

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC  
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENSINO DE CIÊNCIAS, MATEMÁTICA E TECNOLOGIAS**

**PRODUTO EDUCACIONAL**

**GUIA PARA A CRIAÇÃO DE OBJETOS DE  
APRENDIZAGEM PERSONALIZADOS NO  
AMBIENTE ADAPTWEB**

**MARCELLO LUCHT**

**JOINVILLE, SC  
2019**

**Instituição de Ensino:** UNIVERSIDADE DO ESTADO DE SANTA CATARINA  
**Programa:** ENSINO DE CIÊNCIAS, MATEMÁTICA E TECNOLOGIAS  
**Nível:** MESTRADO PROFISSIONAL  
**Área de Concentração:** Ensino de Ciências, Matemática e Tecnologias  
**Linha de Pesquisa:** Tecnologias Educacionais

**Título:** Guia para a criação de Objetos de Aprendizagem personalizados no ambiente AdaptWeb.

**Autor:** Marcello Lucht

**Orientadora:** Dra. Isabela Gasparini

**Data:** 28/06/2019

**Produto Educacional:** Objetos de aprendizagem personalizados no AdaptWeb.

**Nível de Ensino:** Ensino Superior

**Área de Conhecimento:** Computação, Tecnologias, Engenharias

**Tema:** Algoritmos personalizados para as Engenharias

**Descrição do Produto Educacional:**

Este guia consiste na orientação passo a passo para a criação de um minicurso *online* ilustrando com detalhes a concepção da disciplina e atividades no ambiente AdaptWeb. Os objetos de aprendizagem utilizados correspondem à exemplos e exercícios personalizados especialmente para a disciplina de algoritmos considerando os conteúdos e conceitos relacionados aos cursos de Engenharias.

**Biblioteca Universitária UDESC:** <https://www.udesc.br/bibliotecauniversitaria>

**Publicação Associada:** OBJETOS DE APRENDIZAGEM PERSONALIZADOS PARA A DISCIPLINA DE ALGORITMOS E LÓGICA DE PROGRAMAÇÃO CONSIDERANDO OS CURSOS DE ENGENHARIA.

**URL:** <https://www.udesc.br/cct>

Arquivo	Descrição	Formato
4,42 Mb	Texto Completo	Adobe PDF

**Licença de uso:** o Autor é titular dos direitos autorais dos documentos disponíveis e é vedado, nos termos da lei, a comercialização de qualquer espécie sem sua autorização prévia (Lei nº 12.853, de 2013).

## LISTA DE FIGURAS

Figura 1 - Ambiente do Aluno.....	12
Figura 2 - Ambiente de Autoria (Professor).....	13
Figura 3 – Estrutura do minicurso com os novos exemplos e exercícios. ....	15
Figura 4 – Cadastro de novos Cursos.....	16
Figura 5 – Cadastro de Disciplina. ....	17
Figura 6 – Associação da Disciplina com Curso(s). .....	18
Figura 7 – Inicio da estruturação da Disciplina.....	19
Figura 8 – Cadastrando o primeiro conceito do minicurso. ....	20
Figura 9 – Editando um conceito.....	21
Figura 10 – Confirmando dados e associações com os diversos cursos. ....	21
Figura 11 – Cadastrando o primeiro conceito do minicurso. ....	22
Figura 12 – Verificação de Pendências.....	23
Figura 13 – Cadastrando subtópicos.....	23
Figura 14 – Cadastro de conceitos e seções do minicurso. ....	24
Figura 15 – Verificação de Pendências.....	25
Figura 16 – Cadastro e associação de um exemplo a um curso no minicurso. ....	26
Figura 17 – Exercício para Engenharia Elétrica em formato .TXT. ....	27
Figura 18 – Criação de um exercício no Banco de Questões. ....	28
Figura 19 – Visualização de um exercício pelo aluno. ....	29
Figura 20 – Exemplos de EEL e ECV na visão do aluno. ....	31
Figura 21 – Exercícios de EPR e EMC na visão do aluno. ....	32
Figura 22 – Acesso às análises e relatórios.....	33
Figura 23 – Tela da Análise Interativa com todos os seus elementos.....	34
Figura 24 – Três diferentes representações gráficas. ....	35
Figura 25 – Filtro por Curso.....	36
Figura 26 – Gráfico de barras customizado: EEL.....	37
Figura 27 – Gráfico de linhas customizado: EMC. ....	38
Figura 28 – Gráfico de setores customizado: ECV.....	39



## APRESENTAÇÃO

Caro(a) colega Professor(a)

O produto educacional intitulado **Guia para a criação de Objetos de Aprendizagem personalizados no ambiente AdaptWeb** é resultante da pesquisa do Mestrado Profissional em Ensino de Ciências, Matemática e Tecnologias da Universidade do Estado de Santa Catarina. (Udesc), orientado pela Profª Dra. Isabela Gasparini.

O objetivo principal consiste em criar Objetos de Aprendizagem (OA) personalizados dentro do Ambiente de Ensino-Aprendizagem Adaptativo na Web – AdaptWeb. Estes OA's compreendem em um conjunto de 55 novos algoritmos distribuídos da seguinte forma: 16 exercícios e 28 exemplos com enunciados pertinentes para as Engenharias mais 4 exercícios e 7 exemplos generalistas para os cursos da área da Computação e Licenciaturas. Para ratificar estes novos OA's, um minicurso *online* foi aplicado no primeiro semestre de 2019 com alunos do CCT-Udesc e membros da comunidade. O foco principal da personalização deve-se a pesquisas realizadas com alunos no âmbito da dissertação onde foram apontadas diversas dificuldades na aprendizagem de disciplinas relacionadas a programação como Algoritmos e Lógica de Programação (ALP). Algumas dessas causas relacionam a baixa capacidade de abstração, falta de raciocínio lógico e matemático, dificuldades com interpretação de textos e enunciados como fatores mais comuns que levam alunos à desistência e evasão da disciplina. Contudo, esses alunos relataram também não perceber a importância desta disciplina para o seu curso, qual a relação que ela possui com as demais disciplinas e ainda, qual a utilidade em se aprender lógica de programação em cursos que não são da área da computação.

Tais questionamentos também foram verificados em uma na pesquisa realizada com 151 professores de todo o Brasil que lecionam/lecionaram ALP em cursos de Computação, Engenharias e outros. A partir desse contexto é apresentada a criação dos OA's no ambiente AdaptWeb.

Sob este contexto, serão apresentados conceitos básicos sobre o AdaptWeb e OA para que o leitor siga passo a passo toda a estruturação de um minicurso de ALP até efetivamente a criação de exemplos e exercícios. Por fim, acompanhar através de relatórios, gráficos e análises que verifiquem o desempenho do aluno.



## **SUMÁRIO**

1 INTRODUÇÃO .....	9
2 ADAPTWEB .....	10
3 CRIAÇÃO DE OBJETOS DE APRENDIZAGEM PERSONALIZADOS NO AMBIENTE ADAPTWEB .....	14
3.1 ORGANIZAÇÃO E ESTRUTURAÇÃO DO CONTEÚDO .....	14
3.2 CADASTRO DE CURSO .....	16
3.3 CADASTRO DE DISCIPLINA .....	17
3.4 CADASTRO DOS CONCEITOS PRINCIPAIS E SECUNDÁRIOS (SEÇÕES) ...	19
3.5 CADASTRO DE EXEMPLOS E EXERCÍCIOS .....	25
4 ANÁLISES E RESULTADOS .....	31
5 CONSIDERAÇÕES FINAIS .....	40
REFERÊNCIAS .....	41
APÊNDICE A: OBJETOS DE APRENDIZAGEM PERSONALIZADOS .....	42



## 1 INTRODUÇÃO

O ensino de programação nos cursos de graduação é tema de debate na área acadêmica, tanto pela sua importância na formação geral dos estudantes quanto pelos desafios ligados ao seu ensino, as reprovações em disciplinas de Algoritmos e Lógica de Programação (ALP) e a evasão cursos. É preciso, pois, atacar esse problema. Como docente de disciplinas relacionadas a ALP no Ensino Superior na modalidade presencial, posso a experiência de lecionar em turmas de Análise e Desenvolvimento de Sistemas e também em cursos de Engenharias Civil (ECV), Engenharia Elétrica (EEL), Engenharia Química (EQM) e Engenharia de Produção (EPR). No tocante às Engenharias, na maioria das vezes, os alunos são reunidos em uma única classe com o intuito de otimizar carga horária de professores ou utilização de laboratórios por exemplo. Durante diversos semestres pude experimentar metodologias variadas em sala, contudo, o aluno ainda questiona a necessidade da aprendizagem de programação e o uso das habilidades de algoritmos em sua formação e profissão.

Ao longo dos anos venho incrementando exercícios que relacionem a necessidade dos saberes de outras disciplinas na resolução de problemas. Os materiais desenvolvidos, exercícios, trabalhos de implementação, provas e exames passaram a ter questões com problemas de áreas relativas aos cursos de Engenharia e, para ajudar nessa tarefa, contar com um ambiente que possa auxiliar na personalização de conteúdos para cada curso se torna necessário.

Uma das maneiras de se fazer isso desponta na forma de propostas de desenvolvimento, utilizando as Tecnologias da Informação e da Comunicação (TIC), de ambientes *online* de aprendizagem como apoio ao ensino presencial. Neste quesito, o AdaptWeb surge como a ferramenta que possibilita a criação e avaliação das disciplinas personalizadas. Este guia ilustrará o caminho para criação de OA's personalizados no ambiente *online* de ensino-aprendizagem AdaptWeb.

## 2 ADAPTWEB

Os Sistemas Hipermídia Adaptativos (SHA) tem o objetivo de facilitar a navegação e apresentação de conteúdo na Web, entregando para o aluno uma apresentação mais adaptada aos seus interesses melhorando assim a experiência do aluno. Koch (2000) afirma que o SHA busca resolver a sobrecarga cognitiva e a desorientação através da adoção de uma aproximação centrada no usuário. O usuário (aluno) é visto pelo sistema através do modelo de usuário e então o sistema se adapta a ele, exibindo conteúdo e apresentação adequada. Assim cada usuário possui uma visão individual e possibilidades navegacionais particularizadas na utilização com este sistema. Os SHA's são indicados em situações nas quais há a necessidade de oferecer informação seletiva e contextual, para usuários com objetivos e níveis de conhecimento distintos (PALAZZO, 2002), especialmente em cenários educacionais (BRUSILOVSKY 2001; PALAZZO, 2002). Esses sistemas buscam oferecer uma interface modelada conforme as características específicas de cada usuário, ou seja, o estilo, conteúdo, recursos e links das interfaces são dinamicamente selecionados para que sejam apresentados conforme as necessidades, preferências e anseios de um usuário específico (PALAZZO, 2002).

Em síntese, SHA's surgiram como a associação de Sistemas Hipermídia combinado com as características únicas do usuário apresentadas no modelo de usuário. SHA's realizam a adaptação de conteúdo multimídia com recursos hipermídia originados de diferentes formatos e fontes que seja adequado ao perfil de seus usuários (BRUSILOVSKY 2001; PALAZZO, 2002).

Por sua vez, um Ambiente Virtual de aprendizagem (AVA), também denominado *e-learning*, consiste em um ambiente (portal, página, blog, etc) onde são oferecidos conteúdos e atividades organizadas permitindo a prática da aprendizagem significativa, contextualizada e colaborativa (ATANASIO, 2006) disponibilizadas por professores aos seus alunos. Este ambiente deve organizar a informação de maneira original de forma que o visitante tenha uma agradável experiência em sua navegação e meios livres para exprimir o seu pensamento, reduzindo barreiras na transmissão do conhecimento permitindo ainda a colaboração entre os pares com a troca de experiências. O ambiente de aprendizagem consiste em um sistema com suporte a

diversos tipos de atividades a serem realizadas pelo aluno, por meio de um conjunto de ferramentas disponíveis ao uso em diferentes situações do processo de ensino e aprendizagem. Neste contexto, esses ambientes possibilitam a interação do aluno com o objeto de estudo em questão de maneira a integrá-lo à realidade do aluno.

Um AVA não dispõe de salas de aula física, porém, apresenta componentes e características relacionadas como: o aluno, o professor, disponibilidade de materiais e conteúdos didáticos, avaliação de aprendizagem, um tutor (eletrônico ou não), sistemas de gerenciamento dos registros dos alunos e suas atividades, OA's, mídias, etc.

