem-vindos(as)!

¹ O PPGCIMES é uma subunidade acadêmica do Núcleo de Inovação e Tecnologias Aplicadas a Ensino e Extensão (NITAE²), da Universidade Federal do Pará (UFPA).

SUMÁRIO

I NOVA METODOLOGIA, NOVAS POSSIBILIDADES.....	5
II UM MERGULHO NA METODOLOGIA 7CS.....	8
2.1 A teoria inspiradora: aprendizagem significativa e ancoragem de conceitos.....	9
2.2 Descrevendo a Metodologia 7Cs	10
III A METODOLOGIA 7CS EM AÇÃO.....	15
3.1 PontapéInicial.....	17
3.2 Explorando as fases.....	18
3.2.1 Descoberta.....	18
3.2.2 Aperfeiçoamento.....	20
3.2.3 Consolidação	25
IV SUGESTÃO DE LISTAS	29
4.1 Lista 1: estrutura de repetição.....	30
4.2 Lista 2: matrizes.....	33
MÃOS À OBRA.....	37
REFERÊNCIAS	39
APÊNDICE I- GABARITO.....	40

I NOVA METODOLOGIA, NOVAS POSSIBILIDADES



Professores(as), vocês sabem o quanto a disciplina Algoritmos é importante, principalmente em cursos na área de TI, pois é base para outras disciplinas de programação, ainda mais complexas. Para Barcelos *et al.* (2009), o estudo de algoritmos é uma área fundamental no campo da Computação, pois o conhecimento e a compreensão desse conteúdo permitirá a aprendizagem nas linguagens de programação, que em sua essência possuem estratégias de solução de problemas.

A disciplina exige habilidades e competências, tais como raciocínio lógico, leitura e interpretação, capacidade de abstração, domínio de conceitos matemáticos, resolução de problemas, entre outras (RAABE e SILVA, 2005; GOMES e MENDES, 2007; PIVA JÚNIOR e FREITAS, 2010), de maneira integrada, o que não é percebido em grande parte dos alunos, desde a educação básica, interferindo diretamente na aprendizagem e na compreensão dos conteúdos ministrados por vocês.

Confesso a vocês que na graduação em Licenciatura em Computação, realizada na Universidade Federal Rural da Amazônia (UFRA), enfrentei grandes desafios na disciplina Algoritmos. Era algo totalmente novo, que não fazia parte dos conhecimentos que havia construído até ali. Sentia-me muitas vezes frustrado e desmotivado a cada erro cometido na solução dos problemas que me eram postos, o que piorava quando o grau de complexidade dos problemas aumentava. Assim como eu, grande parte dos alunos egressos do ensino médio regular também não possui base técnica de lógica ou algoritmos, o que reflete no aumento da dificuldade de aprendizado na disciplina (VIEIRA; LIMA JUNIOR; VIEIRA, 2015).

Então, imaginem: se seus alunos têm dificuldade em assimilar os conteúdos ministrados e por isso não conseguem obter um bom rendimento, provavelmente terão dificuldade ao longo do curso, gerando um sentimento de desmotivação e frustração, que muitas vezes provoca altos índices de reprovação e evasão. Conforme Giraffa e Mora (2013), a evasão em disciplinas dos níveis iniciais dos cursos de Computação não é um tema novo, é algo recorrente e que vem sendo muito estudado, tanto em nível nacional quanto internacional.



Acreditamos que vocês, quando alunos, tenham passado por situação semelhante. Deve ser incômodo saber que ministram uma

disciplina em que é comum ocorrer altos índices de reprovação. Por isso, imaginamos que vocês já devem ter se perguntado: “Como posso minimizar as dificuldades dos meus alunos?”.

Não existe uma resposta certa ou algo que possa solucionar todos os seus problemas. Mas existe a possibilidade de criar novos caminhos e estratégias de ensino que busquem facilitar a compreensão dos conteúdos da disciplina, tornando a aprendizagem mais significativa. As dificuldades que senti durante meus estudos, somadas às leituras realizadas e aos conhecimentos adquiridos no mestrado, causaram-me grande inquietação, o que me levou a trabalhar em um projeto de dissertação voltado ao ensino e à aprendizagem de algoritmos.

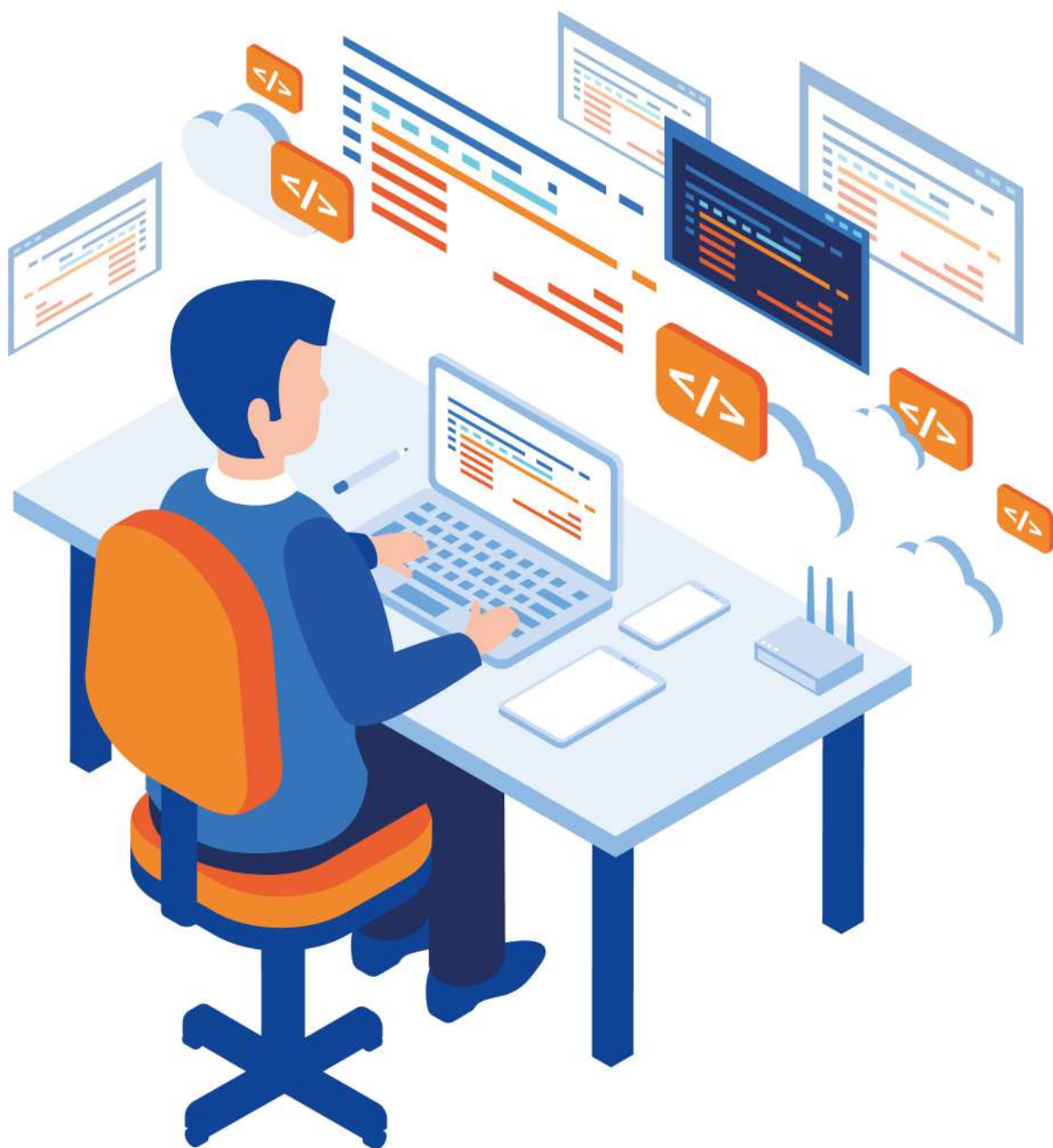
Para isso, utilizamos como base teórica principal a Teoria da Aprendizagem Significativa de Ausubel (2000), que consiste basicamente em aprender por meio da ancoragem de conhecimentos, a qual fundamenta e estrutura a **Metodologia 7Cs**.

Assim, pensando em vocês, professores(as), é que produzimos este material em que é possível encontrar uma explicação da **Metodologia 7Cs**, a base teórica que a sustenta, uma proposta de aplicação com algumas sugestões, além de dois exemplos de listas de exercícios.

Desejamos uma boa leitura do material e esperamos que seja útil em sua atuação docente.

II

UM MERGULHO NA METODOLOGIA 7CS



Neste tópico do material, vocês vão conhecer a Metodologia 7Cs com todas as suas particularidades.



A teoria inspiradora: aprendizagem significativa e ancoragem de conceitos

A Teoria da Aprendizagem Significativa de Ausubel (2000) é a base que fundamenta e estrutura a Metodologia 7Cs. Ausubel verificou que a abordagem tradicional de ensino era pouco produtora e que os conceitos adquiridos pelos alunos, na verdade, não eram interativos; por isso, não construídos e internalizados.

Muito provavelmente, vocês já devem ter passado por experiências em que um conhecimento ou vivência anterior serviu para que compreendessem e/ou aprendessem algo novo. Pois é, essa é a base da Teoria da Aprendizagem Significativa.

Para Ausubel (2000), os indivíduos só aprendem significativamente quando utilizam conteúdos existentes em sua estrutura cognitiva, chamados por ele de subsunçores, que servem como âncoras para interação entre os novos e os conhecimentos anteriores.

A teoria de Ausubel (2000) subdivide-se em três categorias (Subordinada, Superordenada e Combinatória), apresentadas no Quadro 1 a seguir.

Quadro 1 - Categorias de aprendizagem

CATEGORIAS	DESCRIÇÃO
SUBORDINADA	Na aprendizagem subordinada, o novo conhecimento passa a interagir com os subsunçores (conhecimentos preexistentes do indivíduo) para tornar o novo conhecimento cheio de significados.
SUPERORDENADA	Na aprendizagem superordenada, a partir dos subsunçores adquiridos, o indivíduo consegue concretizar uma ideia mais geral e organizar os conhecimentos preexistentes como parte dessa ideia.
COMBINATÓRIA	Na aprendizagem Combinatória, ocorre a união da aprendizagem subordinada e superordenada, o que torna o conhecimento mais sólido.

Fonte: (AUSUBEL, 2000).