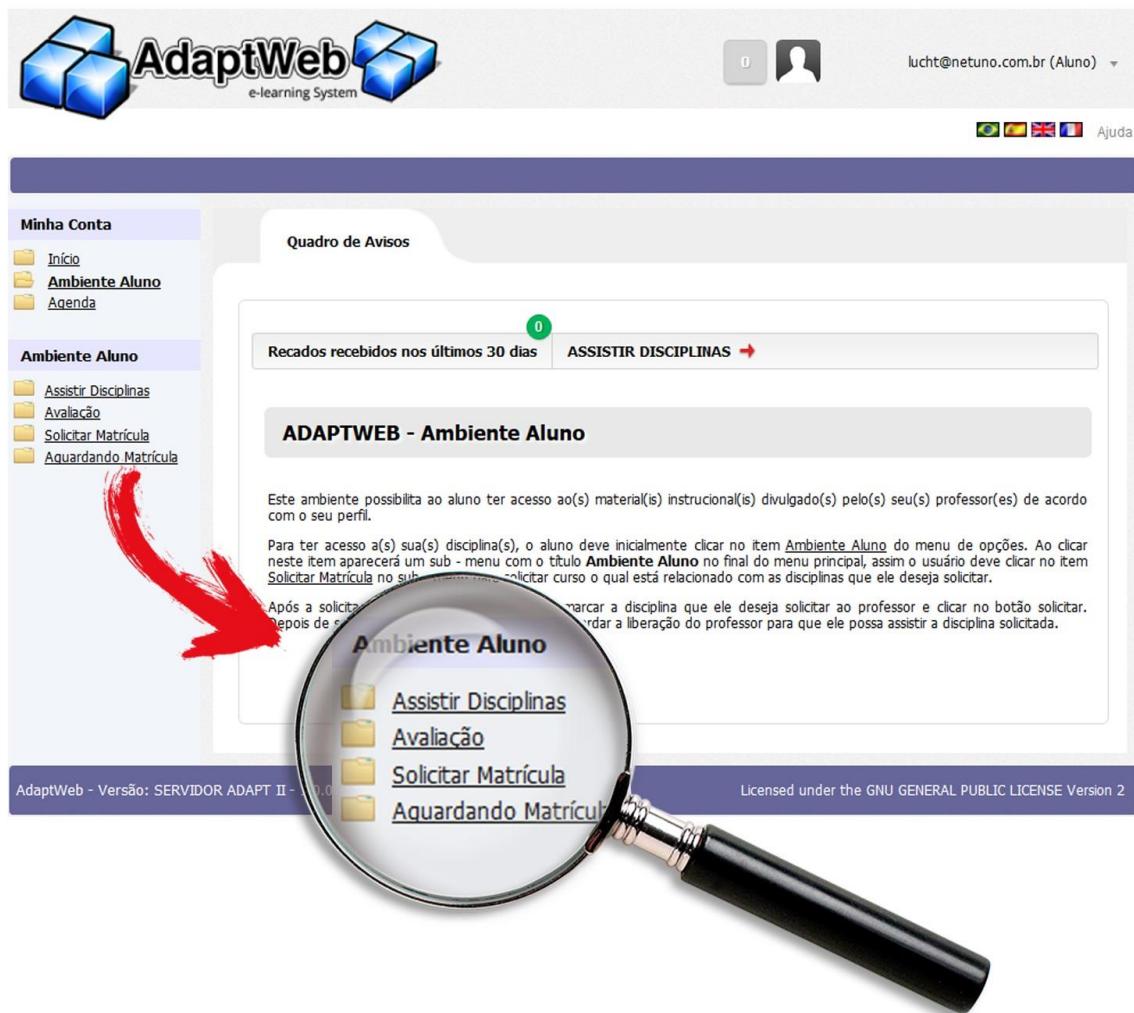
O Minicurso de Algoritmos e Programação (MAP) surgiu em 2015 no Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina (CCT-Udesc) como fonte de pesquisa da professora orientadora deste trabalho. Esse minicurso *online* se baseia em um SHA chamado AdaptWeb (GASPARINI *et al.*, 2003), um SHA educacional que adapta conteúdo, apresentação e a navegação baseado no modelo do usuário. Este ambiente *open source* consiste num sistema *online* baseado em uma estrutura hierárquica de conceitos e critérios pré-estabelecidos e armazenados no formato *Extensible Markup Language* (XML), permitindo a adaptação de acordo com o perfil do aluno.

Sob o ponto de vista de um usuário o AdaptWeb é apresentado como dois ambientes distintos: Ambiente do Aluno e o Ambiente de Autoria (para o professor). A visão geral do ambiente aluno possui:

- Início: direciona o aluno para a tela inicial do ambiente;
- Ambiente Aluno: é a tela representada pela Figura 1;
- Agenda: consiste em uma ferramenta útil que permite o professor agendar eventos (avisos, avaliação, trabalho, atividade, anotação...) adicionando um nome para esse evento, um texto livre como descrição e ainda possibilita anexar arquivos;
- Assistir Disciplina: link que levará o aluno a(s) disciplina(s) autorizadas para serem cursadas;
- Avaliação: após a conclusão da disciplina, caso cadastrada, o aluno poderá acessar uma avaliação disponibilizada pelo professor;
- Solicitar Matrícula: atalho que permite ao aluno solicitar matrícula no(s) curso(s) liberado(s) pelo professor;

- Aguardando Matrícula: usado para visualizar as matrículas ainda não aprovadas.

Figura 1 - Ambiente do Aluno.



Fonte: Minicurso de algoritmos (2019).

Já o Ambiente de Autoria (Figura 2) possui um grupo de atalhos que permite ao professor as mesmas atividades do Ambiente do Aluno e ainda dispõe de um grupo de ferramentas que lhe proporciona a criação e gestão de disciplinas, solicitações de matrículas, gerência sobre seus cursos e disciplinas (inclusões, alterações, exclusão), liberação das disciplinas, confirmação das solicitações de matrícula, ferramentas de análise dos alunos. Possui ainda um banco de questões que permite a inclusão, alteração e exclusão de exercícios que serão associadas posteriormente aos tópicos e a Avaliação Final.

Figura 2 - Ambiente de Autoria (Professor).

Fonte: Minicurso de algoritmos (2019).

Este Produto Educacional tem por objetivo desmistificar esse Ambiente de Autoria de modo que o leitor possa a partir deste, criar seus cursos, relacionar disciplinas e principalmente, cadastrar conteúdos, materiais, links, **exercícios** e **exemplos**. Dessa forma, e com um pouco de prática, o professor poderá utilizar o AdaptWeb como uma opção para a criação de cursos online aproveitando as características adaptativas e multimídia que a plataforma apresenta.

Para conduzir o leitor no processo de criação de OA's personalizados no ambiente AdaptWeb, alguns esclarecimentos sobre os termos e estruturas utilizadas são apresentados a seguir.

### 3 CRIAÇÃO DE OBJETOS DE APRENDIZAGEM PERSONALIZADOS NO AMBIENTE ADAPTWEB

Para L'Allier (1997) um OA consiste em um componente digital, unidade, lição que pode ser reutilizado para a criação de novas estruturas instrucionais singulares. Dessa forma ele define OA como sendo a menor estrutura instrucional independente que possui objetivo, atividade de aprendizagem e uma avaliação.

De acordo com a IEEE-LTSC P1484.12 (2000), um OA pode ser definido como qualquer entidade digital ou não, que pode ser referenciada, utilizada ou ainda reutilizada durante o processo de aprendizagem suportado pela tecnologia.

A criação dos OA's decorreu de uma motivação específica com diferentes dificuldades inerentes ao domínio dos conteúdos abordados. Para contextualizar melhor o problema, o CCT-Udesc oferta regularmente nove cursos de graduação: Ciência da Computação (BCC), Tecnologia em Análise e Desenvolvimento de Sistemas (TADS), Engenharia Civil (ECV), Engenharia Elétrica (EEL), Engenharia Mecânica (EMC), Engenharia de Produção (EPR), Licenciatura em Física (FIS), Licenciatura em Matemática (MAT) e Licenciatura em Química (QUI). Estudos realizados sobre o Projeto Pedagógico de Curso de cada um dos cursos citados constatou que somente o curso de QUI não possuía a disciplina de ALP (ou equivalente) em sua matriz curricular. Aprofundando-se ainda mais no assunto ficou claro através de pesquisa realizada neste grupo que os alunos das Engenharias eram os que ofereciam resistências, demandavam maiores esforços e esboçavam até repulsa pela disciplina. Diante disso foram ampliados os estudos sobre as ementas disponíveis no site da Udesc e propostos novos exemplos e exercícios personalizados baseados em suas matrizes curriculares.

Diante disso, convém organizar em etapas o desenvolvimento do minicurso de ALP antes do uso da ferramenta para o alcance do seu objetivo.

#### 3.1 ORGANIZAÇÃO E ESTRUTURAÇÃO DO CONTEÚDO

Antes da criação de uma disciplina, é necessário que o professor organize dentro da estrutura do minicurso o material a ser disponibilizado. Neste processo os

objetivos da disciplina, sequência didática, conteúdos abordados devem ser organizados em tópicos que, no AdaptWeb recebem o nome de conceitos.

A estrutura do minicurso de ALP foi proposto por Santos (2017). Neste trabalho foram incluídos exemplos e exercícios personalizados. A Figura 3 apresenta a estrutura e organização original (em cinza) com a quantidade de exercícios e exemplos por tópico e subtópico, e os novos exemplos e exercícios elaborados (em verde).

Figura 3 – Estrutura do minicurso com os novos exemplos e exercícios.

Conceito	Gerais		Adaptados para Engenharias								
	Exemplos existentes	Exercícios existentes	Exemplos genéricos	Exercícios Genéricos	Exemplos Civil	Exercícios Civil	Exemplos Elétrica	Exercícios Elétrica	Exemplos Mecânica	Exercícios Mecânica	Exemplos Produção
1 Seja bem-vindo ao minicurso											
1.1 Objetivo do minicurso											
1.2 Ferramentas de aprendizagem											
1.3 O que você vai ver neste curso?											
1.4 Como estudar?											
2 Da lógica à programação											
2.1 Por que aprender a programar?											
2.2 Aplicações de algoritmos											
2.3 Algoritmos e programação											
3 Interpretadores e compiladores											
3.1 Interpretadores de algoritmos (Portugol IDE)											
3.2 Compiladores de linguagem C (DevC++)											
3.3 Outras ferramentas de aprendizagem											
4 Algoritmos	6										
4.1 Criando um algoritmo											
4.2 Tipos de algoritmo											
5 Sobre narrativas	1										
5.1 Tipos de narrativas	3										
5.2 Exercitando as narrativas	3										
6 Sobre pseudocódigos											
6.1 Estrutura de um pseudocódigo											
6.2 Predefinições para escrita de pseudocódigos											
7 Programa	6										
7.1 Principais conceitos da linguagem C											
7.2 Estrutura de um programa											
7.2.1 main()											
7.2.2 system()											
7.2.3 #include											
7.3 Constantes e comandos de atribuição	8										
7.4 Comandos de entrada e saída	4										
7.4.1 printf											
7.4.2 scanf											
7.5 Strings	6										
8 Dados, variáveis e operadores	11	1	1	1	1	1	1	1	1	1	1
8.1 Dados	1										
8.2 Operadores básicos	4	1	1	1	1	1	1	1	1	1	1
8.3 Operadores aritméticos em C											
8.4 Operadores relacionais											
8.5 Operadores lógicos											
8.6 Precedência entre todos os operadores											
9 Estrutura de controle: Decisão	9	1	1	1	1	1	1	1	1	1	1
9.1 Estrutura de decisão simples	6	1	1	1	1	1	1	1	1	1	1
9.2 Estrutura de decisão composta	6	1	1	1	1	1	1	1	1	1	1
9.3 Estrutura de decisão encadeada	2										
10 Estrutura de controle: Laços (repetição)	2	1	1	1	1	1	1	1	1	1	1
10.1 Laço while (enquanto)	4	1	1	1	1	1	1	1	1	1	1
10.2 Laço do... while (repita)	2										
10.3 Laço for (para)	2	1	1	1	1	1	1	1	1	1	1
11 Vetores e matrizes	1		1	1	1	1	1	1	1	1	1
11.1 Vetores	2	1	1	1	1	1	1	1	1	1	1
11.2 Matrizes	1	1	1	1	1	1	1	1	1	1	1

Fonte: Minicurso de algoritmos (2019).