CURIOSIDADE:



David Paul Ausubel nasceu em 25 de outubro de 1918 no Brooklyn, New York e morreu em 9 de julho de 2008. Ele se baseia na perspectiva cognitivista para explicação do processo de aprendizagem, em que os conceitos armazenados pelos indivíduos podem ser manipulados e usados como forma de conexão e integração de informações que facilitam a compreensão de novos conhecimentos (MOREIRA; MASINI, 1982).

Acreditamos que vocês devem estar se perguntando como a teoria se relaciona com a Metodologia 7Cs.

Cada uma das categorias propostas por Ausubel (2000) serviu de inspiração para a divisão da Metodologia 7Cs em três fases (Descoberta, Aperfeiçoamento e Consolidação), apresentadas e melhor detalhadas no próximo tópico.



Descrevendo a Metodologia 7Cs

Podemos dizer a vocês que a Metodologia 7Cs busca transformar o modo de praticar a resolução de problemas algorítmicos, de forma a desenvolver no aluno habilidades e competências a fim de que seja minimizada sua dificuldade em compreender os conteúdos da disciplina Algoritmos.

Essa metodologia contém em sua estrutura sete dimensões (Compreender, Conceber, Completar, Compatibilizar, Corrigir, Construir e Criar), e cada uma delas possui um objetivo, visando traçar um caminho para a construção do conhecimento sobre os conteúdos.

Veja, no Quadro 2, a descrição dessas dimensões propostas, com seus objetivos e habilidades e competências que se pretende que os alunos desenvolvam. Perceba a presença da ação **Colaborar**, prevista por nós para estimular, em todas as dimensões o emprego de atividades colaborativas. Segundo Moreira (2011, p. 50), a colaboração potencializa a aprendizagem significativa por viabilizar a troca de informações e a negociação dos significados.

Quadro 2 - Descrição da Metodologia 7Cs.

C O L A B O R A R	DIMENSÃO	DESCRIÇÃO	OBJETIVOS	HABILIDADES E COMPETÊNCIAS	PRÉ-REQUISITOS
	COMPREENDER	Os alunos devem compreender um novo conhecimento, por meio da interpretação e análise de algoritmos simples.	Estimular a capacidade de interpretar trechos de algoritmos ou algoritmos inteiros simples, já construídos.	<ul style="list-style-type: none"> •Compreender e interpretar os funcionamentos da estrutura de um algoritmo; •Desenvolver a capacidade de solução de problemas. 	Conhecer os conceitos básicos da estrutura a ser trabalhada.
	CONCEBER	Após passar pelo processo de compreensão, os alunos concebem seus primeiros algoritmos, a partir de situações problemas dadas.	Estimular a capacidade de solucionar problemas, ao conceber seus primeiros algoritmos.	<ul style="list-style-type: none"> •Interpretar os comandos das questões. •Desenvolver a capacidade de solução de problemas. 	<p>Passar pela dimensão Compreender.</p> <p>Conhecer minimamente a linguagem de programação a ser utilizada pelo professor.</p>
	COMPLETAR	Partindo do princípio de que os alunos compreenderam o conteúdo e que foram capazes de conceber seus primeiros algoritmos, serão capazes de analisar um algoritmo incompleto e perceber e completar o está faltando para ser executado.	Estimular a capacidade de analisar um algoritmo já construído, para identificar os elementos que faltam para se tornar executável.	<ul style="list-style-type: none"> •Interpretar os comandos das questões e códigos. •Desenvolver a capacidade de percepção de erros e de ajuste de algoritmo. 	<p>Passar pelas dimensões Compreender e Conceber.</p> <p>Saber os elementos necessários para execução de um algoritmo.</p>
	COMPATIBILIZAR	Os alunos devem, a partir de um algoritmo apresentado com a sequência de passos de forma desordenada, identificar quais comandos devem ser realocados e compatibilizá-los, para que possa ser executável.	Estimular a capacidade de organização lógica e encadeamento dos passos na construção de um algoritmo.	<ul style="list-style-type: none"> •Interpretar os comandos das questões. •Desenvolver a capacidade de organizar o encadeamento das ideias. •Desenvolver a capacidade de ajuste de algoritmo. 	<p>Passar pelas dimensões Compreender, Conceber e Completar.</p> <p>Saber os elementos necessários para execução de um algoritmo.</p>

Fonte: Elaborado pelo autor.

Quadro 2 - Descrição da Metodologia 7Cs (Cont.).

C O L A B O R A R	DIMENSÃO	DESCRIÇÃO	OBJETIVOS	HABILIDADES E COMPETÊNCIA	PRÉ-REQUISITOS
	CORRIGIR	Os alunos terão que analisar e corrigir algoritmos implementados com erros diversificados.	Estimular a capacidade de analisar e perceber erros em algoritmos já construídos.	<ul style="list-style-type: none"> • Interpretar os comandos das questões. • Desenvolver a capacidade de solução de problema. • Construir algoritmos com eficiência 	<p>Passar pelas dimensões Compreender, Conceber, Completar e Compatibilizar.</p> <p>Saber os elementos necessários para execução de um algoritmo.</p> <p>Saber os elementos necessários para execução de um algoritmo.</p>
	CONSTRUIR	Os alunos devem utilizar todos os conhecimentos já adquiridos para construir a solução de problemas mais complexos.	Estimular a solução de problemas que necessitam da construção de algoritmos com nível de complexidade maior.	<ul style="list-style-type: none"> • Aprender de maneira colaborativa. • Potencializar a solução de problemas. • Estimular o processo criativo. 	<p>Passar pelas dimensões Compreender, Conceber, Completar e Compatibilizar e Corrigir.</p> <p>Saber interpretar os comandos, analisar, organizar, corrigir e solucionar.</p>
	CRIAR	Os alunos deverão criar uma situação-problema em forma de comando de questão, com certo nível de complexidade, mas viável de ser solucionado.	Estimular a solução de problemas e a capacidade de criar a partir dos conhecimentos adquiridos anteriormente.	<ul style="list-style-type: none"> • A capacidade de trabalhar em grupo. • Aprender de maneira colaborativa. • Potencializar a solução de problemas. • Estimular o processo criativo. 	<p>Passar pelas dimensões Compreender, Conceber, Completar e Compatibilizar e Corrigir e Construir;</p> <p>Ter conhecimentos para criar um comando de questão, interpretar comandos, e saber solucionar problemas.</p>

Fonte: Elaborado pelo autor.

É importante deixar claro que as dimensões seguem uma ordem lógica, subdividida em três fases inspiradas nas categorias de aprendizagem propostas por Ausubel (2000), conforme já citado. As cores indicadas no Quadro 2 representam as fases propostas pela Metodologia 7Cs, sistematizadas no Quadro 3.

Quadro 3 - Fases da Metodologia 7Cs.

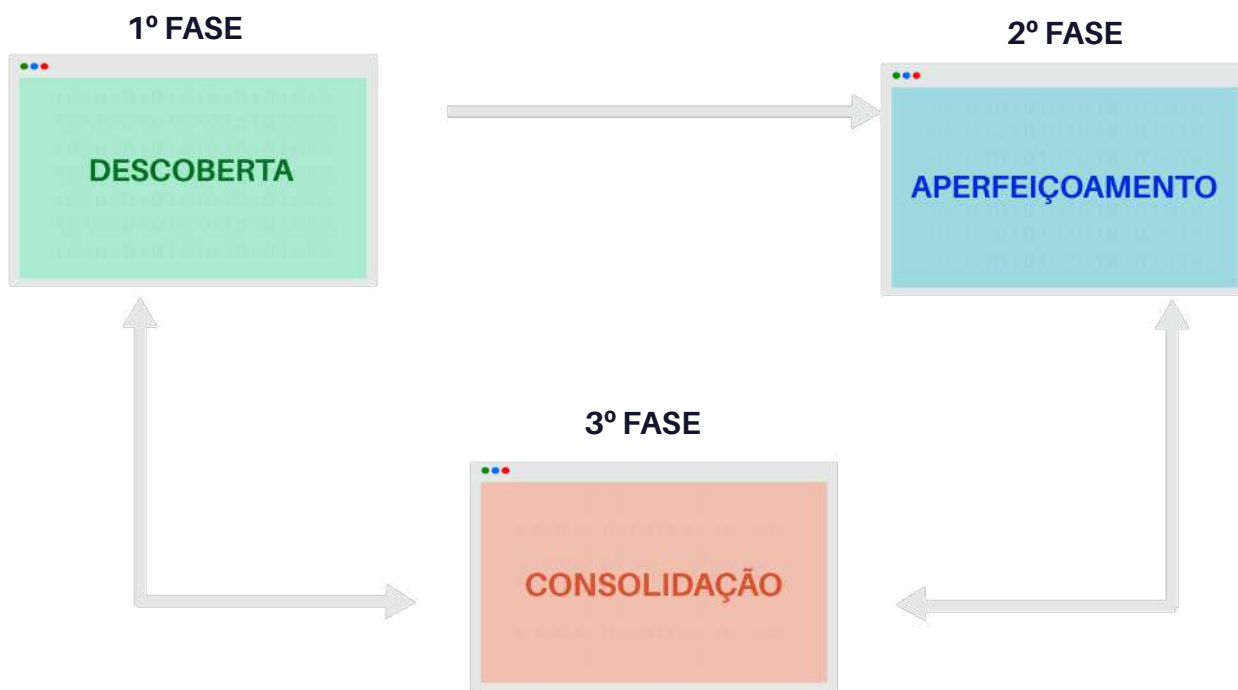
METODOLOGIA 7CS				Ausubel (2000)
FASE	DESCRIÇÃO	DIMENSÃO	O QUE SE ESPERA	CATEGORIA DE APRENDIZAGEM
DESCOBERTA	Consiste no contato inicial dos alunos quando começam a interagir com os conceitos e estruturas básicas de algoritmos.	<ul style="list-style-type: none"> • Compreender • Conceber 	Por meio da compreensão e interpretação da estrutura e construção dos seus primeiros algoritmos, pretende-se que os alunos adquiriram conhecimentos prévios, base para a próxima fase.	SUBORDINADA
APERFEIÇOAMENTO	Consiste em sensibilizar seus alunos a perceber características importantes para construir um algoritmo coerente e de fato executável.	<ul style="list-style-type: none"> • Completar • Compatibilizar • Corrigir 	É o momento em que, a partir dos subsunções adquiridos na fase anterior, os alunos irão concretizar a solução dos problemas propostos nas dimensões da fase Aperfeiçoamento, com base em conhecimentos preexistente	SUPERORDENADA
CONSOLIDAÇÃO	Esta fase consiste em consolidar e combinar os conhecimentos adquiridos visando à construção de algoritmos eficientes.	<ul style="list-style-type: none"> • Construir • Criar 	É nesta fase que os alunos irão combinar os conhecimentos aprendidos nas duas primeiras fases para solucionar problemas com maior grau de exigência e/ou complexidade, tornando o conhecimento mais sólido.	SUPERORDENADA

Fonte: Elaborado pelo autor.