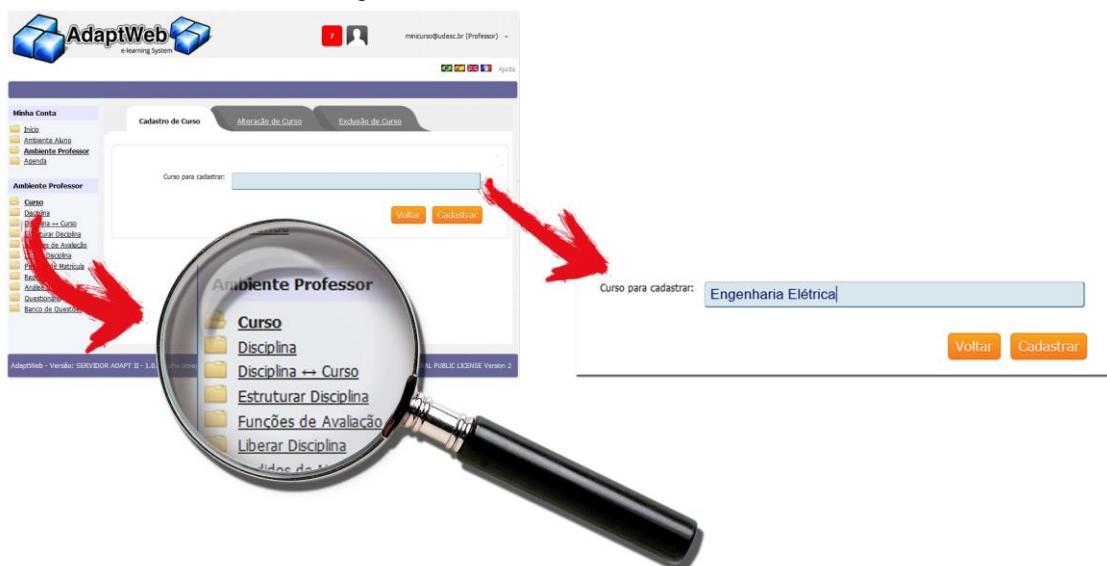
A opção de iniciar a criação de OA's a partir do tópico 8 deveu-se ao fato de que antes desse ponto na estrutura do curso não há características ou aplicabilidade de conceitos a ser personalizada. Trata-se de fundamentos que são desenvolvidos igualmente para qualquer curso.

O conjunto de exercícios e exemplos foram criados de maneira a estabelecer relação com os conteúdos apresentados aos estudantes nas suas respectivas salas de aula. Nota-se também que alguns enunciados poderiam ser reaproveitados em mais de um curso, porém, optou-se por escolher aqueles que mais se aderem a proposta do curso em questão. Além disso, novos exercícios e exemplos também foram criados para os cursos de Licenciatura em Física e Matemática, Ciência da Computação e Análise e Desenvolvimento de Sistemas (enunciados esses que abordam características em comum aos cursos relacionados: álgebra e cálculo).

### 3.2 CADASTRO DE CURSO

Para criar um novo curso no AdaptWeb o professor precisa realizar uma sequência de passos. No menu “Ambiente do Professor” à esquerda da tela, o link “Curso” irá levar o professor para a tela de cadastro de Curso. Aí basta preencher com o nome do curso que se deseja criar e em seguida clicar no botão “Cadastrar” para efetivar a gravação de um novo curso. A Figura 4 ilustra o cadastro de um novo curso “Engenharia Elétrica” – uma das Engenharias do minicurso de ALP.

Figura 4 – Cadastro de novos Cursos.



Fonte: Elaborado pelo Autor (2019).

Para o experimento do Minicurso de Algoritmos 2019 - ALG2019-01 foram cadastrados os seguintes Cursos: Análise e Desenvolvimento de Sistemas, Ciência da Computação, Engenharia Civil, Engenharia Elétrica, Engenharia Mecânica, Engenharia da Produção e Sistemas, Física e Matemática. Nesta tela ainda pode-se visualizar as opções para “Alteração de Curso” e ainda, “Exclusão de Curso”.

### 3.3 CADASTRO DE DISCIPLINA

Depois dos cursos cadastrados, a etapa seguinte consiste no cadastro da(s) disciplina(s). Diversas disciplinas com seus respectivos assuntos podem ser cadastradas e associadas aos cursos. Neste produto educacional criou-se a disciplina chamada de **Minicurso de Algoritmos 2019 – ALG2019-01**. Segue a sequência de passos a ser realizada apresentada na Figura 5.

Figura 5 – Cadastro de Disciplina.

Minha Conta

- Início
- Ambiente Aluno
- Ambiente Professor**
- Agenda

Ambiente Professor

- Curso
- Disciplina**
- Disciplina → Curso
- Estruturar Disciplina
- Funções de Avaliação
- Liberar Disciplina
- Pedidos de Matrícula
- Repositório
- Análise Interativa
- Questionário
- Banco de Questões

Cadastro de Disciplina

O campo “Identificador” auxilia na identificação da disciplina para consultas futuras. Ela pode ser preenchida com o ano, semestre, turma, etc. EXEMPLO: 2013-1 Sem.

Disciplina para cadastrar:

Identificador para Disciplina:

Gamificação:  Ativada  Desativada

Posições no Ranking: 5

Voltar Cadastrar

AdaptWeb - Versão: 5.0.0 - Liberado em: 2019-01-01 - Copyright © 2019 - ALG2019-01 - Licensed under the GNU GENERAL PUBLIC LICENSE Version 2

Disciplina para cadastrar: Minicurso de Algoritmos 2019 - ALG2019-01

Identificador para Disciplina: ALG2019-01

Gamificação:  Ativada  Desativada

Posições no Ranking: 5

Fonte: Elaborado pelo Autor (2019).

A opção de ativação da Gamificação permite que o aluno acompanhe o seu desempenho dinamicamente durante o curso através de métricas pré-definidas pelo AdaptWeb. Da mesma forma como no cadastro anterior dos cursos, na tela de cadastro de Disciplina é permitido ao professor Alterar ou Excluir uma disciplina já existente.

Outro detalhe muito importante nesta etapa de inserção de cursos e disciplina no AdaptWeb consistem em fazer a “relação” do(s) curso(s) com a disciplina que se está criando (ou já criada). Essa característica do AdaptWeb permite associar uma disciplina com diversos cursos favorecendo ao professor por não precisar criar individualmente e repetidamente uma disciplina inteira quando aplicada à diversos cursos e já tornando a disciplina propícia a ser personalizada. Com a simples marcação nas caixinhas o professor pode relacionar os cursos que terão acesso aos conteúdos de sua disciplina conforme a Figura 6.

Figura 6 – Associação da Disciplina com Curso(s).

**Minha Conta**

- Início
- Ambiente Aluno
- Ambiente Professor**
- Agenda

**Ambiente Professor**

- Curso
- Disciplina
- Disciplina ↔ Curso**
- Escrever Disciplina
- Funções de Avaliação
- Liberar Disciplina
- Pedidos de Matrícula
- Repositório
- Análise Interativa
- Questionário
- Banco de Questões

**Disciplina/Curso**

Para que seja possível estruturar o conteúdo de uma disciplina, a disciplina deve estar vinculada a pelo menos um curso.

Disciplinas cadastradas: Minicurso de Algoritmos 2019-01 - ALG2019-01

Vincular disciplina com os cursos:

- Análise e Desenvolvimento de Sistemas
- Ciência da Computação
- Engenharia Civil
- Engenharia Elétrica
- Engenharia Mecânica
- Engenharia Produção e Sistemas
- Física
- Matemática
- Minicurso 2016/2
- Minicurso 2017/1-1
- Minicurso 2017/1-2
- Minicurso 2018-1
- Outros Cursos do Minicurso 2018-2

Voltar Gravar Alterações

### 3.4 CADASTRO DOS CONCEITOS PRINCIPAIS E SECUNDÁRIOS (SEÇÕES)

Nesta fase os materiais criados são inseridos na estrutura da disciplina. Conforme demonstra a Figura 7, o professor deve clicar no menu “Estruturar Disciplina” no menu à esquerda da tela e então selecionar o título da disciplina já cadastrada para iniciar a sua estruturação.

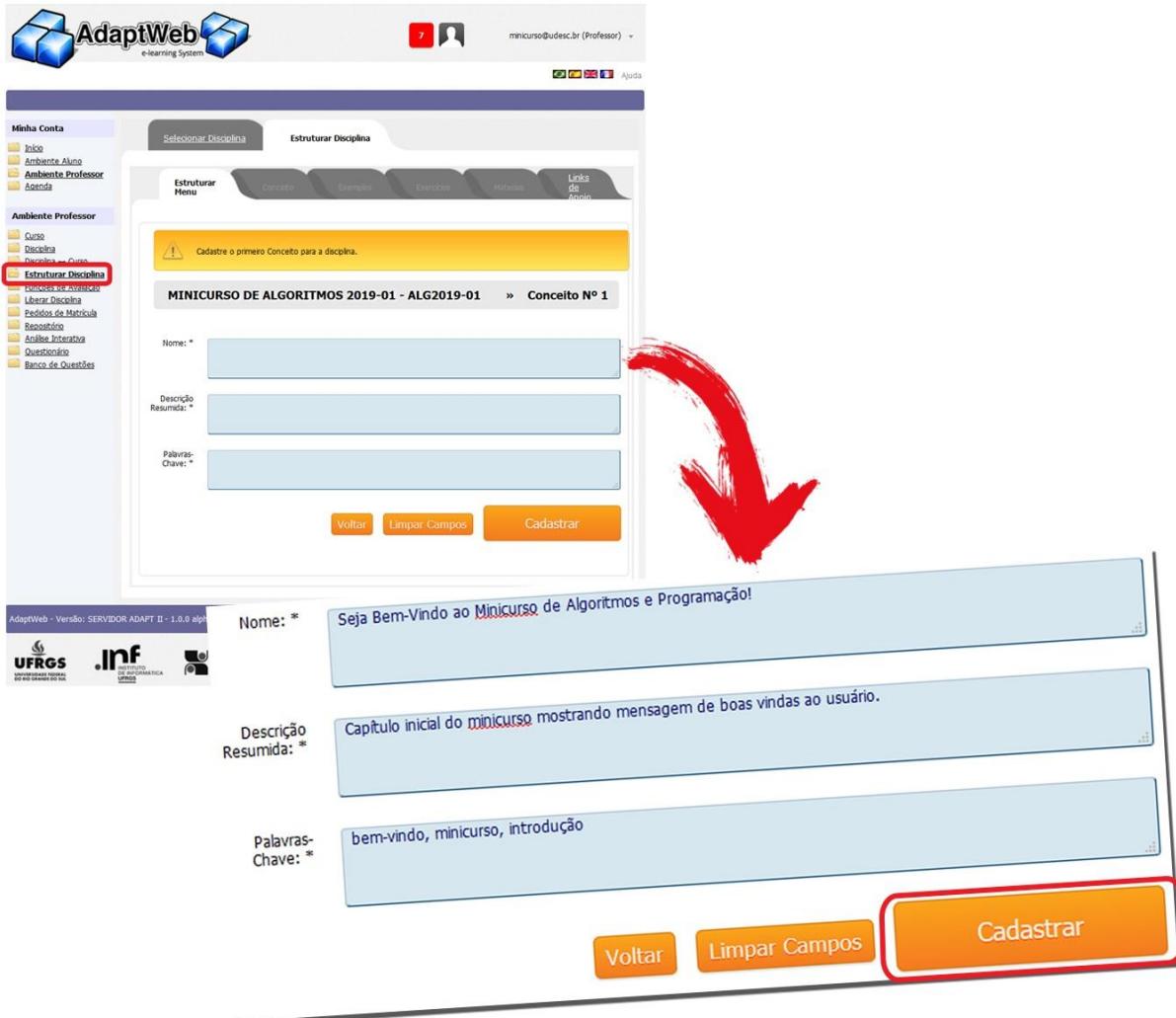
Figura 7 – Início da estruturação da Disciplina.

The screenshot shows the AdaptWeb e-learning System interface. At the top, there is a navigation bar with the logo 'AdaptWeb e-learning System', a user icon, and the email 'minicurso@udesc.br (Professor)'. Below the navigation bar, there are language and help icons. The main area has a purple header bar with the text 'Minha Conta' and 'Ambiente Professor'. On the left, under 'Ambiente Professor', there is a sidebar with several options: 'Início', 'Ambiente Aluno', 'Ambiente Professor' (which is highlighted in blue), and 'Agenda'. A red box highlights the 'Estruturar Disciplina' option in the sidebar. The main content area has tabs 'Selecionar Disciplina' and 'Estruturar Disciplina', with 'Estruturar Disciplina' being active. A green info box says: 'Selecione a disciplina para Estruturar o Conteúdo. OBS: Somente estão listadas as disciplinas vinculadas com pelo menos um curso.' Below this, a dropdown menu shows 'Minicurso de Algoritmos 2019-01 - ALG2019-01'. At the bottom right of the content area are 'Voltar' and 'Estruturar Disciplina' buttons. At the very bottom of the page, there is a footer bar with the text 'AdaptWeb - Versão: SERVIDOR ADAPT II - 1.0.0 alpha omega pi' and 'Licensed under the GNU GENERAL PUBLIC LICENSE Version 2'.

Fonte: Elaborado pelo Autor (2019).