Com base no Quadro 3, a ancoragem dos conhecimentos, ilustrada na Figura 1, ocorre da seguinte forma: a fase de Descoberta (primeira) serve de âncora para a fase de Aperfeiçoamento (segunda) e ambas servem de âncora para a fase de Consolidação (terceira).

Figura 1: Fases e ancoragem de conhecimentos da Metodologia 7Cs.

ANCORAGEM DE CONHECIMENTOS

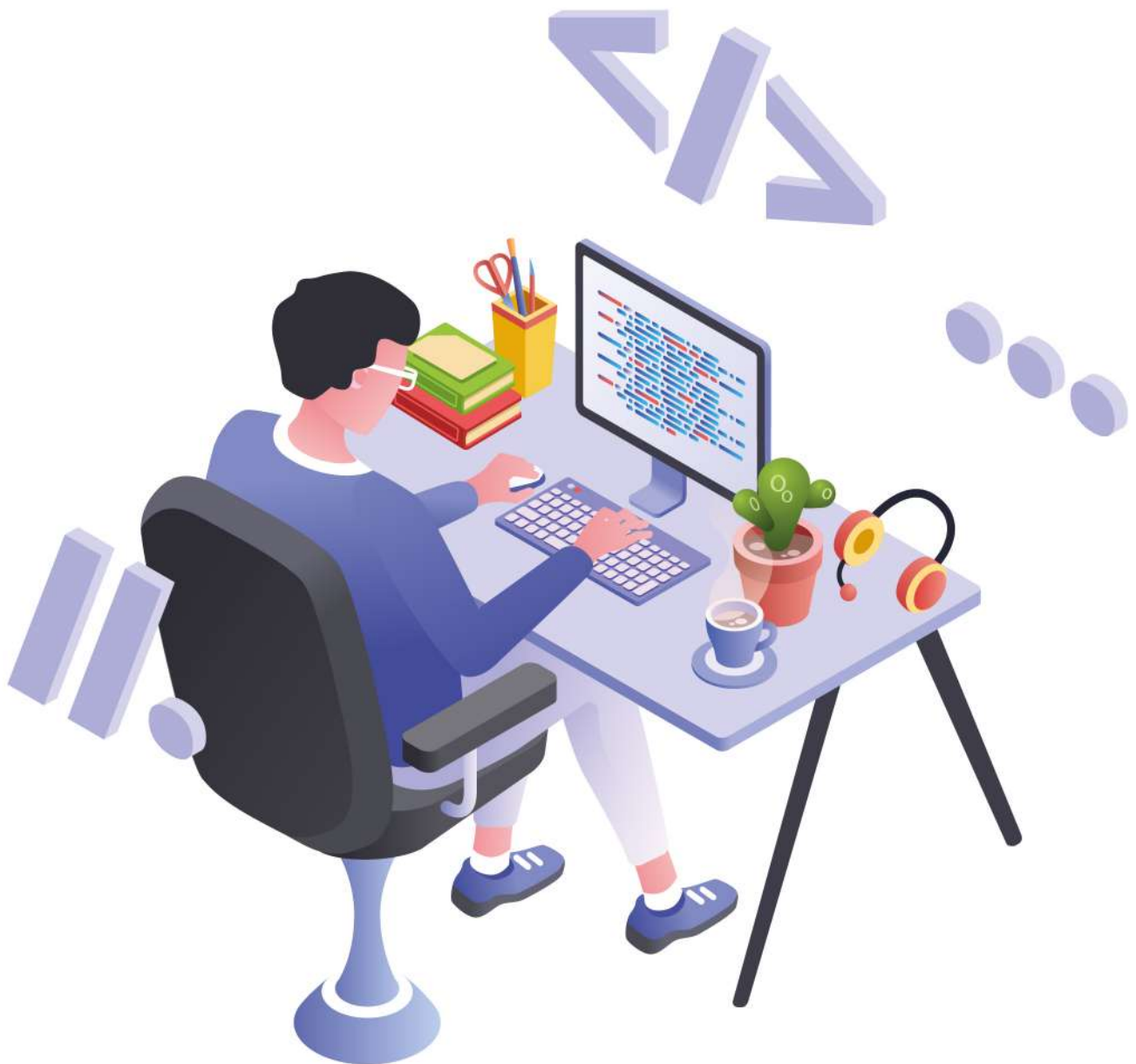


Fonte: Elaborado pelo autor.

Na teoria de Ausubel (2000), os indivíduos aprendem melhor quando o novo conhecimento é relacionado com outros preexistentes. Por isso, ao partirmos do princípio de que a maioria dos alunos dificilmente possui conhecimentos prévios sobre os conteúdos relacionados à construção de algoritmos, criamos um caminho metodológico para permitir que, a cada fase, o aluno possa adquirir conhecimentos prévios que sirvam como âncoras para a fase seguinte.

III

A METODOLOGIA 7Cs EM AÇÃO



A Metodologia 7Cs passou por um teste piloto em que realizamos uma parceria com o Programa de Nivelamento em Algoritmos (PNA), no Instituto de Ciências Exatas e Naturais (ICEN), da Universidade Federal do Pará (UFPA), no ano de 2019. O PNA consiste em um curso de nivelamento dos conteúdos básicos de algoritmos, ofertado na semana do calouro da UFPA para os cursos de graduação em Ciências da Computação e Sistemas de Informação. As fases da metodologia foram aplicadas na prática do curso, e nos renderam bons frutos.

Após o teste piloto, produzimos um artigo para o Congresso da Sociedade Brasileira de Computação (CSBC2019)², um dos maiores eventos de Computação do Brasil. Nesse artigo publicado, relatamos a experiência vivenciada no PNA, na categoria de trabalhos de pesquisa em andamento, no *Workshop* sobre Educação em Computação. E para nossa imensa felicidade, nosso artigo foi premiado, recebendo uma menção honrosa pelo trabalho produzido.

Realizamos, também, a aplicação de parte da Metodologia 7Cs na disciplina Algoritmos ofertada no primeiro semestre de 2019, no curso de graduação em Ciência da Computação/ICEN/UFPA. Nessa experiência, percebemos a possibilidade real da utilização da metodologia e sua viabilidade de implementação em sala de aula.

E vocês, professores, como podem implementar a metodologia em sala de aula?

Não queremos dar uma receita pronta, mas apresentamos uma proposta com exemplos e sugestões de aplicação para cada uma das dimensões da Metodologia 7Cs. Esclarecemos que a metodologia não foi criada para interferir na forma como os conteúdos são ministrados, mas sim na forma como os alunos podem praticar a construção de soluções de problemas.

Vocês já devem ter percebido que na metodologia existe uma sequência lógica para a construção do conhecimento de maneira gradativa. Por esse motivo, indicamos que ela seja seguida, pois uma dimensão serve de base para outra. Entretanto, sintam-se livres para criar suas próprias formas de empregar a metodologia a partir de suas vivências e realidades. O importante é que cada dimensão seja utilizada em sua essência, respeitando a fase em que se encontra.



² Link para acesso ao artigo: <https://sol.sbc.org.br/index.php/wei/article/view/6648/6544>

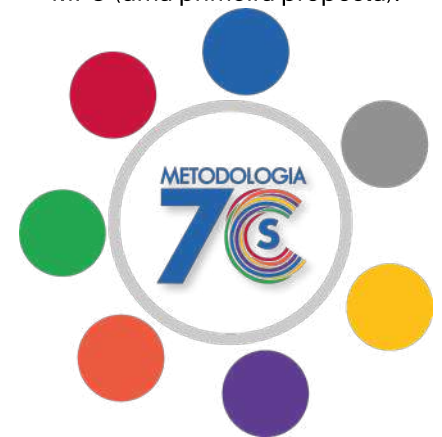


Pontapé inicial

Por ser uma proposta nova, sugerimos que vocês inicialmente apresentem-na a seus alunos a fim de que eles se familiarizem com suas particularidades e entendam o porquê de seu uso. Ao pensar em uma apresentação para mostrar o potencial da metodologia de uma maneira divertida e chamar a atenção e curiosidade dos alunos sobre ela, criamos um jogo intitulado M7C que trabalha cada uma das dimensões (o gabarito do jogo para impressão encontra-se no Apêndice I).

O objetivo da utilização desse jogo com os alunos é trabalhar, de maneira intuitiva, a Metodologia 7Cs em sala de aula e, como bônus, alguns conceitos iniciais de algoritmos (formas de representação algorítmicas, entrada e saída de dados e estrutura sequencial). O jogo foi estruturado em forma de tabuleiro (ver Figura 2) e contém sete territórios, cada um representando uma dimensão.

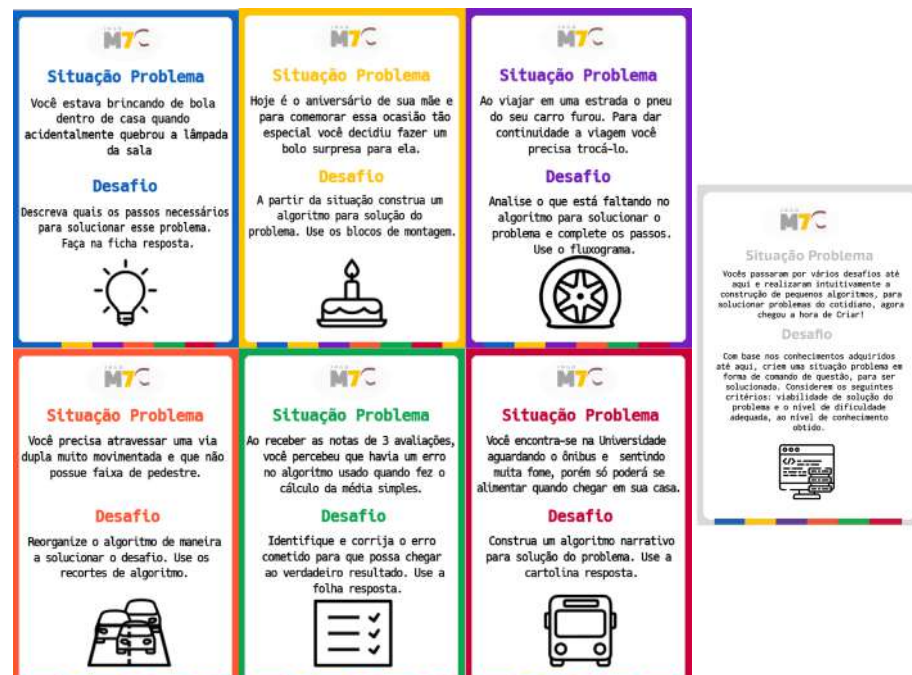
Figura 2: Tabuleiro do jogo M7C (uma primeira proposta).