A partir do botão “Estruturar Disciplina” de cor laranja uma nova tela irá surgir. Nesta tela o professor deve começar a transcrever a estrutura concebida na seção 3.1. Todos os exemplos e exercícios deste produto educacional se baseiam no experimento realizado. É obrigatório o preenchimento do conceito inicial da disciplina, bem como a descrição resumida e as palavras chave. O clique em Cadastrar irá confirmar a entrada de dados conforme a Figura 8.

Figura 8 – Cadastrando o primeiro conceito do minicurso.



Fonte: Elaborado pelo Autor (2019).

Após a confirmação a partir do botão laranja em desta tela o professor terá criado o Conceito nº1 de sua disciplina criando assim o primeiro tópico da estrutura do minicurso (Figura 9). Vale a atenção para o título da tela (MINICURSO DE ALGORITMOS 2019-01 - ALG2019-01 » Conceito N° 1) que serve como referência de qual conceito na estrutura o professor está criando ou editando. Para finalizar o cadastro do conceito nº1 (ou de outro conceito qualquer) é preciso voltar ao menu Estrutar Disciplina, selecionar o respectivo conceito com o mouse (Figura 9), preencher os campos e selecionar o arquivo (.HTM) referente ao conceito que se está cadastrando (Figura 10).

Figura 9 – Editando um conceito.



Fonte: Elaborado pelo Autor (2019).

Figura 10 – Confirmando dados e associações com os diversos cursos.

The screenshot displays the AdaptWeb e-learning system interface. On the left, there's a sidebar with navigation links for 'Minha Conta' (Início, Ambiente Aluno, Ambiente Professor, Agenda) and 'Ambiente Professor' (Curso, Disciplina, Disciplina --> Curso, Estruturar Disciplina, Funções de Avaliação, Liberar Disciplina, Pedidos de Matrícula, Repositório, Análise Interativa, Questionário, Banco de Questões). The main content area is titled 'ESTRUTURA DE DISCIPLINA' and shows a sub-section for 'Conceito N° 1'. The 'Conceito N° 1' page contains fields for 'Nome' (Seja Bem-Vindo ao Minicurso de Algoritmos e Programação!) and 'Descrição Resumida' (Boas vindas aos estudantes.). There's also a 'Palavras-Chave:' field containing 'Boa Vindas, Ciéncia da Computação, TADS, Licenciatura Física, Licenciatura Matemática, Engenharia Mecânica, Engenharia Elétrica, Engenharia Civil, Engenharia Produção, Seja Bem Vindos ao Minicurso de Algoritmos e Programação!'. A 'Cursos Vinculados ao Conceito:' section lists various engineering disciplines with checkboxes. Below this is a 'Pré-Requisitos:' section with a note about selecting multiple prerequisites using the CTRL key. At the bottom, there are buttons for 'Voltar para Estruturar Menu' and 'Gravar as alterações'. Further down, there's a section for 'Enviar Arquivos do Conceito:' with a file input field and a 'Enviar arquivo do computador' button, along with an option to 'Criar documento novo'. At the very bottom, there's footer information about the system version and partners.

Fonte: Elaborado pelo Autor (2019).

Uma disciplina cadastrada pode ser associada com vários cursos distintos, onde cada disciplina poderá ter seus conceitos, exercícios, exemplos e materiais complementares específicos.

Como passo seguinte, é permitido ao professor cadastrar um novo conceito (a) ou verificar pendências (b) (Figura 11). Caso a escolha do professor seja a opção (a), a tela da Figura 8 voltará a ser apresentada, porém, com o título “MINICURSO DE ALGORITMOS 2019-01 - ALG2019-01 » Conceito Nº 2”.

Figura 11 – Cadastrando o primeiro conceito do minicurso.

The screenshot shows the AdaptWeb e-learning system interface. At the top, there is a navigation bar with the logo 'AdaptWeb e-learning System', user icons, and the email 'minicurso@udesc.br (Professor)'. Below the navigation bar, the main menu includes 'Minha Conta' (My Account), 'Ambiente Professor' (Professor Environment), and 'Selecionar Disciplina' (Select Discipline). The 'Selecionar Disciplina' section is active, showing tabs for 'Estruturar Menu' (Structure Menu), 'Conceito' (Concept), 'Exemplos' (Examples), 'Exercícios' (Exercises), 'Materiais' (Materials), and 'Links de Apoio' (Support Links). A green callout box provides instructions: 'Passe o mouse em cima dos conceito para Cadastrar ou Excluir itens.' and 'Acesse o conceito desejado e adicione exemplos, exercícios e materiais complementares. Terminada a criação do conceito, clique em VERIFICAR PENDÊNCIAS e por fim clique em GERAR CONTEÚDO para disponibilizar a disciplina aos alunos.' Below this, a sub-menu for 'MINICURSO DE ALGORITMOS 2019-01 - ALG2019-01' is shown with options 'Cadastrar / Excluir Conceitos' and 'Minicurso de Algoritmos 2019-01 - ALG2019-01'. A red box highlights the message '1 - Seja Bem-Vindo ao Minicurso de Algoritmos e Programação!'. At the bottom of this section are buttons for 'Voltar para Selecionar Disciplina', 'Cadastrar Conceito 2', and 'Verificar Pendências'. The 'Cadastrar Conceito 2' button is labeled '(a)'. The 'Verificar Pendências' button is labeled '(b)'. The bottom part of the screenshot shows another section titled 'MINICURSO DE ALGORITMOS 2019-01 - ALG2019-01 B » Movimentar Conceitos'. It includes fields for 'Conceito que será movimentado:' (with a dropdown menu containing '1-Seja Bem-Vindo ao Minicurso de Algoritmos e Programação!') and 'Movimentar APÓS qual conceito?' (with a dropdown menu containing '1-Seja Bem-Vindo ao Minicurso de Algoritmos e Programação!'). A large orange button at the bottom right is labeled 'Movimentar Conceito'. The footer of the page includes links to 'AdaptWeb - Versão: SERVIDOR ADAPT II - 1.0.0 alpha omega pi', 'Licensed under the GNU GENERAL PUBLIC LICENSE Version 2', and logos for UFRGS, INF, UDESC Joinville, and CNPq.

Fonte: Elaborado pelo Autor (2019).

No entanto, se a opção escolhida for (b), o AdaptWeb fará uma verificação de pendências e apresentará um relatório com as inconsistências encontradas (caso existam). A Figura 12 mostra que o primeiro conceito cadastrado não possui pendências.

Figura 12 – Verificação de Pendências.

Conceito						
Número Tópico	Nome Tópico	Descrição	Palavra-chave	Curso	Arquivo Principal	Arquivos Associados
1	Seja Bem-Vindo ao Minicurso de Algoritmos e Programação!	Ok	Ok	ok	Ok	Não possui

Fonte: Elaborado pelo Autor (2019).

Para a criação de um subtópico de um conceito (1.1, 1.2, ...), o professor precisa posicionar o mouse sobre o conceito a ser estruturado e selecionar a opção marcada na Figura 13.

Figura 13 – Cadastrando subtópicos.



Fonte: Elaborado pelo Autor (2019).

No ambiente de autoria, a visualização da estrutura em fase de inserção pode ser visualizada na Figura 14. Na parte esquerda da figura enxerga-se o primeiro conceito (1 – Seja Bem-Vindo...) e na parte direita da mesma figura, após o cadastro de toda a estrutura, a visualização completa do Minicurso de Algoritmos 2019 - ALG2019-01.

Figura 14 – Cadastro de conceitos e seções do minicurso.

The screenshot displays two main sections of the AdaptWeb interface:

- Cadastro de Conceito (Concept Registration):** This section shows a list of concepts registered for the course. A large red arrow points from the left towards this list. The concepts listed include:
  - 1 - Seja Bem-Vindo ao Minicurso de Algoritmos e Programação -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 1.1 - Observação do minicurso -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 1.2 - Ferramentas de Aprendizagem -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 1.3 - Como usar o minicurso -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 1.4 - Como estudar -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 2 - Minicurso de Algoritmos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 2.1 - Por que aprender a programar? -- (Física) [Engenharia Civil] [Engenharia Elétrica]
  - 2.2 - Aplicação de Algoritmos e Programação -- (Física) [Engenharia Civil] [Engenharia Elétrica]
  - 2.3 - Algoritmos e Programação -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 3 - Interpretadores e compiladores -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 3.1 - Interpretadores de algoritmos (Português) -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 3.2 - Compiladores de linguagens de programação -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 3.3 - Outras ferramentas de programação -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 4 - Algoritmos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 4.1 - Criando um algoritmo -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 4.2 - Tipos de algoritmos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 5 - Sobre narrativas -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 5.1 - Tipos de narrativas -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 5.2 - Exercitando narrativas -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 6 - Sobre procedimentos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 6.1 - Tipos de procedimentos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 6.2 - Procedimento para a realização de procedimentos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7 - Programas -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.1 - Estrutura de um programa -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.2 - Linguagem C - principais conceitos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.2.1 - inteiros -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.2.2 - sistemas -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.2.3 - processos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.3 - Estrutura de um processo -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.4 - Comandos de Entrada/Saída -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.4.1 - print -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.4.2 - scanf -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 7.5 - Strings -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8 - Dados, variáveis e operadores -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8.1 - Dados -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8.2 - Operadores básicos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8.3 - Operadores aritméticos -- (C) -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8.4 - Operadores lógicos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8.5 - Operadores lógicos -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 8.6 - Prioridades entre todos os Operadores -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 9 - Estrutura de Controle - Decida -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 9.1 - Estrutura de decisão simples -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 9.2 - Estrutura de decisão composta -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 9.3 - Estrutura de decisão aninhada -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 10 - Estrutura de Repetição - Laço -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 10.1 - Laço while (impar) -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 10.2 - Laço do... while -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 10.3 - Laço for (para) -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 11 - Vetores e matrizes -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 11.1 - Vetores -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
  - 11.2 - Matrizes -- (Física) [Engenharia Civil] [Engenharia Elétrica] [Engenharia Mecânica]
- Movimentar Conceitos (Move Concepts):** This section shows a list of concepts that have been moved. A yellow box at the top states: "A movimentação de um conceito move juntamente todos os seus subconceitos." Below this, it lists the moved concepts:
  - MINICURSO DE ALGORITMOS 2019-01 - ALG2019-01 B » Movimentar Conceitos
  - Conceito que será movimentado: 1-Seja Bem-Vindo ao Minicurso de Algoritmos e Programação...
  - Movimentar APÓS qual conceito? 1-Seja Bem-Vindo ao Minicurso de Algoritmos e Programação...
  - Movimentar Conceito

Fonte: Elaborado pelo Autor (2019).

Os exemplos, exercícios e materiais complementares precisam estar relacionados aos respectivos conceitos. É necessário bastante atenção durante esta etapa de estruturação para que os conceitos e subtópicos sejam inseridos corretamente, associando os arquivos correspondentes para cada conteúdo.

Trechos da grade de Conceitos, Material Complementar e Exemplo do minicurso implementado podem ser visto na Figura 15. Ao *click* do botão Gerar Conteúdo, o AdaptWeb irá processar todos os cadastros e deixará o minicurso pronto para a liberação para os alunos. A mensagem final apresentada pela ferramenta está no rodapé da Figura 15.

Figura 15 – Verificação de Pendências.