Fonte: Elaborado pelo autor.

Cada território contém diferentes situações-problemas e desafios relacionados ao cotidiano. As situações cotidianas estimularão nos alunos a ancoragem de conhecimentos, ao relacionar vivências anteriores com as novas, tornando um ambiente propício à aprendizagem significativa. Na Figura 3, são apresentadas as cartas Desafios.

Figura 2 - Cartas do Jogo



Fonte: Elaborado pelo autor.

São distribuídas diferentes situações-problemas e desafios relacionados ao cotidiano para que os alunos compreendam determinados conceitos de algoritmos e para que, em cada território do jogo, possa suscitar uma dimensão da Metodologia 7Cs. Quando o jogo é iniciado, todos os grupos formados realizam os desafios propostos ao mesmo tempo. O grupo que apresentar a melhor resposta marcará o território correspondente ao desafio realizado. Vencerá aquele que, ao final do jogo, tiver marcado mais territórios.

Durante o jogo, vocês ficam responsáveis por conduzir todo o processo a cada desafio realizado. Devem analisar e verificar as respostas dadas pelos alunos, bem como validar o grupo vencedor de cada uma das dimensões. Além disso, após cada desafio, devem fornecer o *feedback* para os alunos sobre a dimensão correspondente à Metodologia 7Cs e sobre os conteúdos iniciais trabalhados. Uma explicação completa vocês encontram no Apêndice i.



Explorando as Fases

DESCOBERTA

As dimensões presentes nesta fase são: Compreender e Conceber.

Compreender (C1)

(Analogia à palavra) Compreender é buscar entender e assimilar alguma coisa. É um processo cognitivo que necessita da interpretação de algo, para que assim seja compreendido.

É o momento em que os alunos buscam **compreender** trechos de códigos já construídos por meio de análise e interpretação, antes mesmo de conceber seus primeiros algoritmos com base na estrutura que está sendo trabalhada.

A ideia é que, antes de começar a praticar qualquer conteúdo de algoritmos, os alunos possam compreender a estrutura e suas funcionalidades por meio da análise e interpretação de códigos já construídos, e assim obter melhores requisitos para iniciar a construção dos seus próprios algoritmos.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Compreender e interpretar os funcionamentos da estrutura de um algoritmo;
- Desenvolver a capacidade de solução de problemas.

Observem que:

- É importante que os algoritmos ou trechos destes sejam escolhidos cuidadosamente, pois é o primeiro contato dos alunos com a estrutura trabalhada;
- É interessante que os alunos realizem as atividades no papel, para que a aprendizagem seja mais construtiva à medida que os eles adquirem um maior nível de concentração.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

- A partir dos trechos de algoritmos a seguir, descreva, de acordo com sua compreensão, o que será executado:

a) Var

```
    Num: integer;  
    Begin  
    Writeln ('digite um número');  
    Readln (num);  
    Writeln ('o antecessor de ', num, ' eh: ', num-1);  
End.
```

b) num:= 10;

```
    For cont:= -1 to 3 do  
    Begin  
    Writeln (num);  
    Num:= num - cont;  
End;
```

Obs.: vejam que usamos neste exemplo duas estruturas diferentes (a - estrutura sequencial, b - estrutura de repetição) apenas como demonstração, mas na prática devem ser aplicadas no momento em que for iniciado o ensino de suas respectivas estruturas.

Conceber (C2)

Conceber é o primeiro passo para se concretizar algo, que ainda pode ser amadurecido e trabalhado.

É o momento em que os alunos **concebem** seus primeiros algoritmos simples e, assim, demonstram a sua compreensão sobre a estrutura de algoritmos trabalhada.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Interpretar os comandos das questões;
- Desenvolver a capacidade de solução de problemas.

Observem que:

- É interessante que os alunos possam fazer as soluções primeiramente no papel, para depois passarem para o computador. Assim, eles trabalham menos com tentativas e erros, como o computador permite fazer;
- Uma boa técnica que pode ser interessante ser estimulada nessa dimensão é o teste de mesa, o qual permite simular o comportamento do código construído no papel, para buscar identificar erros mesmo sem o computador.

Obs.: o teste de mesa é realizado de maneira manual para validar a coerência lógica de um algoritmo construído.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

- A partir dos comandos de algoritmos a seguir, conceba a solução dos seus algoritmos.

- Escreva um algoritmo para ler um valor inteiro (do teclado) e escrever (na tela) o seu antecessor.
- Faça um algoritmo que leia três notas de um aluno, calcule e escreva a média final desse aluno. Considerar que a média é ponderada e que o peso das notas é 2, 3 e 5.

Obs.: usamos neste exemplo duas perguntas básicas de estrutura sequencial, apenas como demonstração, mas na prática devem ser aplicadas no momento em que for iniciado o ensino de suas respectivas estruturas.

APERFEIÇOAMENTO

As dimensões presentes nesta fase são: Completar, Compatibilizar e Corrigir.

Completar (C3)

(Analogia à palavra) Nem tudo que é incompleto tem um bom funcionamento. É preciso que todas as peças possam se fazer presentes para que nosso objetivo seja alcançado com eficiência.

Os alunos devem ser capazes de identificar, com base em algoritmos disponibilizados, quais elementos estão faltando, para que, ao completá-los, possam alcançar a solução desejada para o problema exposto.

Passar por esta dimensão é importante, pois, para a construção de algoritmos eficientes, se faz necessário o entendimento de que todos os elementos são essenciais para atingir o objetivo desejado.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Interpretar os comandos das questões e códigos;
- Desenvolver a capacidade de percepção de erros e de ajuste de algoritmo.

Observem que:

• É importante que sejam escolhidos trechos de algoritmos bem elaborados, de forma a aumentar a probabilidade de sucesso dos alunos na atividade;

Mais uma vez, acreditamos que os alunos devam trabalhar primeiramente no papel, realizando o teste de mesa.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

— A partir do comando, analise e identifique o que falta para que o algoritmo possa cumprir com o que esta sendo solicitado:

?

Comando da questão: Construir um algoritmo que leia dois números e efetue a adição. Caso o valor somado seja maior que 20, este deverá ser apresentado somando-se a ele mais 8; caso o valor somado seja menor ou igual a 20, este deverá ser apresentado subtraindo-se 5.

Código Incompleto

```
Program_Completar;
//Declaração de variáveis
Var
  a, b, soma: integer;
Begin
  writeln('Digite um valor para B:');
  readln(b);
  if (soma > 20) then
    writeln('O valor da A + B é maior que
20,
resultado final: ', soma +8)
    writeln('O valor de A + B é menor ou
igual a 20, resultado final: ', soma - 5)
  else writeln('Valores inválidos');
  readln;
end.
```

Resposta Esperada

```
Program_Completar;
//Declaração de variáveis
Var
  a, b, soma: integer;
Begin
  writeln('Digite um valor para A:');
  readln(a);
  writeln('Digite um valor para B:');
  readln(b);
  soma:= a + b;
  if (soma > 20) then
    writeln('O valor da A + B é maior que
20, resultado final: ', soma +8)
  else if (soma <= 20) then
    writeln('O valor de A + B é menor ou
igual a 20, resultado final: ', soma - 5)
  else writeln('Valores inválidos');
  readln;
end.
```

*Obs: a coluna resposta está exemplificando uma possível resposta dos alunos.

Compatibilizar (C4)

(Analogia à palavra) A ordem e sequência de nossas ações precisam ser compatíveis, para que assim possamos alcançar nossos objetivos!

Aqui, os alunos analisarão e interpretarão o problema a ser solucionado, para em seguida reordenar as linhas de comando, de modo a encadear os passos de forma lógica, corretamente. Pensamos na construção da dimensão Compatibilizar para mostrar a importância do encadeamento correto das informações na construção de um algoritmo.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Interpretar os comandos das questões;
- Desenvolver a capacidade de organizar o encadeamento das ideias;
- Desenvolver a capacidade de ajuste de algoritmo.

Observem que:

- A dimensão Compatibilizar pode ser feita em forma de dinâmica em que os alunos recebem um algoritmo recortado para organizar de acordo com a solução do problema disponibilizado pelo professor;
 - Para os alunos terem uma melhor visualização da proposta, o professor pode projetar o comando e o algoritmo desordenado.
 - Por ser um código já inteiro, os alunos podem fazer diretamente no computador, para que possam testar e verificar os erros de ordenamento nas linhas de comando.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

— A partir do algoritmo a seguir, analise e interprete o comando da questão, realocando as linhas de comando de forma ordenada e coerente para que o algoritmo possa corresponder ao que se pede.



Comando da questão: Faça um programa que receba três notas, apresente a média e mostre, se o aluno foi aprovado, se sua média for maior que igual ou maior que 7, quanto precisa tirar em novo exame e reprovado se sua média for entre 0 e 3.

Código Incompleto

```
Program_compatibilizar;
Begin
  Var Nota1, Nota2, Nota3, Media, Nota_exame: Real;
  Begin
    Write ('digite a terceira nota: ');
    ReadLn(Nota3);
    Write
('Digite a primeira Nota: ');
    ReadLn(Nota1);
    Write('Digite a segunda nota: ');
    ReadLn(Nota2);
    Writeln('Média aritmética: ',Media:);
    If (Media >=0) And (Media <3)
    Then Writeln('reprovado');
    If (Media >=3) And (Media < 7) Then
    Media := (Nota1 + Nota2 + Nota3) / 3;
    Begin
      Writeln('exame ');
      Nota_exame := 12 - Media;
      Writeln ('deve tirar nota ',Nota_exame:4:2,' para ser aprovado');
    End;
    If (Media >= 7) And (Media < 10)
    Then Writeln('aprovado');
    ReadLn;
  End.
```

Resposta Esperada

```
Program_compatibilizar;
Begin
  Var Nota1, Nota2, Nota3, Media, Nota_exame: Real;
  Begin
    Write('Digite a primeira Nota: ');
    ReadLn(Nota1);
    Write('Digite a segunda nota: ');
    ReadLn(Nota2);
    Write ('digite a terceira nota: ');
    ReadLn(Nota3);
    Media := (Nota1 + Nota2 + Nota3) / 3;
    Writeln('Média aritmética: ',Media:);
    If (Media >=0) And (Media <3)
    Then Writeln('reprovado');
    If (Media >=3) And (Media < 7) Then
    Begin
      Writeln('exame ');
      Nota_exame := 12 - Media;
      Writeln ('deve tirar nota ',Nota_exame:4:2,' para ser aprovado');
    End;
    If (Media >= 7) And (Media < 10)
    Then Writeln('aprovado');
    ReadLn;
  End.
```

*Obs.: percebam que na dimensão Compatibilizar o algoritmo vai ser disponibilizado inteiro sem a necessidade de o aluno acrescentar algo a mais, apenas ordenar as linhas de comando

Corrigir (C5)

(Analogia à palavra) Muitas vezes cometemos erros que impedem uma boa execução das nossas tarefas e atividades e é preciso corrigi-los para que o problema seja solucionado.