Número Tópico	Nome Tópico	Descrição	Palmácia	Cury	Arquivo Principal	Arquivos Associados
	Saia Bem-Vindo ao Módulo de Algoritmos e Programação		Ok	Ok	Ok	Não possui
1.1	Objetivo do módulo		Ok	Ok	Ok	Não possui
1.2	Ferramentas de Aprendizagem		Ok	Ok	Ok	Não possui
1.3	O que você vai ver neste módulo?		Ok	Ok	Ok	Não possui
1.4	Como estudar		Ok	Ok	Ok	Não possui
2	Da lógica à programação		Ok	Ok	Ok	Não possui
2.1	Por que aprender a programar?		Ok	Ok	Ok	Não possui
2.2	Avaliação de Algoritmos e Programação		Ok	Ok	Ok	Não possui
2.3	Algoritmos e Programação		Ok	Ok	Ok	Não possui
3	Interpretações e compreensões		Ok	Ok	Ok	Não possui
3.1	Interpretadores de algoritmos (Português IDE) (DEV-C++)		Ok	Ok	Ok	Não possui
3.2	Interpretadores de linguagens (DEV-C++)		Ok	Ok	Ok	Não possui
3.3	Outras ferramentas de interpretação		Ok	Ok	Ok	Não possui
4	Algoritmos		Ok	Ok	Ok	Não possui
4.1	Criando um algoritmo		Ok	Ok	Ok	Não possui
4.2	Tipo de um algoritmo		Ok	Ok	Ok	Não possui
4.3	Sobre narrativas		Ok	Ok	Ok	Não possui
4.5	Tipos de narrativas		Ok	Ok	Ok	Não possui
5	Exercitando narrativas		Ok	Ok	Ok	Não possui
6	Algoritmos compostos		Ok	Ok	Ok	Não possui
6.1	Estrutura de um pseudocódigo		Ok	Ok	Ok	Não possui
6.2	Aplicação de estrutura para escrita de pseudocódigos		Ok	Ok	Ok	Não possui
7	Programação		Ok	Ok	Ok	Não possui
7.1	Entendendo o que é um programa		Ok	Ok	Ok	Não possui
7.2	Linguagem C - principais conceitos		Ok	Ok	Ok	Não possui

Número Tópico	Nome Tópico	Número	Descrição	Cury	Arquivo Principal	Arquivos Associados
	Material Complementar					
1	Leia Bem-Vindo	1	Seja bem-vindo	Ok	Ok	Não possui
		2	Créditos	Ok	Ok	Não possui
		3	Seja bem-vindo	Ok	Ok	Não possui
		4	Créditos	Ok	Ok	Não possui
1.1	Objetivo do módulo					
1.2	Ferramentas de Aprendizagem					
1.3	O que você vai ver neste módulo?					
1.4	Como estudar					
2	Da lógica à programação					
2.1	Por que aprender a programar?					
2.2	Avaliação de Algoritmos e Programação					
2.3	Algoritmos e Programação					
3	Interpretações e compreensões					
3.1	Interpretações de algoritmos (Português IDE)	1	Outros interpretadores de pseudocódigo	Ok	Ok	Não possui

Número Tópico	Nome Tópico	Número	Descrição	Cury	Arquivo Principal	Arquivos Associados
	Exemplo					
1	Saia Bem-Vindo ao Módulo de Algoritmos e Programação	1	Objetivo do módulo	Ok	Ok	Não possui
1.2	Ferramentas de Aprendizagem	1.2	O que você vai ver neste módulo?	Ok	Ok	Não possui
1.3	O que você vai ver neste módulo?	1.3	Como estudar	Ok	Ok	Não possui
1.4	Como estudar	1.4	Da lógica à programação	Ok	Ok	Não possui
2	Da lógica à programação	2	Por que aprender a programar?	Ok	Ok	Não possui
2.2	Avaliação de Algoritmos e Programação	2.2	Avaliação de Algoritmos e Programação	Ok	Ok	Não possui
2.3	Algoritmos e Programação	2.3	Algoritmos e Programação	Ok	Ok	Não possui
3	Interpretações e compreensões	3	Interpretações e compreensões	Ok	Ok	Não possui
3.1	Interpretações de algoritmos (Português IDE)	3.1	Interpretações de algoritmos (Português IDE)	Ok	Ok	Não possui

#### ARQUIVOS DO CURSO GERADOS COM SUCESSO

X

Fonte: Elaborado pelo Autor (2019).

### 3.5 CADASTRO DE EXEMPLOS E EXERCÍCIOS

Para a inserção de exemplos e exercícios no AdaptWeb, o professor deve observar algumas regras como por exemplo: os exercícios estão sempre relacionados ao tópico principal, ou seja, aos tópicos 1, 2, 3...11 da estrutura de conceitos. Já os exemplos e materiais complementares podem ser inseridos em tópicos principais ou em subtópicos diretamente em uma tela durante a estruturação do menu.

Na inserção de um exemplo específico do curso de Engenharia Elétrica para o conceito “9.1 – Estrutura de decisão simples”, a ordem dos passos é ilustrada na sequência de imagens da Figura 16: (a) tela principal do cadastro de exemplos, (b) ampliação com detalhes do preenchimento a serem inseridos. Todos os campos são de preenchimento obrigatório e a opção de seleção do Nível de Complexidade (Fácil, Médio, Complexo) determinará quantas estrelas o exemplo receberá quando for exibido ao aluno (uma, duas ou três estrelas respectivamente). Ainda na mesma tela deve-se clicar no botão Selecionar Arquivo e buscar no repositório de arquivos

organizado pelo professor o arquivo .HTM correspondente (c). Como passo seguinte, o professor deve clicar em Enviar que está no canto inferior direito dessa tela, finalizando assim a criação de um exemplo. A sequência mostra ainda a visualização deste exemplo na tela do aluno que cursa a disciplina (d) no formato de um link e, ao click deste aluno, a janela de exibição do exemplo cadastrado (e).

Figura 16 – Cadastro e associação de um exemplo a um curso no minicurso.

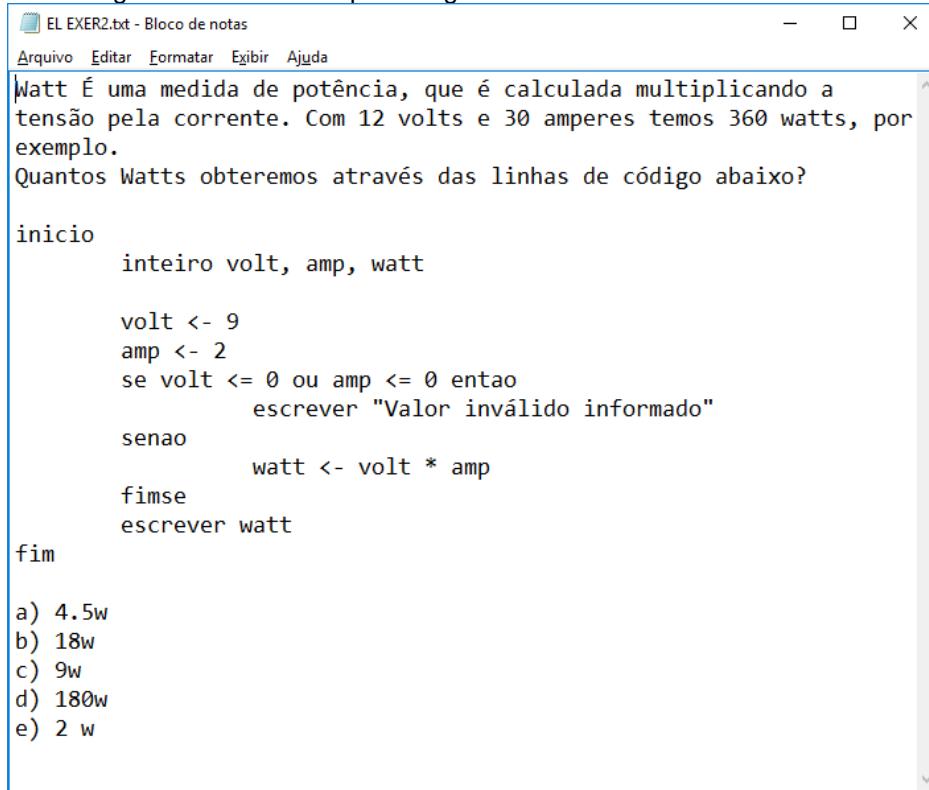


Fonte: Elaborado pelo Autor (2019).

O processo para a inserção dos exercícios requer alguns procedimentos extras, todavia, facilmente de serem executados. Antes de mais nada e como recomendação, a organização dos conteúdos previstos na seção 3.1 faz-se fundamental. Para este minicurso em questão, os exercícios foram individualmente preparados por

assunto/tema em arquivos texto (.TXT) como a Figura 17 exemplifica. Este enunciado representa um exercício do tipo “assinala a alternativa correta” correspondente ao tópico 9 do curso de Engenharia Elétrica cujo tema é “9 – Estrutura de controle: Decisão”.

Figura 17 – Exercício para Engenharia Elétrica em formato .TXT.



```

EL EXER2.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
Watt É uma medida de potência, que é calculada multiplicando a tensão pela corrente. Com 12 volts e 30 amperes temos 360 watts, por exemplo.
Quantos Watts obteremos através das linhas de código abaixo?

inicio
    inteiro volt, amp, watt

    volt <- 9
    amp <- 2
    se volt <= 0 ou amp <= 0 entao
        escrever "Valor inválido informado"
    senao
        watt <- volt * amp
    fimse
    escrever watt
fim

a) 4.5w
b) 18w
c) 9w
d) 180w
e) 2 w

```

Fonte: Elaborado pelo Autor (2019).

Para que os seus exercícios estejam disponíveis em qualquer minicurso, o professor deverá cadastrar os exercícios em uma opção denominada “Banco de Questões” no menu principal do AdaptWeb. Esse banco agrega em uma tela todas as questões disponíveis para serem utilizadas como “exercício” ou “avaliação”. A opção Avaliação possibilita ao professor verificar os conhecimentos dos participantes através de uma prova a ser realizada no próprio ambiente do AdaptWeb no final do período da execução do minicurso. As questões podem ser elaboradas nos formatos: dissertativas, múltipla escolha, verdadeiro/falso e assinalar a correta.

Na Figura 18 é apresentada a tela inicial para cadastro de novos exercícios que poderão ser acrescentados na disciplina. Nesta ilustração usou-se o exercício da Figura 16 e de acordo com o gabarito, a segunda opção é a alternativa correta.

Figura 18 – Criação de um exercício no Banco de Questões.

The screenshot shows the AdaptWeb e-learning System interface. On the left, there's a sidebar with 'Minha Conta' and 'Ambiente Professor' sections. The main area is titled 'Banco de Questões' and 'Nova Questão'. It displays fields for 'Questão de:' (Assessment), 'Disciplina(s)' (Subject), 'Tipo da Questão' (Question Type), 'Enunciado' (Statement), and 'Resposta(s)' (Answers). The 'Enunciado' field contains a snippet of code. The 'Resposta(s)' section lists five options with radio buttons: '4.5w', '18w' (selected), '9w', '180w', and '2 w'. Below the answers, a note says 'É obrigatório fornecer, no mínimo, duas alternativas.' (It is mandatory to provide at least two alternatives.) There are 'Justificativa:' (Justification) and 'Adicionar uma Nova Resposta' (Add a New Response) fields. At the bottom right are 'Voltar' (Back) and 'Salvar' (Save) buttons. The footer includes logos for UFRGS, INF, UDESC Joinville, and CNPq.

Fonte: Elaborado pelo Autor (2019).

Depois do exercício devidamente inserido, a visualização pelo aluno pode ser verificada na Figura 19 onde o enunciado aparece destacado e logo abaixo a janela da execução do exercício com a resposta correta assinalada. Percebe-se que durante o cadastro da questão, a resposta correta era a segunda opção e durante a execução passou a ser a quarta opção. O AdaptWeb faz a aleatorização das opções dos

exercícios para que não se crie o efeito “gabarito” sempre com a mesma sequência formatada.

Figura 19 – Visualização de um exercício pelo aluno.

The screenshot shows a web-based learning environment for algorithmic programming. At the top, there's a navigation bar with icons for 'Mapa', 'Configurações', 'Ajuda', and 'Sair'. Below it, a blue header bar says 'Minicurso de Algoritmos 2019-01 - A ...' and 'Estrutura de Controle - Decisão'. On the left, a sidebar titled 'Categorias' lists sections like 'Conceito', 'Exemplos', 'Exercícios' (which is selected), and 'Materiais Complementares'. The main content area is titled 'Lista de Exercícios para Estrutura de Controle - Decisão'. It contains several questions, each with a star rating and a [Pseudocódigo] link. One question is highlighted with a red box and a red arrow pointing to it. The question asks about calculating power (Watt) given voltage (volt) and current (amp). Below the question, a green box displays the pseudocode:

```

    inicio volt, amp, watt
    inteiro volt, amp, watt
    volt <= 9
    amp <= 2
    se volt <= 0 ou amp <= 0 entao
        escrever "Valor invalido informado"
    senao
        watt <- volt * amp
    fimse
    escrever watt
  
```

Below the pseudocode, a dropdown menu shows options: 4.5W, 9W, 180W, 18W (which is checked), and 2W. A note at the bottom explains that WATT corresponds to VOLT \* AMP. A red box highlights the '18W' option. A red arrow points from the question text in the main window down to this detailed view.

Fonte: Elaborado pelo Autor (2019).

Com isso, nota-se que os procedimentos para a criação de exemplos e exercícios no ambiente AdatpWeb são relativamente simples e exigem conhecimentos básicos em operacionalização de programas. Toda a atenção voltam-se à etapa de planejamento correto da estrutura e organização dos tópicos da disciplina e também

do cadastro correto da(s) resposta(s) dos exercícios no banco de questões para que não ocorram inconsistências.

A representação de todos os novos exemplos e exercícios personalizados cadastrados nesta edição do minicurso estão no Apêndice A. Com o término do processo de autoria, o minicurso estará pronto para ser liberado pelo professor. O passo seguinte consiste no acesso pelo aluno.

## 4 ANÁLISES E RESULTADOS

Todo o desenvolvimento deste trabalho foi pautado na melhoria do ensino aprendizagem, mais especificamente para os alunos da disciplina de ALP dos cursos de Engenharia (ECV, EEL, EMC e EPR). Sendo assim, faz sentido enxergar mais exemplos da criação dos OA's sob a perspectiva do aluno. Na Figura 20 são apresentados dois recortes das telas que contém o link e o exemplo, ambos para um mesmo conteúdo - Conceito 10.3: Laço for (para). É possível notar os exemplos comuns a todos e, com uma pequena indicação do nome do curso, o novo exemplo personalizado.