Consiste na capacidade de os alunos identificarem erros em algoritmos já construídos e corrigi-los. Na estrutura de um algoritmo, é fundamental que todos os elementos em sua construção estejam corretos, para que se possa alcançar a solução do problema proposto. Por isso, é importante o estímulo à capacidade de identificar erros em um código e corrigi-los.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Corrigir códigos com erros;
- Desenvolver a capacidade de análise e ajuste de algoritmos.

Observem que:

- É interessante que primeiramente os alunos façam a correção dos algoritmos de maneira escrita;
- Após corrigir as questões no papel, podem também realizar o teste de mesa, para posteriormente executar os algoritmos na linguagem de programação utilizada pelo professor;
- O erro no código pode ser de sintaxe, com elementos trocados, como um sinal de mais ("+") no lugar do sinal de menos ("-"), entre outros.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

— A partir do código a seguir, analise e interprete o comando da questão para identificar no código disponibilizado os possíveis erros e corrigi-los de forma a alcançar o que se pretende:



Comando da questão: Desenvolver um algoritmo que leia dois números e mostrar o valor do maior número ou dizer se os valores são iguais.

Código Incompleto

```
Program_Corriger;
Var Num1, Num2: Real;
Begin
  Write ('Digite o primeiro número: ');
  Readln(Num1);
  Write('Digite o segundo número: ');
  Readln(Num2);
  If Num1 > Num2 then
  Writeln ('O maior número : ',Num2)
  Else
  If Num2 < Num1 then
  Writeln('O maior número : ', Num3)
  Else
  Writeln('Os números são iguais');
  Readln;
End.
```

Resposta Esperada

```
Program_Corriger;
Var Num1, Num2: Real;
Begin
  Write ('Digite o primeiro número: ');
  Readln(Num1);
  Write('Digite o segundo número: ');
  Readln(Num2);
  If Num1 > Num2 then
  Writeln ('O maior número : ',Num1)
  Else
  If Num2 > Num1 then
  Writeln('O maior número : ', Num2)
  Else
  Writeln('Os números são iguais');
  Readln;
End.
```

*Obs: a dimensão Corrigir possui um grau de dificuldade elevado, principalmente porque cabe aos alunos o desafio de corrigir códigos feitos por outras pessoas.

CONSOLIDAÇÃO

As dimensões presentes nesta fase são: Construir e Criar.

Construir (C6)

(Analogia à palavra) Construir é começar algo do zero, é usar as habilidades e competências adquiridas durante a vida para solucionar determinados problemas.

Nesta dimensão, consolidam-se todos os conhecimentos adquiridos anteriormente na solução de problemas mais complexos, que exigem maior esforço e conhecimento dos alunos.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Interpretar os comandos das questões;
- Desenvolver a capacidade de solução de problemas;
- Construir algoritmos com eficiência.

Observem que é interessante:

• Os alunos praticarem primeiramente no papel e posteriormente na linguagem de programação, para melhor apropriação da construção e encadeamento das informações de um algoritmo;

Vocês fornecerem um *feedback* das atividades realizadas pelos alunos.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

— A partir do comando disponibilizado e dos aprendizados alcançados até o presente momento, construa a solução do problema



Comando da questão: Escrever um algoritmo que lê um conjunto de 4 valores i , a , b , c , em que i é um valor inteiro e positivo e a , b , c , são quaisquer valores reais e os escreva. A seguir:

- Se $i=1$ escrever os três valores a , b , c em ordem crescente.
- Se $i=2$ escrever os três valores a , b , c em ordem decrescente.
- Se $i=3$ escrever os três valores a , b , c de forma que o maior entre a , b , c fique dentre os dois.

Código Incompleto

```
Program_ Construir;
var
  n1,n2,n3,primeiro,segundo,terceiro,aux,i:
integer;
Begin
  writeln('Informe os tres numeros:');
  Readln(n1,n2,n3);
  primeiro := n1;
  segundo := n2;
  terceiro := n3;
  if (primeiro > segundo) then
  begin
    aux:= primeiro;
    primeiro:= segundo;
    segundo:= aux;
  end;
  if (primeiro > terceiro) then
  Begin
  aux:= primeiro;
```

Resposta Esperada

```
primeiro:= terceiro;
terceiro:= aux;
end;
if (segundo > terceiro) then
  Begin
    aux:= segundo;
    segundo:= terceiro;
    terceiro:= aux;
  end;
  Writeln ('Digite a opção desejada:');
  Readln(i);
  Case i of
    1: writeln('Resultado: ',primeiro,' ', segun-
do, ' ', terceiro);
    2: writeln('Resultado: ',terceiro,' ', segun-
do, ' ', primeiro);
    3: writeln('Resultado: ',segundo,' ', ter-
ceiro,' ', primeiro);
  End;
end
```

*Obs.: perceba que a diferença entre as dimensões Conceber e Construir está no nível de complexidade dos exercícios.

Criar (C7)

(Analogia à palavra) Criar, é fazer com que algo seja construído a partir de nós mesmos, com os nossos conhecimentos preexistentes. É desenvolver e compor na mente.

Nesta dimensão, com base nas habilidades e competências já adquiridas, os alunos deverão criar um comando de questão com uma situação-problema, adequado e viável de ser resolvido. Objetiva-se proporcionar um ambiente criativo, interativo e de contribuição entre os alunos para integração e compartilhamento dos conhecimentos ao estimular discussões e reflexões sobre a construção de um algoritmo, tornando a aprendizagem muito mais significativa.

As habilidades e competências que pretendemos desenvolver nos alunos são:

- Capacidade de trabalhar em grupo;
- Aprendizado de maneira colaborativa;
- Potencialização da solução de problemas;
- Estimulação do processo criativo.

Observem que:

- Para a formação dos grupos, se possível, é interessante mesclar alunos que têm mais facilidade na construção de algoritmos com os que têm menos;
- Vocês, professores(as), podem auxiliar os alunos em dúvidas que não estejam relacionadas à solução do problema de forma direta;
- Os comandos das questões podem ser feitos em folhas destacadas para facilitar na hora de passar para a outra equipe após o sorteio;
- Ao validar as respostas, é interessante chamar um representante do grupo que criou o comando da questão para verificar se a solução solicitada foi alcançada;
- Se dispuser de tempo suficiente, peçam que os alunos apresentem a solução desenvolvida por eles como forma de disseminar o conhecimento;
- É importante observar se todos os alunos estão realmente interagindo e contribuindo para criar os comandos das questões, bem como a solucionar os problemas após o sorteio.

Para ficar mais compreensível, deixamos aqui um exemplo de exercício:

— Para esta dimensão, propomos que vocês façam uma dinâmica, com dois momentos, apresentados a seguir:

1º MOMENTO

- Explicar e apresentar a atividade, seu objetivo e como será desenvolvida;
- Dividir a sala em grupos;
- Lançar o desafio para que os alunos criem um comando de questão com uma situação-problema, levando em consideração os seguintes requisitos:
 - O comando da questão deve ser elaborado com a participação de todos os integrantes do grupo;
 - O problema deve viável de ser solucionado;
 - O problema deve ter um nível de complexidade de acordo com os conhecimentos adquiridos até o momento.

Acreditamos que o tempo de 30 minutos é suficiente para os grupos elaborarem os comandos das questões.

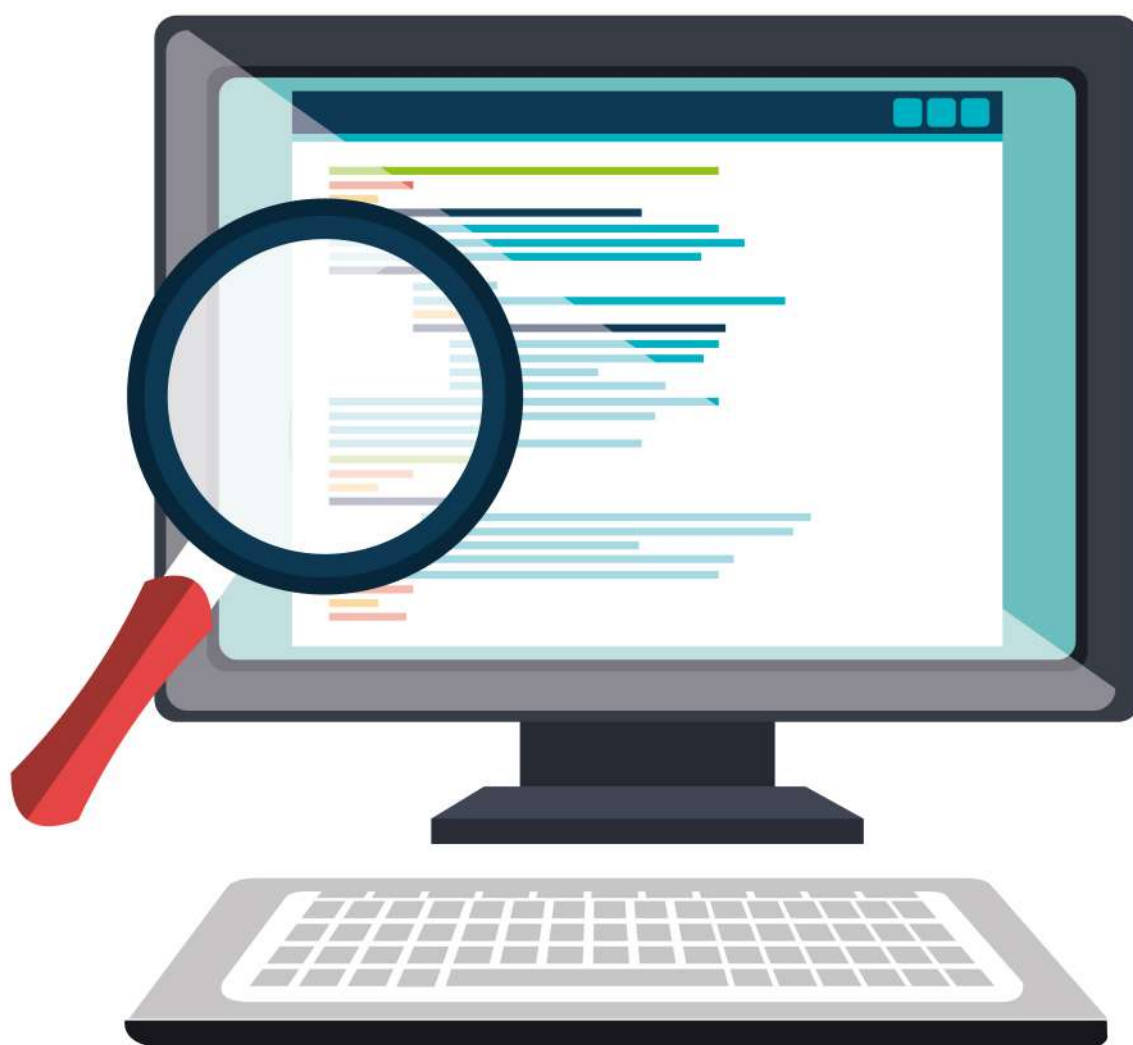
2º MOMENTO

- Após a elaboração dos comandos das questões, um novo desafio. Vocês devem numerar os grupos e realizar um sorteio respeitando a seguinte condição:
 - Se o número retirado for da própria equipe, devolva-o para o sorteio;
 - Se o número for diferente da sua equipe, então deverá solucionar o problema proposto.

Os grupos devem iniciar o desafio aos seus comandos e, ao terminarem a solução do problema sorteado, devem executar o código construído para que vocês possam validar as respostas.

IV

SUGESTÃO DE LISTAS



Não vamos conseguir determinar quantos exercícios de cada dimensão vocês precisarão desenvolver e passar aos alunos para alcançar o objetivo que pretendemos. A quantidade vai depender da necessidade, considerando o tempo que você possui disponível para a prática durante a disciplina. Indicamos que vocês possam implementar cada fase separadamente, para que os alunos possam seguir corretamente a proposta da metodologia.

Para inspirar vocês na elaboração de exercícios que possam atender às sete dimensões, apresentamos dois exemplos de listas de exercícios envolvendo estrutura de repetição e estrutura de dados homogêneas (matrizes), lembrando que a metodologia pode ser facilmente implementada para as demais estruturas. Em cada lista, deixamos os comandos na cor proposta no Quadro 3, de forma a facilitar a identificação das fases Descoberta, Aperfeiçoamento e Consolidação

4.1 Lista 1: Estrutura de Repetição

1) Analisem e interpretem os trechos de códigos a seguir, identificando, em sua resposta, o que compreendem . (Use caneta e papel)

```
a) i:= 1; a:= 2; b:= 1;
   While i < 4 do
   Begin
   i := i + 1;
   a := a * b;
   Writeln (a, ' ', i);
   End;
```

```
a1) i := 1; a := 2; b :=1;
     Repeat
     i := i + 1;
     a := a * b;
     writeln (a, ' ', i);
     Until i > 3;
```

```
b) i := 1; j := 20;
   While i < 4 do
   Begin
       j := j - i;
       i := i +1;
   End;
   Writeln (j);
```

```
b1) j := 20;
     For i := 1 to 4 do
       j := j - i;
       Writeln (j);
```

2) Conceba um algoritmo que leia 10 números inteiros, maiores que zero, e imprima os seguintes resultados:

- A soma de todos os números ímpares.
- O produto de todos os números pares.
- Quantos números múltiplos de 10 (inclusive), foram digitados.
- Quantos números múltiplos de 3 (inclusive), foram digitados.

Fazer nas três formas de repetição.

3) Escreva um algoritmo que realize a potência de A (numero inteiro) por B (Número inteiro e positivo), ou seja A elevado a B, por meio de multiplicações sucessivas. Esses dois valores são passados pelo usuário por meio do teclado. Com base no comando, complete o algoritmo a seguir:

```
Programa Completar;
Var
  a, b, i: integer;
Begin
  Writeln (' Digite dois números ');
  readln (a);
  then writeln ('não há como fazer potência')
  if b =
    then pot :=1
    else
      pot := pot* ;
if b >= 0
  then writeln (a, ' elevado a', b, '=', pot);
End.
```

4) Elabore um algoritmo que entre com números inteiros enquanto forem positivos e imprima a média dos números digitados. Percebam que as linhas de códigos do algoritmo, não estão em uma ordem correta. De acordo com o comando, compatibilize (reorganizar) o algoritmo a seguir:

```
Program Compatibilizar
  Cont: interger;
Var
  Total, media, num: real;
Begin
  Write ('digite um número');
  Cont:= 1;
  Total:= num;
  While (num>0) do
  Read (num);
  Begin
    Writeln ('digite um número:');
    If (num>0) then
      Total:= total + num;
      Begin
        Cont := Cont + 1;
        Read (num);
      End;
    End;
  End;
media := total/cont;
writeln (' A media dos valores digitados é:', media: 4:2);
End.
```

5) Faça um algoritmo que leia a idade de um número indeterminado de pessoas. Quando o usuário digitar uma idade ≤ 0 , o programa deve parar com a entrada de dados. Em seguida o algoritmo deverá apresentar:

- A idade do mais velho
- A idade do mais novo
- A quantidade de adolescentes (de 12 a 17 anos)

Partindo do comando, encontre os possíveis erros, no algoritmo a seguir, e os corrija:

```
Program Corrigir;
Var
  Idade, menor, maior, cont: integer;
Begin
  Repeat
  Writeln ('Digite a idade'):
  Read ('idade');
  if (idade <> 0) then
  begin
  if (idade < maior) then
    maior := idade;
  if (idade > maior) then
    maior = idade
  if (idade >= 12) or (idade <= 17) then
    cont := cont + 1;
  end;
until (idade <= 0);
writeln ('A idade do mais velho é de', menor, 'anos');
Writeln ('A idade do mais novo é de', maior, 'anos');
Writeln (' Numero de adolescentes (13 a 17 anos):', idade);
End.
```

6) Construa um algoritmo que leia o número de gols marcados pelo Brasil e o número de gols marcados pela Argentina, escrevendo o nome do time vitorioso ou a palavra EMPATE. A seguir, escreva a mensagem "Novo Jogo? 1. Sim 2. Não" e solicitar uma resposta. Se a resposta for 1 o algoritmo deve ser executado novamente solicitando o número de gols marcados pelos times em uma nova partida, caso contrário deve ser encerrado escrevendo:

- Quantos Jogos fizeram parte da estatística O número de vitórias do Brasil
- O número de vitórias da Argentina
- O número de empates

7) Formem grupos e criem um comando de questão com a estrutura de repetição, levando em consideração os seguintes critérios:

- Ter um nível de complexidade aceitável;
- Criar uma situação problema viável de ser solucionada;
- Deixar o comando bem claro e coerente.

4.2 Lista 2: matrizes

1) A partir de sua compreensão, escreva o que os programas a e b fazem.

```
a) Program Comprender;
Var
matriz: array[1..2, 1..2] of integer;
i, j: integer;
Begin
  for i:=1 to 2 do
    For j:=1 to 2 do
      begin
        writeln('Digite um numero para a posição ',i,', ',j);
        read(matriz[i][j]);
      End;
    clrscr;
  for i:=1 to 2 do
    begin
      for j:=1 to 2 do
        begin
          if i+j = 3 then write(' ',matriz[i][j], ' ')
            else write(' - ');
        end;
        writeln;
      end;
    End.
```

```
b) Program Comprender 1;
Var
  matriz: array[1..7, 1..8] of integer;
  l, c: integer;
Begin
  for l:=1 to 7 do
    begin
      for c:=1 to 8 do
        begin
          matriz[l][c] := l + c;
          write(matriz[l][c]:4);
        end;
        writeln;
      end;
    End.
```

2) Com base nos conhecimentos adquiridos até aqui, concebam a solução dos problemas a seguir.

a) Desenvolva um programa em Pascal que implemente uma matriz 3x3 de números reais, leia os valores e depois apresente para o usuário apenas os elementos da diagonal principal.

b) Desenvolva um programa em Pascal para armazenar a agenda do usuário. Portanto, o programa deve receber os seguintes dados pessoais referente aos contatos do usuário: nome, endereço, CEP, Bairro e Telefone. O tipo de dado que essa matriz deve receber é do tipo String. Considere que o usuário poderá ter apenas 7 contatos na agenda.

c) Desenvolva um programa em que leia uma matriz de ordem 4. Após coloque em um vetor apenas os elementos da diagonal principal. Escreva a matriz e o vetor.

3) Faça um programa que gere a seguinte matriz:

```
1 1 1 1 1 1
1 2 2 2 2 1
1 2 3 3 2 1
1 2 3 3 2 1
1 2 2 2 2 1
1 1 1 1 1 1
```

Com base no comando, complete o algoritmo a seguir:

```
Program Completar;
Var
mat: array [1.. ] of integer;
c: integer;
Begin
  For l := 1 to 6 do
    Begin
      if (l = 1) or (l = ) then mat [l,c] := 1;
      if ((l >= 2) and (l <= 5)) and ((c >=2) (c <= 5));
      if ((l >=) and (l <= 4)) and ((c >= 3) and ( c <=4)) then mat
[1] :=3;
    end;
  for l := 1 to 6
  Begin
    Writeln;
    For c:= 1 to 6 do
      Write (mat [l,c], '');
    end;
  End.
```

4) Desenvolva um algoritmo que leia do usuário um conjunto de números de uma matriz quadrada de ordem 3 e apresente ao usuário os números armazenados. Regra de constituição de matriz quadrada: número de colunas = número de linhas.

De acordo com o comando, compatibilize (reorganizar) o algoritmo a seguir:

```
Program Compatibilizar;
  matQ : array [1..3,1..3] of integer;
Begin
  Var
  for i:=1 to 3 do
```

```

Begin
  i, j: integer;
  Writeln('Apresentação da Matriz 3x3');
  Begin
    Write('Informe o valor da linha ',i,' coluna ', j,' :');
    Write([' ',matQ[i,j],']');
    read(matQ[i,j]);
    end;
End.
  for i:=1 to 3 do
    for j:=1 to 3 do
      Begin
        end;
  for j:=1 to 3 do
    end;

```

5) Desenvolva um algoritmo em pascal que monte uma nota fiscal, para isso utilize a estrutura de matriz, na primeira coluna deve armazenar a quantidade de produto solicitado, na segunda linha o valor unitário e na terceira linha o valor total por produto. Considere que a nota fiscal pode ter 10 linhas e o usuário informará na primeira interação os 10 produtos.