Figura 20 – Exemplos de EEL e ECV na visão do aluno.

### Lista de Exemplos para Laço for (para)

**Estrutura de controle – Laços (repetição)**

**Exemplo - Pseudocódigo – Watt x Joule**

Um dispositivo de 1 Watt consome 1 Joule de energia a cada segundo. Se você multiplicar o número de watts pelo número de segundos, terá joules como resultado. Desenvolva um algoritmo que solicite a quantidade de watts de uma lâmpada e o tempo (em segundos). Ao final, calcule a quantidade de Joules para 5 diferentes lâmpadas e intervalos de tempo após este período de acordo com a fórmula:  $J = W * s$

```

início
    real W
    inteiro cont, s
    para cont de 1 ate 5 passo 1
        escrever "Informe a quantidade de Watts: "
        ler W
        escrever "Informe a quantidade de tempo em segundos: "
        ler s
        escrever W * s, " Joules \n"
    próximo
fim

```

### Lista de Exemplos para Laço for (para)

**Estrutura de controle – Laços (repetição)**

**Exemplo - Pseudocódigo - Multiplicação**

Desenvolva um algoritmo que calcule a multiplicação de dois números inteiros sem utilizar o operador de multiplicação \*\*\*.

```

início
    inteiro multiplicando, multiplicador, resultado, cont
    escrever "Informe um valor para o Multiplicando: "
    ler multiplicando
    escrever "Informe um valor para o Multiplicador: "
    ler multiplicador
    para cont de 1 ate multiplicador passo 1
        resultado <- resultado + multiplicando
    próximo
    escrever "O resultado é: ",resultado
fim

```

Fonte: Elaborado pelo Autor (2019).

Da mesma maneira é apresentado na Figura 21 dois exercícios personalizados: um para EPR e o segundo para EMC.

Figura 21 – Exercícios de EPR e EMC na visão do aluno.

#### **Lista de Exercícios para Estrutura de controle - Laços (repetição)**

★★★ [Pseudocódigo] Comparando as estruturas "enquanto...faz" e "faz...enquanto", é correto afirmar que:

★★★ O programa obtém do usuário o valor n e imprime na tela os n primeiros termos de uma sequência de Fibonacci. Observe que há uma variável faltando, exatamente a que mostra qual o termo da sequência será apresentado na tela. Que variável é esta? #include int main(int argc, char \*argv){ { int anterior=1, atual=1, n, proximo, total=0; printf("digite ate qual termo da serie fibonacci deseja obter"); scanf("%d",&n); printf("od ", anterior); printf("%d ", atual); while (total<=n) { proximo = anterior + atual; anterior = atual; atual = proximo; printf("%d ", proximo); total++; } system("PAUSE"); return 0;}}

★★★ (Eng. de Produção) - [Pseudocódigo] Em uma célula produtiva A temos 450 peças em um dia e na célula B 300 peças/dia. Considerando que o ritmo de produção de peças dia atualmente é de 27 para A e 36 para B, em quantos dias B alcançará A em número de peças? O algoritmo a seguir ilustra essa situação.

```

Pergunta: (Eng. de Produção) - [Pseudocódigo] Em uma célula produtiva A temos 450 peças em um dia e na célula B 300 peças/dia. Considerando que o ritmo de produção de peças dia atualmente é de 27 para A e 36 para B, em quantos dias B alcançará A em número de peças? O algoritmo a seguir ilustra essa situação.

inicio
    inteiro A, B, dias
    A <- 450
    B <- 300
    dias <- 0
    enquanto A >= B faz
        A <- A + 27
        B <- B + 36
        dias <- dias + 1
    fim enquanto
    escrever dias, " dias"
fim

 10 dias
 9 dias
 31 dias
 20 dias
 17 dias

```

**Enviar**

#### **Lista de Exercícios para Estrutura de controle - Laços (repetição)**

★★★ [Pseudocódigo] Comparando as estruturas "enquanto...faz" e "faz...enquanto", é correto afirmar que:

★★★ O programa obtém do usuário o valor n e imprime na tela os n primeiros termos de uma sequência de Fibonacci. Observe que há uma variável faltando, exatamente a que mostra qual o termo da sequência será apresentado na tela. Que variável é esta? #include int main(int argc, char \*argv){ { int anterior=1, atual=1, n, proximo, total=0; printf("digite ate qual termo da serie fibonacci deseja obter"); scanf("%d",&n); printf("od ", anterior); printf("%d ", atual); while (total<=n) { proximo = anterior + atual; anterior = atual; atual = proximo; printf("%d ", proximo); total++; } system("PAUSE"); return 0;}}

★★★ (Eng. Mecânica) - [Pseudocódigo] Ao visitar as obras de uma torre de comunicação, um distraído Engenheiro deixou cair de seu bolso a sua trena. Considerando que a equação de queda livre possui a fórmula  $S = (g * t^2)/2$  e o objeto levou 4 segundos para atingir o solo, a que altura estava esta pessoa? Considere a aceleração da gravidade como  $9,8m/s^2$ . O algoritmo a seguir demonstra essa equação. A partir do teste de mesa, qual será o resultado?

```

Pergunta: (Eng. Mecânica) - [Pseudocódigo] Ao visitar as obras de uma torre de comunicação, um distraído Engenheiro deixou cair de seu bolso a sua trena. Considerando que a equação de queda livre possui a fórmula  $S = (g * t^2)/2$  e o objeto levou 4 segundos para atingir o solo, a que altura estava esta pessoa? Considere a aceleração da gravidade como  $9,8m/s^2$ . O algoritmo a seguir demonstra essa equação. A partir do teste de mesa, qual será o resultado?

inicio
    real s, g, t
    g <- 9,8
    para t de 1 ate 4 passo 1
        s <- ( g * ( t * t ) ) / 2
        proximo
        escrever s, "\n"
    fim

 82,8m
 102,8m
 34,2m
 78,4m
 19,6m

```

**Enviar**

Fonte: Elaborado pelo Autor (2019).

O AdaptWeb agrega também várias métricas que permitem ao professor visualizar o desempenho individual do aluno de um determinado curso e ainda de forma coletiva possibilitando gerar diferentes análises sobre suas disciplinas. A Figura 22 destaca o link à esquerda que dá acesso a opção “Análise Interativa” e no centro da tela é exibido as opções das disciplinas cadastradas disponíveis. Todos os dados a seguir referem-se ao Minicurso de Algoritmos 2019 - ALG2019-01.

Figura 22 – Acesso às análises e relatórios.

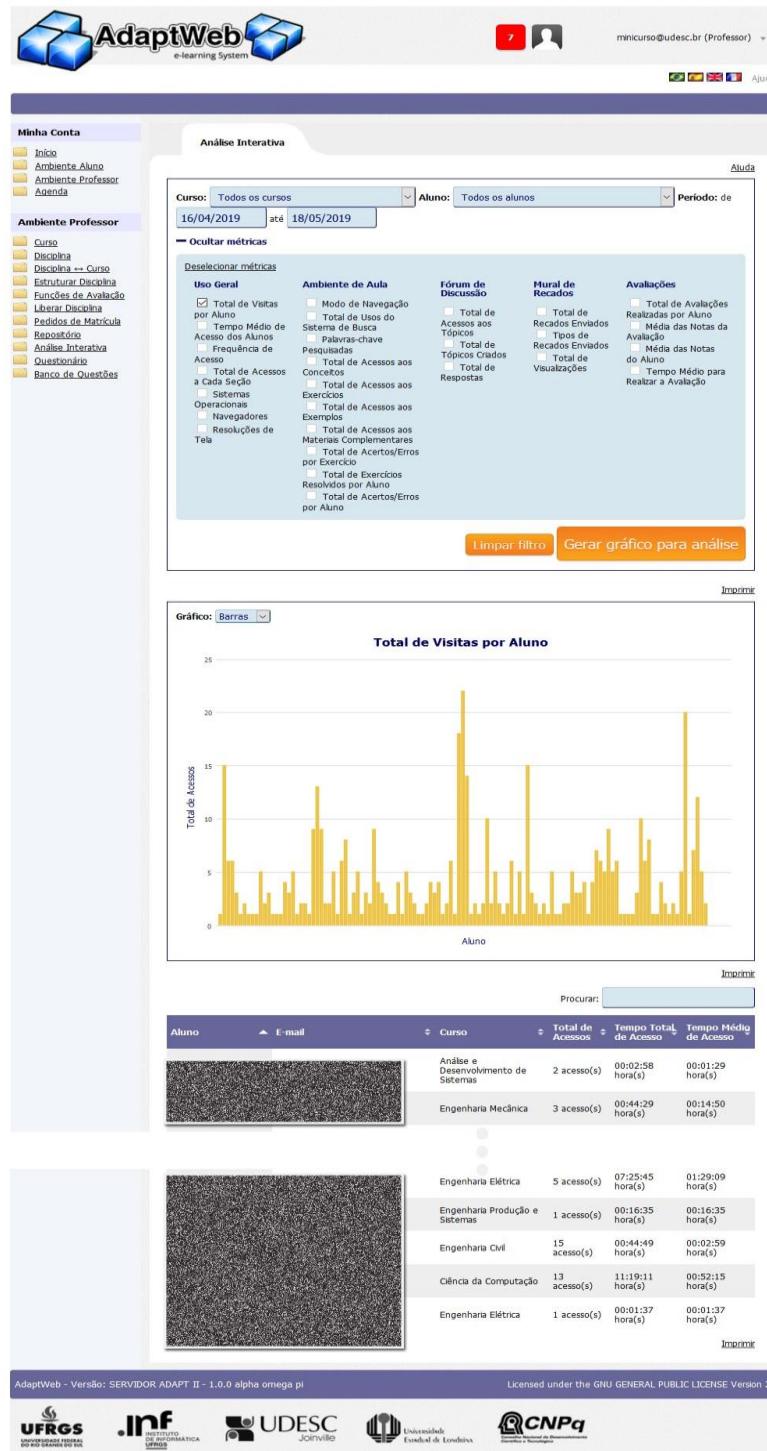
The screenshot shows the AdaptWeb interface. On the left, there's a sidebar with 'Minha Conta' and 'Ambiente Professor' sections. Under 'Ambiente Professor', the 'Análise Interativa' option is highlighted with a red box. The main content area is titled 'Análise Interativa' and lists several disciplines with their status. One discipline, 'Minicurso de Algoritmos 2019-01 - ALG2019-01', is also highlighted with a red box. The footer contains logos for UFRGS, INF, UDESC Joinville, Universidade Estadual de Londrina, and CNPq.

Disciplina	Situação
<a href="#">Minicurso de Algoritmos 2017-1</a>	Liberada
<a href="#">Minicurso de Algoritmos 2018-1</a>	Liberada
<a href="#">Minicurso de Algoritmos 2018-2</a>	Liberada
<a href="#">Minicurso de Algoritmos 2019-01 - ALG2019-01</a>	Liberada

Fonte: Elaborado pelo Autor (2019).

Imediatamente ao click do professor na opção da disciplina escolhida, a janela ilustrada pela Figura 23 irá surgir. Por questões éticas e de privacidade presentes no Termo de Consentimento Livre e Esclarecido acordado com os participantes deste minicurso, seus nomes e e-mails serão preservados. Aplicando-se apenas o filtro de datas (inserindo a data inicial e a data final da disciplina), selecionando assim todos os cursos e alunos obtém-se uma relação extensa. Assim sendo, apenas uma amostra do início e do final da relação de alunos está sendo exibida.

Figura 23 – Tela da Análise Interativa com todos os seus elementos.

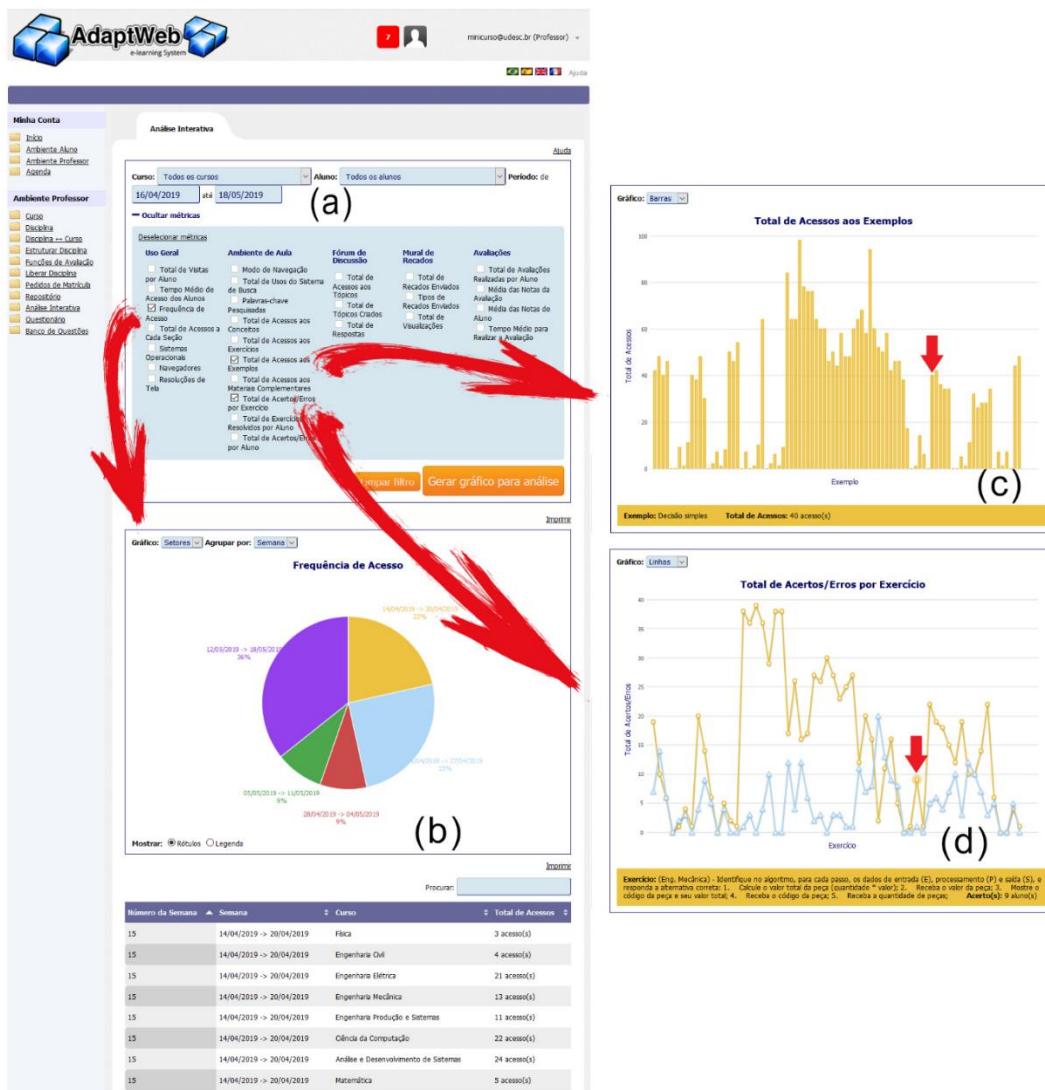


Fonte: Elaborado pelo Autor (2019).

Combinações distintas de gráficos e relatórios sobre os cursos podem ser emitidos pelo AdaptWeb. A Figura 24 ilustra 3 situações. Usando o seguinte filtro para consulta: (a) Curso (Todos os cursos) + Aluno (Todos os alunos) + Período (de 16/04/2019 até 18/05/2019) + a métrica (Frequência de Acesso) obteve-se como resultado o gráfico de setores (b) cujo agrupamento de dados foi por semana

(podendo-se filtrar por dia ou por mês). Se analisado as cores azul claro e amarelo, percebe-se grande número de acessos nas duas primeiras semanas e novamente, na última semana (em roxo) onde havia também a Avaliação Final e o Questionário de Satisfação. Abaixo de cada gráfico há uma listagem em formato de relatório no qual o professor poderá ordenar os dados por colunas. No gráfico de barras (c) é possível visualizar a quantidade de acessos para cada exemplo cadastrado. No detalhe, passa-se o cursor do mouse sobre uma coluna específica e em uma caixa de texto abaixo é exibido a descrição do exemplo cadastrado e a quantidade de acessos. E ilustrando através de gráficos de linhas (d) tem-se uma relação de acertos e erros por exercícios. Novamente com o ponteiro do mouse pode-se verificar em detalhes o enunciado do OA e seus acertos (em amarelo) ou erros (em azul).

Figura 24 – Três diferentes representações gráficas.



Fonte: Elaborado pelo Autor (2019).

Conforme pode-se constatar, o AdaptWeb oferece dezenas de possibilidades de relatórios que permite ao professor fazer diversas análises sob diferentes aspectos (Uso Geral, Ambiente de Aula, Fórum de Discussão, Mural de Recados e Avaliações) facultando-o a emitir pareceres, opiniões e entendimentos sobre a disciplina e aos cursos no qual ela foi ofertada (Figura 25).

Figura 25 – Filtro por Curso.

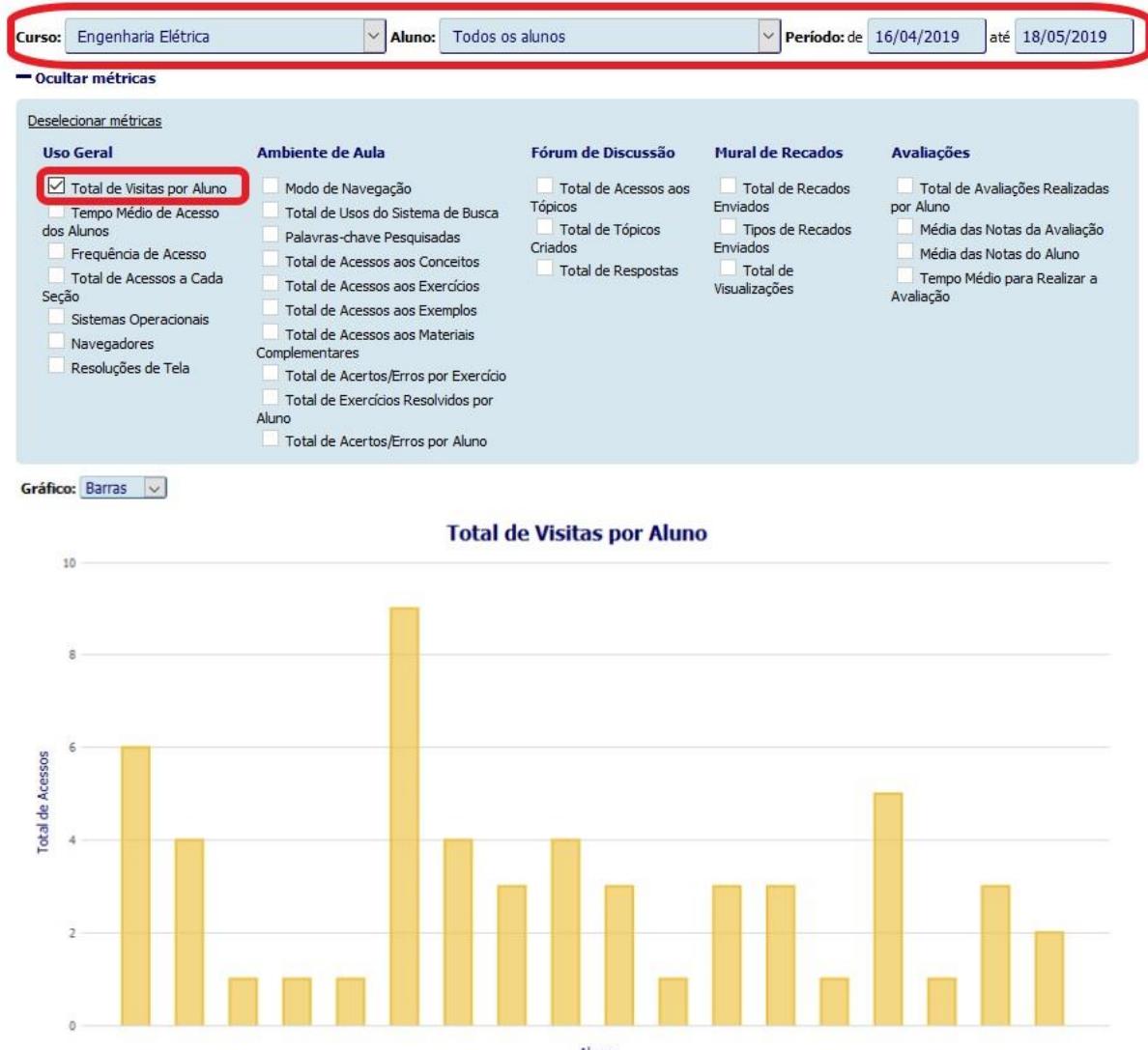
The screenshot shows a user interface for a reporting system. At the top, there is a dropdown menu labeled 'Todos os cursos' which lists various courses: Ciência da Computação, Física, Engenharia Civil, Engenharia Elétrica, Engenharia Mecânica, Engenharia Produção e Sistemas, Análise e Desenvolvimento de Sistemas, and Matemática. Below this, there is a main search form with fields for 'Curso' (set to 'Todos os cursos'), 'Aluno' (set to 'Todos os alunos'), and 'Período: de 16/04/2019 até 18/05/2019'. There is also a checkbox for 'Ocultar métricas' and a link 'Deseleccionar métricas'. The main area displays five categories of metrics: 'Uso Geral', 'Ambiente de Aula', 'Fórum de Discussão', 'Mural de Recados', and 'Avaliações'. Under each category, there is a list of checkboxes representing different metrics. At the bottom right of the interface are two buttons: 'Limpar filtro' and 'Gerar gráfico para análise'.

Fonte: Elaborado pelo Autor (2019).

Para efeitos de acompanhamento e análise pelo professor, os relatórios podem ser customizados por curso, aluno, por período combinados a uma das 27 métricas disponíveis. Para ilustrar algumas dessas capacidades, na Figura 26 observa-se a partir do filtro (Curso: Engenharia Elétrica + Aluno: Todos os alunos + Período de 16/04/2019 até 18/05/2019) + a métrica Total de Visitas por Aluno, a quantidade de acessos por aluno desse curso. Fica como dica passar com o cursor do mouse sobre

qualquer uma das barras do gráfico para que seja exibido o nome desse aluno com a quantidade de seus acessos.

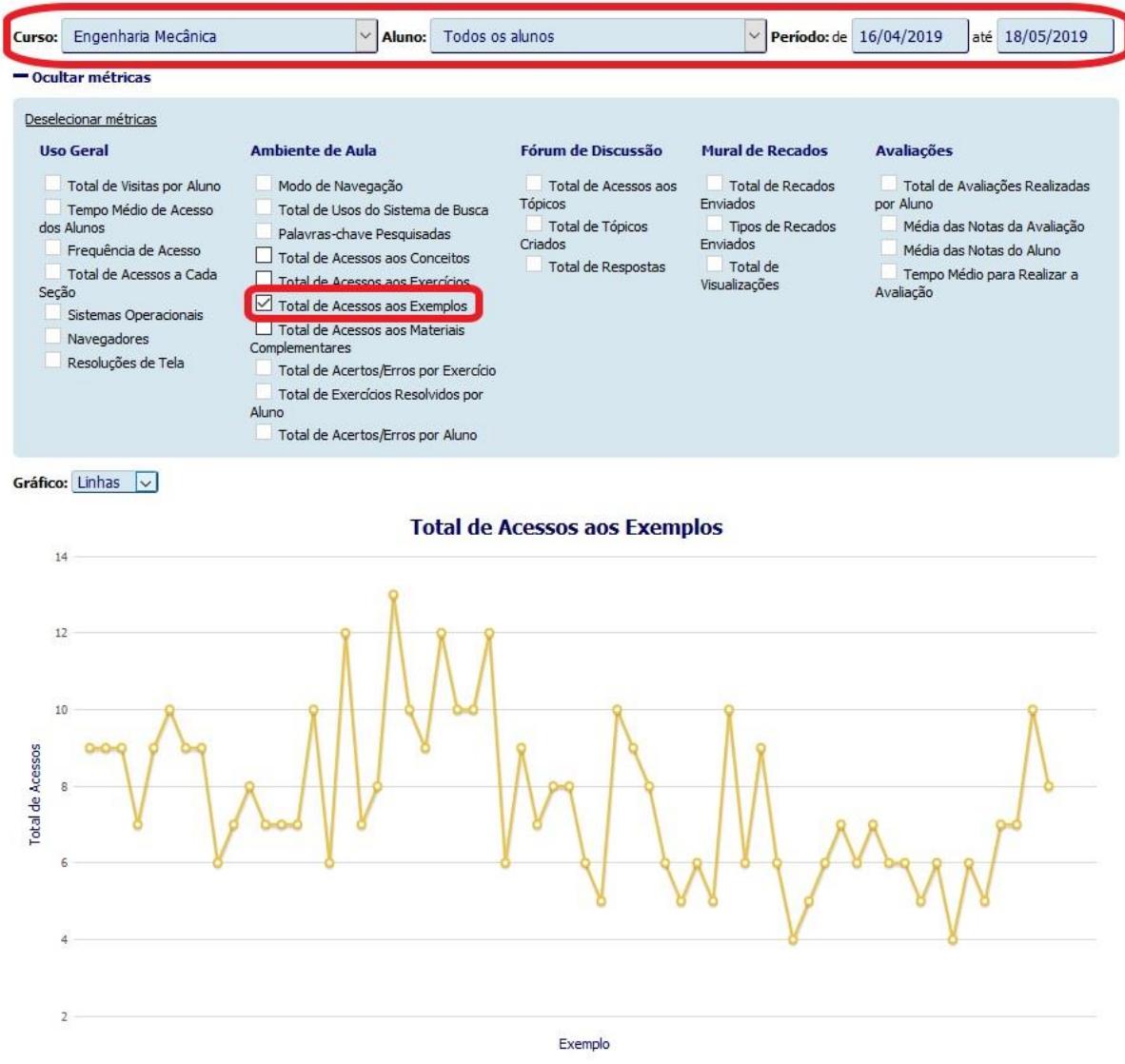
Figura 26 – Gráfico de barras customizado: EEL.