Partindo do comando, encontre os possíveis erros, no algoritmo a seguir, e os corrija:

```

Program Corrigir ;
Var
nota: array [1..15, 1..2] of inteiro;
i, j: real;
Begin
  Writeln('Nota fiscal');
  for i:=1 to 13 do
    for j:=1 to 3 do
      Case j of
        1: Begin
          Writeln('=====');
          Write('Informe a quantidade : ');
          readln(nota[i,i]);
          end;
        2: Begin
          Write('Informe o valor unitário : ');
          readln(nota[i,j]);
          end;
        4: Begin
          nota[i,j] = 1; nota[i,j] = nota[i,j+1]+nota[i,j +2];
          end;
      end;
  clrscr;
  writeln('||||||||||||||||||||||||||||||||||||||||||||||||||||||||');

  writeln('Nota fiscal final');

```

```

writeln ( ' Quantidade || Valor Unitário || Valor total ');
for i:=1 to 15 do
  Begin
    for j:=1 to 2 d
      Begin
        Case i of
          1: Write( ' ,nota[i,j]:2:0,' ||');
          3: Write( ' ,nota[i,j]:2:2,' ||');
          3: Write( ' ,nota[i,j]:2:2);
        end;
      end;
    end;
  End.

```

6) Leiam atentamente os comandos das questões e construam a solução dos problemas.

a) Criar um programa em Pascal que entre com valores para duas matrizes inteiras de ordem 5, gere e imprima a matriz diferença.

b) O Triângulo de Pascal é formado por uma composição especial de números inteiros. Cada um dos números do triangulo é formado pela soma de dois números da linha anterior: um imediatamente acima e o outro, acima e a esquerda. Fazer um programa em Pascal para ler um numera inteiro positivo que representa a altura de um Triângulo de Pascal, gerar e imprimir os valores desse Triângulo no monitor de vídeo do computador (a altura máxima da matriz deve ser 50). Veja um exemplo de execução do programa, a seguir:

Altura = 7

```

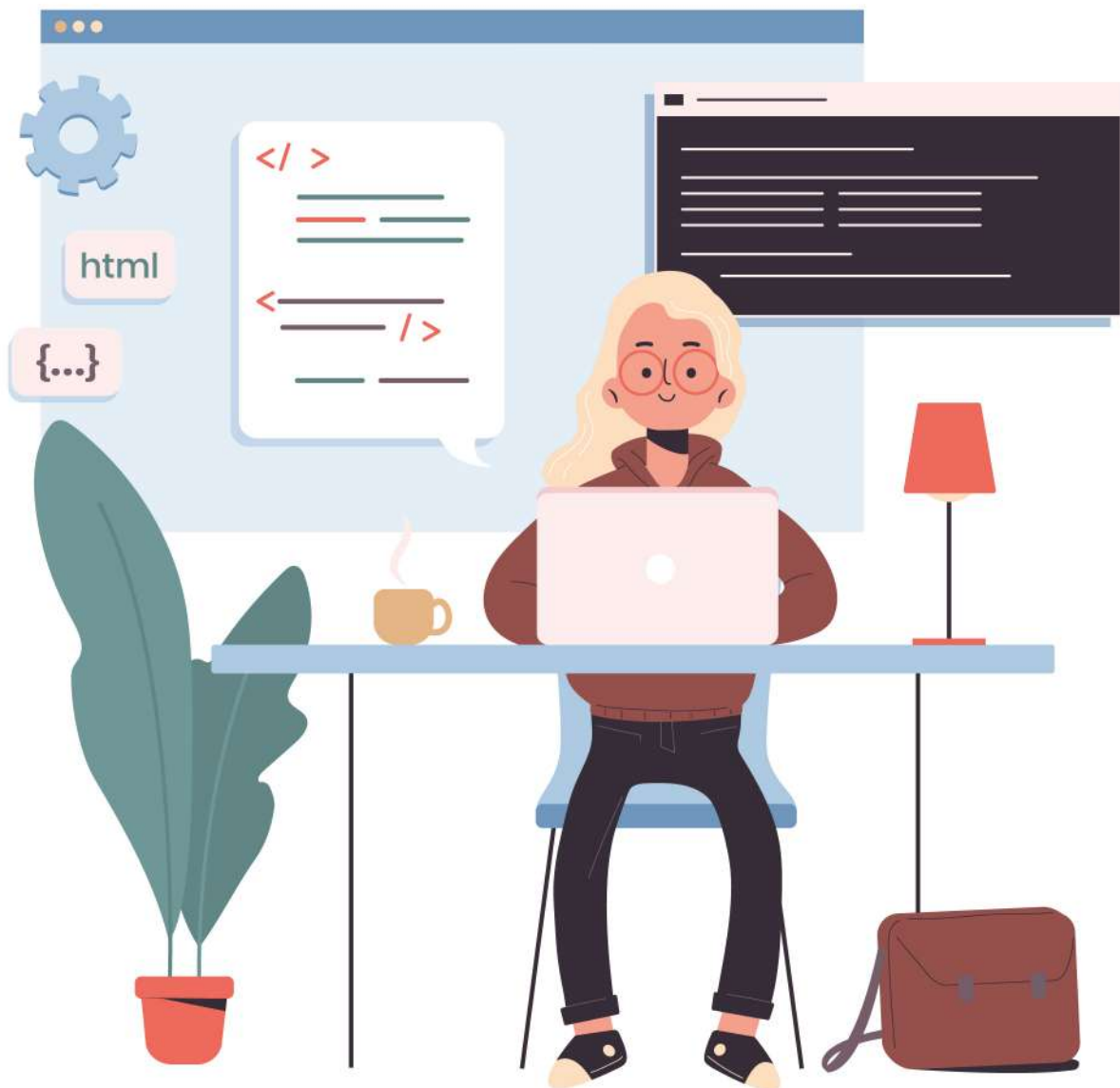
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

```

7) Formem grupos e criem um comando de questão com uma situação-problema para ser resolvida com Matrizes, levando em consideração os seguintes critérios:

- Ter um nível de complexidade aceitável;
- Criar uma situação problema viável de ser solucionada;
- Deixar o comando bem claro e coerente.

MÃOS À OBRA!



Sabemos que muitos trabalhos voltados à disciplina Algoritmos já foram desenvolvidos. Todos buscam minimizar as dificuldades com soluções julgadas viáveis de serem implementadas em sala de aula. A Metodologia 7Cs é uma entre essas propostas que pretendem contribuir com vocês, professores(as), e com seus alunos em seu processo de aprendizagem significativa, visando diminuir o cenário corriqueiro de altos índices de reprovação e evasão na disciplina Algoritmos.

Chegou a hora por nós desejada, a de vocês explorarem todos os conhecimentos e discussões realizadas até aqui para colocarem em prática o uso das sete dimensões. Venham conosco e experimentem a Metodologia 7Cs.

Criamos um grupo no *Facebook*³ para que vocês possam acessar novidades sobre a Metodologia 7Cs, relatar e compartilhar conosco suas experiências ao aplicá-la em sala de aula, apresentando os resultados alcançados. Sua participação será de grande valia para aprimorar nossa proposta e ampliar seu raio de ação.

Boa experimentação!

Aguardamos vocês no *Facebook*

³ Link de acesso ao grupo do facebook: <https://www.facebook.com/groups/558756151522251/?ref=share>

REFERÊNCIAS BIBLIOGRÁFICAS

AUSUBEL, David P. **Aquisição e retenção de conhecimentos**. Lisboa: Plátano Edições Técnicas. Tradução do original *The acquisition and retention of knowledge*, 2000.

BARCELOS et al. O uso de mobile learning no ensino de algoritmos. **RENOTE** - Revista Novas Tecnologias na Educação, [v. 7](#), n. 3, 2009.

GIRAFFA, L. M. M.; MORA, M. C. Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In: III CONFERENCIA SOBRE EL ABANDONO EN LA EDUCACIÓN SUPERIOR (III CLABES), v. 3, 2013, Espanha. **Anais...** 2013.

GOMES, A. and MENDES, A. J. (2007). Learning to program-difficulties and solutions. In International Conference on Engineering Education-ICEE. In: INTERNATIONAL CONFERENCE ON ENGINEERING EDUCATION - ICEE, 2007. Coimbra, Portugal. **Anais...** 2007.

PIVA Jr., D. e FREITAS, R. L. Estratégias para melhorar os processos de Abstração na disciplina de Algoritmos. In: XXI SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 2010, João Pessoa, PB. **Anais...** 2010.

RAABE, A. L. A.; SILVA, J. M. C. da. Um ambiente para atendimento às dificuldades de aprendizagem de algoritmos. In: XIII WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 2005. São Leopoldo, RS. **Anais...** 2005.

VIEIRA, E. C.; LIMA JUNIOR, A. T. de; VIEIRA, P. de P. V. **Dificuldades no Processo de Aprendizagem de Algoritmos**: uma Análise dos Resultados na Disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ. Campus Paracambi, 2015.

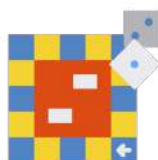
APÊNDICEI- GABARITO





Apresentação do Jogo M7C

O jogo M7C foi criado com intuito de apresentar a Metodologia 7Cs de ensino, concebida por um mestrando do Programa de Pós-Graduação Criatividade e Inovação em Metodologias de Ensino Superior (PPGCIMES). A metodologia é estruturada em três fases e composta por sete dimensões, são elas: Compreender, Conceber, Completar, Compatibilizar, Corrigir, Construir e Criar. Cada desafio do jogo representa das dimensões. Como bônus o jogo trabalha conceitos iniciais de algoritmos como sobre o que é, um algoritmo, algoritmos narrativos, estrutura sequencial e de seleção.



Estrutura do Jogo

O jogo é estruturado em forma de tabuleiro, com sete territórios. Cada território possui um desafio a ser conquistado.



No jogo contém sete cartas com diferentes situações problemas e desafios, relacionados ao cotidiano. Além disso, existem as cartas “Ajuda” para auxiliar nos cinco primeiros desafios, caso seja solicitado.



Passos Iniciais

1. Monte o tabuleiro de maneira que fique visível a todos.
2. Divida a sala em grupos, da maneira que achar melhor.
3. Identifique cada grupo com uma cor específica.
4. Disponibilize algo para marcar os territórios no tabuleiro, representando a cor do grupo vencedor.
5. Entregue as cartas com as situações problemas e desafios, de maneira que só possam ver ao seu comando, ou, projete na parede para que todos os grupos vejam e iniciar cada desafio ao mesmo tempo.
6. Entregue um “Kit de Recursos” com as folhas respostas do jogo.
7. Explique as regras do jogo M7C.



Regras do jogo

1. Você professor(a) terá a voz de comando, responsável por autorizar o início de cada desafio.
2. Os grupos devem ter um tempo mínimo de solução dos desafios, a critério do professor.

3. Para cada etapa do jogo, os grupos devem solicitar a folha de resposta correspondente ao desafio.

4. Nos cinco primeiros desafios, os grupos podem solicitar a Carta Ajuda.

5. Cada grupo terá o direito a apenas uma Carta Ajuda durante o jogo.

6. Nos dois primeiros desafios (dimensões Compreender e Conceber) e no penúltimo (dimensão Construir), todos os grupos devem ser avaliados, para verificar os algoritmos mais completos.

7. Nos desafios das dimensões Completar, Compatibilizar e Corrigir, o grupo que primeiro manifestar resolver o desafio, pode ser avaliado. Caso a resposta esteja correta, passa para próximo desafio.

8. A cada desafio, você professor(a), é responsável por analisar as respostas e determinar o grupo vencedor.

9. O grupo que vencer em determinado desafio, precisa marcar no tabuleiro do jogo o território correspondente.

10. Após os desafios é preciso realizar um feedback sobre o que cada um representou, em relação a Metodologia 7Cs e a própria disciplina.



Etapas do Jogo

Compreender: é a habilidade de interpretar, assimilar e compreender a situação problema vivenciada e as ações que devem ser realizadas sobre ele.

Conceber: é o ato de conceber as primeiras soluções dos problemas, após a compreensão, refletindo na aplicação do conhecimento obtido inicialmente.

Completar: é a capacidade de identificar os passos, que precisam ser inseridos para completar e solucionar o problema.

Compatibilizar: é a organização das ações, em uma ordem lógica, para solucionar o problema com sucesso.

Corrigir: é a capacidade de analisar e identificar possíveis erros e corrigi-los, para executar a solução do problema de forma coerente.

Construir: é a união de todos os conhecimentos obtidos durante o percurso, para solução de um problema, com nível de dificuldade mais elevado.

Criar: é o estímulo ao processo criativo, para criar situações problemas que outras pessoas precisam resolver.



Curiosidade

Na Metodologia 7Cs a ação Colaborar é transversal em suas dimensões, por isso a importância da formação de grupos no jogo, para que possa estimular o trabalho colaborativo e o compartilhamento de informações que potencializam a construção do conhecimento e a aprendizagem significativa (MOREIRA,).



Curiosidade

O jogo M7C passou por um teste piloto durante a realização de um curso do Programa de Nivelamento em Algoritmos (PNA), no Instituto de Ciências Exatas e Naturais (ICEN), da Universidade Federal do Pará (UFPA), no ano de 2019. O PNA consiste em um curso dos conteúdos básicos de algoritmos, ofertado na semana do calouro da UFPA para os cursos de graduação em Ciências da



Curiosidade

O que motivou a escolha das cores? As cores foram escolhidas considerando o significado de cada dimensão presente na Metodologia 7Cs.

Azul: estimula a confiança e a compreensão.

Amarelo: Provoca otimismo e ação.

Roxo: leva a imaginação e inspiração.

Laranja: estimula e gera a criatividade.

Verde: simboliza o crescimento e harmonia.

Vermelho: promove o entusiasmo, a conquista e vitória.

Cinza: é a cor do equilíbrio, ela está associado à tecnologia.

Recursos

Para cada desafio do jogo deve ter um recurso, que intitulamos “Folhas Resposta”, para solução do problema.

Desafio C1 - Ficha Resposta

Desafio C2 - Blocos de Montagem

Desafio C3 - Fluxograma

Desafio C4 - Recorte de um código de Algoritmo

Desafio C5 - Papel A4

Desafio C6 - Papel A3

Desafio C7 - Papel A4



Chegou a hora de aprender e brincar!

Explore o jogo M7C e de maneira divertida apresente e use a Metodologia 7Cs. Compartilhem suas experiências com o uso do jogo e os resultados obtidos, no grupo intitulado "Metodologia 7Cs", criado no Facebook.

Acesse ao link: <https://www.facebook.com/groups/558756151522251/>

Boa experimentação!

Aguardamos vocês no Facebook

TABULEIRO DO JOGO



CARTAS DO JOGO



Situação Problema

Você estava brincando de bola dentro de casa quando acidentalmente quebrou a lâmpada da sala

Desafio

Descreva quais os passos necessários para solucionar esse problema. Faça na ficha resposta.



Situação Problema

Hoje é o aniversário de sua mãe e para comemorar essa ocasião tão especial você decidiu fazer um bolo surpresa para ela.

Desafio

A partir da situação construa um algoritmo para solução do problema. Use os blocos de montagem.



Situação Problema

Ao viajar em uma estrada o pneu do seu carro furou. Para dar continuidade a viagem você precisa trocá-lo.

Desafio

Analise o que está faltando no algoritmo para solucionar o problema e complete os passos. Use o fluxograma.



Situação Problema

Você precisa atravessar uma via dupla muito movimentada e que não possui faixa de pedestre.

Desafio

Reorganize o algoritmo de maneira a solucionar o desafio. Use os recortes de algoritmo.



Situação Problema

Ao receber as notas de 3 avaliações, você percebeu que havia um erro no algoritmo usado quando fez o cálculo da média simples.

Desafio

Identifique e corrija o erro cometido para que possa chegar ao verdadeiro resultado. Use a folha resposta.



Situação Problema

Você encontra-se na Universidade aguardando o ônibus e sentindo muita fome, porém só poderá se alimentar quando chegar em sua casa.

Desafio

Construa um algoritmo narrativo para solução do problema. Use a cartolina resposta.





Situação Problema

Vocês passaram por vários desafios até aqui e realizaram intuitivamente a construção de pequenos algoritmos, para solucionar problemas do cotidiano, agora chegou a hora de Criar!

Desafio

Com base nos conhecimentos adquiridos até aqui, criem uma situação problema em forma de comando de questão, para ser solucionada. Considerem os seguintes critérios: viabilidade de solução do problema e o nível de dificuldade adequada, ao nível de conhecimento obtido.

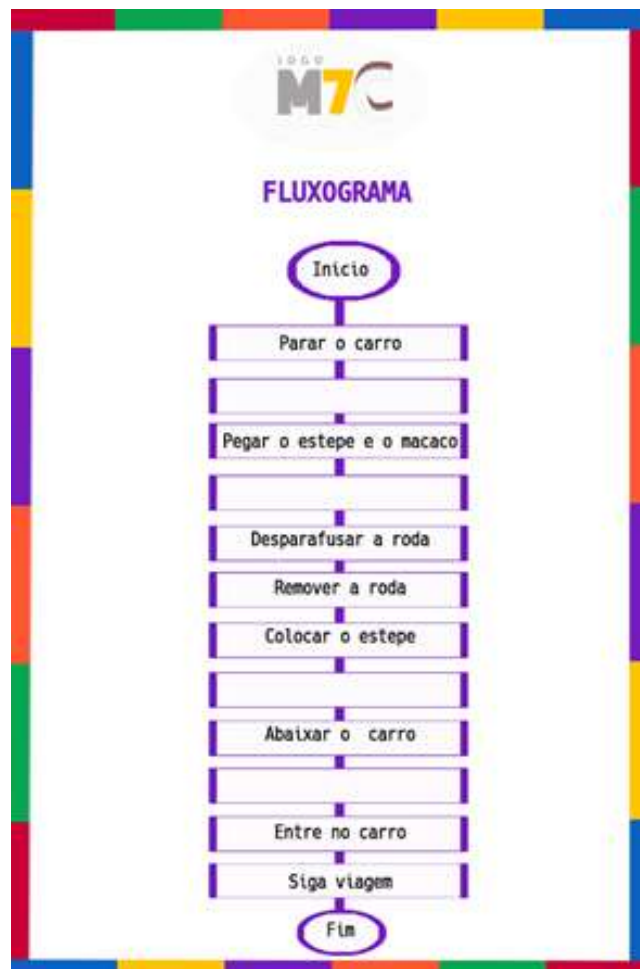


RECURSOS PARA TAREFA



FICHA RESPOSTA

1. _____
2. _____
3. _____
4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____



Início

Digite suas notas

Primeira nota

Segunda nota

Terceira nota

Média das notas

Divisão /Primeira nota+(Segunda
nota+Terceira nota)

Se sua nota for maior que 5

Então

Escrever Você foi aprovado

Se não

Escrever você foi reprovado

O resultado da sua Média foi:

Fim



CARTA AJUDA



Fazer café usando coador

Início

- 1 - Pegar uma vasilha de alumínio com água
- 2 - Levar ao fogo ate ferver
- 3 - Acrescentar o açúcar na água
- 4 - Pegar o bule e o coador
- 5 - Colocar o pó na medida desejada no coador
- 6 - Despejar a água fervente
- 7 - Após coar todo o café, retirar o coador
- 8 - Pegar a garrafa de café
- 9 - Despejar o café na garrafa

Fim





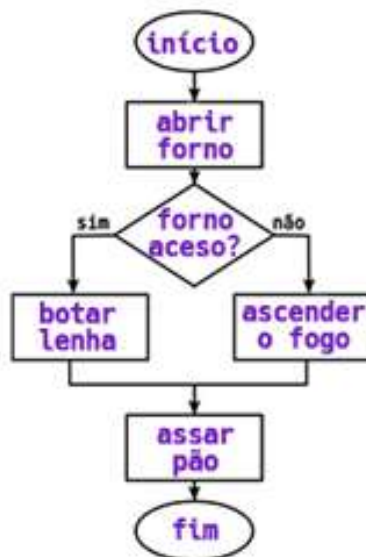
BOLO DE MILHO

INGREDIENTES:

- 3 OVOS 
- 1  DE LEITE
- 1  DE ÓLEO
- 1/2  DE AÇÚCAR
- 1  FLOCOS DE MILHO
- 1 LATA DE MILHO 
- 1  DE FERMENTO



Colocar o pão no forno





Xuxa - Atravessar a rua

Xuxa, quando a gente ta caminhando e quer atravessar a rua, o que devemos fazer?

(Xuxa)

Tem que parar, olhar o sinal, olhar para um lado e para o outro, tem que esperar, fechar o sinal, para atravessar.

QR



Tabela de Prioridade

1ª	()
2ª	/, *
3ª	+, -

Ex.: $(2+4)-2*4+(4/2)$

$$6-2*4+2$$

$$6-8+2$$

$$-2+2$$

$$= 0$$

METODOLOGIA

70S

The logo features a large, solid blue '7' on the left. To its right is a '0' composed of multiple concentric, overlapping curved lines in a rainbow spectrum (red, orange, yellow, green, blue, purple). A solid blue 'S' is positioned inside the '0'. The entire graphic is set against a background of abstract, colorful curved shapes in shades of pink, green, purple, and yellow.