Fonte: Elaborado pelo Autor (2019).

Verificando outro curso de Engenharia, a EMC, foi utilizado o filtro (Curso: Engenharia Mecânica + Aluno: Todos os alunos + Período de 16/04/2019 até 18/05/2019) + a métrica Total de Acessos aos Exemplos. O formato da representação visual é o gráfico de linhas conforme a Figura 27.

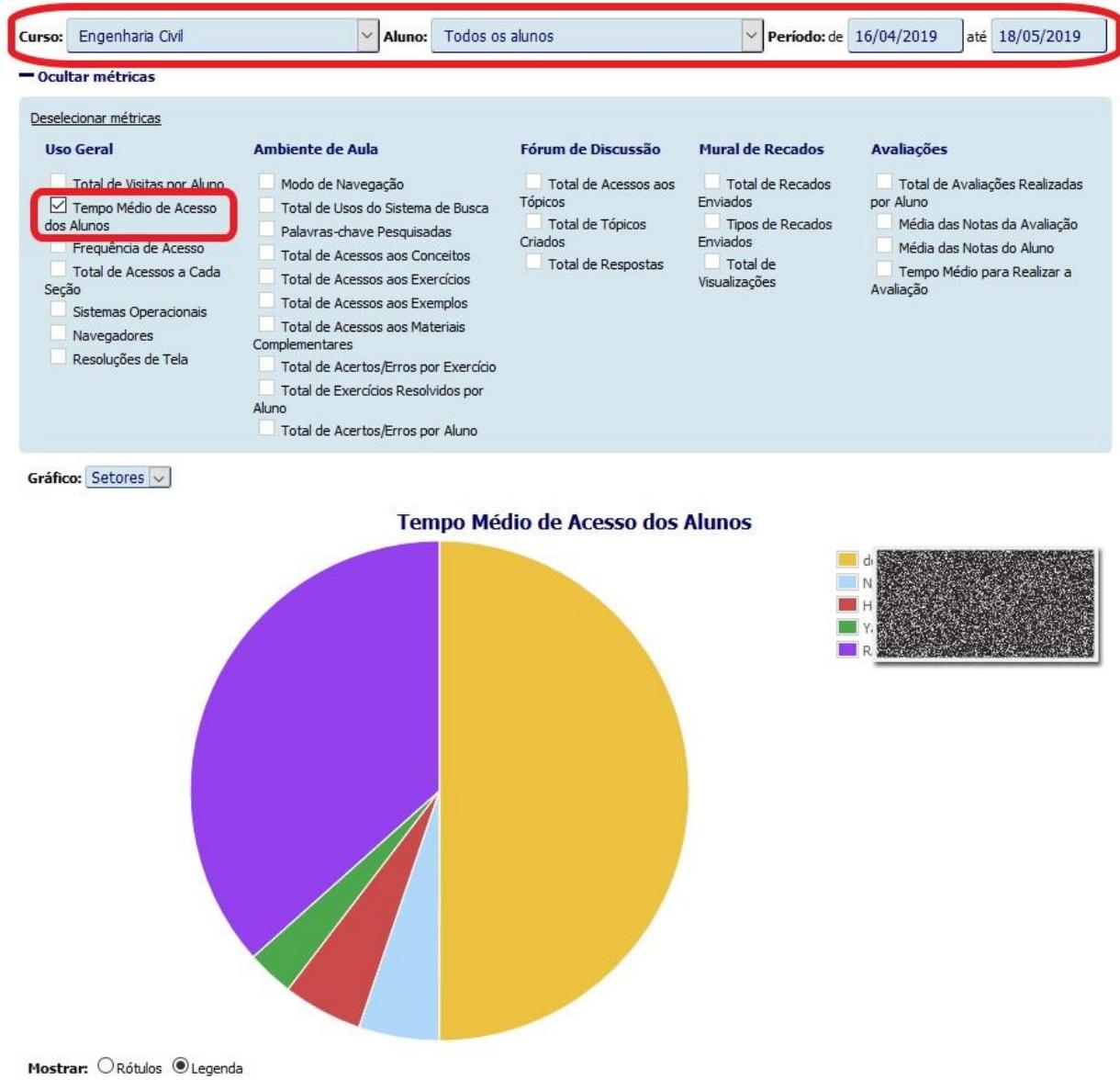
Figura 27 – Gráfico de linhas customizado: EMC.



Fonte: Elaborado pelo Autor (2019).

Diversos outras análises podem ser realizadas a partir da ferramenta Análise Interativa. Finalizando este estudo a representação do Tempo Médio de Acessos dos alunos da turma de Engenharia Civil é exibido na Figura 28 com uma tarja nos nomes dos alunos que estão sob garantia de anonimato.

Figura 28 – Gráfico de setores customizado: ECV.



Fonte: Elaborado pelo Autor (2019).

## 5 CONSIDERAÇÕES FINAIS

A disciplina de ALP há muito já é reconhecida como um divisor de águas nos cursos relacionados às áreas de Computação e Sistemas. Se as dificuldades estão presentes para estas turmas, que a possuem como disciplina-base para todo do curso, para os demais cursos como as Licenciaturas, Engenharias e outros, é considerada pelos alunos como um saber periférico, dispensável, difícil e gerador de inúmeras insatisfações.

O trabalho desenvolvido utilizou como metodologia a criação personalizada de OA's para situações e contextos presentes nas disciplinas de ALP do CCT-Udesc. A fim de validar a vivência em sala de aula do autor deste produto educacional, o AdaptWeb surgiu como o meio para coletar dados e verificar tal experiência.

Os objetivos propostos foram alcançados através da dissertação que faz parte deste estudo. A elaboração deste **Guia para a criação de Objetos de Aprendizagem personalizados no ambiente AdaptWeb** possibilitará que outros professores amparados por este trabalho possam criar suas disciplinas associadas ao seu curso com OA's personalizados de acordo com as suas necessidades. Os alunos participantes do Minicurso de Algoritmos 2019 - ALG2019-01 contribuíram positivamente corroborando que exemplos e exercícios personalizados para o seu curso de formação auxiliam na compreensão dos conteúdos, agregam novos conhecimentos e facilita o entendimento da disciplina de ALP. Tal complemento reforça a manutenção dos alunos em sala de aula, diminuindo evasões e desistências bastante comuns na disciplina. Para que o professor possa visualmente entender o processo de construção de OA's personalizados com base neste guia, um vídeo instrucional detalha as etapas: <[https://www.youtube.com/watch?v=5si3\\_LTZLA8](https://www.youtube.com/watch?v=5si3_LTZLA8)>.

As limitações observadas surgem na forma de temas para trabalhos futuros como comparar o desempenho de alunos participantes em um próximo experimento com/sem personalizações, a verificação de desempenho de alunos da modalidade presencial de ALP em grupos de participantes/não participantes do minicurso personalizado. Como solicitação dos próprios alunos pode-se aumentar a quantidade de exercícios e exemplos personalizados para as próximas edições.

## REFERÊNCIAS

ATANASIO, V. PEREIRA, F. O. R.; PEREIRA, A. T. C. Laboratório experimental para ensino de iluminação em arquitetura através de um AVA. In: Encontro Nacional de Tecnologia do Ambiente Construído, 2016, Florianópolis. XI Encontro Nacional de Tecnologia no Ambiente Construído, Florianópolis: ENTAC, 2016. Disponível em: <[https://www.researchgate.net/publication/279537148\\_LABORATORIO\\_EXPERIMENTAL\\_PARA\\_ENSINO\\_DE\\_ILUMINACAO\\_EM\\_ARQUITETURA\\_ATRAVES\\_DE\\_UM\\_AVA](https://www.researchgate.net/publication/279537148_LABORATORIO_EXPERIMENTAL_PARA_ENSINO_DE_ILUMINACAO_EM_ARQUITETURA_ATRAVES_DE_UM_AVA)>. Acesso em: 8 ago. 2017.

BRUSILOVSKY, P. **Adaptive Hypermedia**. User Modeling and User Adapted Interaction. University of Pittsburgh, Pittsburgh, 2001.

DOS SANTOS, L. V. **O uso da metodologia ADDIE no Design Instrucional de um minicurso online de algoritmos e programação para o ensino superior**. Joinville. Dissertação (Mestrado) – Centro de Ciências e Tecnologias, UDESC, 2017.

GASPARINI, I. **Interface adaptativa no ambiente AdaptWeb**: navegação e apresentação adaptativa baseada no modelo do usuário. Porto Alegre. Dissertação (Mestrado) - Instituto de Informática, UFRGS, 2003.

IEEE-LTSC P1484.12, IEEE Learning Technology Standard Committee (LTSC). **Learning Object Metadata Working Document**. [S.I.]: Institute of Eletrical and Electronics Engineers, 2000. Disponível em: <[https://standards.ieee.org/standard/1484\\_12\\_1-2002.html](https://standards.ieee.org/standard/1484_12_1-2002.html)> acesso em 25 out. 2018.

KOCH, N. P. **Software Engineering for Adaptive Hypermedia Systems**. Tese (PhD). Universitat München, Alemanha, 2000.

L'ALLIER, J. **Frame of Reference: NETg's Map to the Products, Their Structure and Core Beliefs**. NETg. 1997.

PALAZZO, L. A. M. **Sistemas de Hipermídia Adaptativa**, In: Anais do XXII Congresso da Sociedade Brasileira de Computação (SBC), Florianópolis, 2002.

## APÊNDICE A: OBJETOS DE APRENDIZAGEM PERSONALIZADOS

Relação completa dos novos OA's criados para o Minicurso de Algoritmos 2019

- ALG2019-01 separados individualmente em Exemplos e Exercícios e por arquivos.

Conceito
8 Dados, variáveis e operadores
<p>Exercícios Genéricos Arquivo: GN EXER1.TXT</p> <p>Na elaboração de um algoritmo para a resolução da equação de <math>X^2 + 3y - Z = 0</math>, relate as colunas:</p> <ol style="list-style-type: none"> <li>1. Dados de entrada    ( ) Resultado &lt;- (<math>X*X + 3*Y - Z</math>)</li> <li>2. Processamento    ( ) Escreva (Resultado)</li> <li>3. Saída              ( ) Leia (X, Y, Z)</li> </ol> <p>( ) 2, 3, 1 ( ) 3, 1, 2 ( ) 1, 2, 3 ( ) 3, 2, 1 ( ) 2, 1, 3</p>

<p>Exercícios Civil Arquivo: EC EXER1.TXT</p> <p>Identifique no algoritmo, para cada passo, os dados de entrada (E), processamento (P) e saída (S), e responda a alternativa correta:</p> <ol style="list-style-type: none"> <li>1. Receba o código do bloco cerâmico;</li> <li>2. Receba o valor do bloco cerâmico;</li> <li>3. Receba a quantidade de blocos;</li> <li>4. Calcule o valor total (quantidade * valor);</li> <li>5. Mostre o código do bloco cerâmico e o valor total orçado.</li> </ol> <p>( ) 1-E, 2-E, 3-E, 4-P, 5-S. ( ) 1-S, 2-S, 3-S, 4-S, 5-E. ( ) 1-P, 2-P, 3-P, 4-P, 5-S. ( ) 1-E, 2-E, 3-E, 4-S, 5-P. ( ) 1-P, 2-P, 3-P, 4-E, 5-S.</p>
--

<p>Exercícios Elétrica Arquivo: EL EXER1.TXT</p> <p>Identifique no algoritmo, para cada passo, os dados de entrada (E), processamento (P) e saída (S), e responda a alternativa correta:</p> <ol style="list-style-type: none"> <li>1. Receba o código do componente;</li> <li>2. Receba o valor do componente;</li> <li>3. Receba a quantidade do componentes;</li> <li>4. Calcule o valor total (quantidade * valor);</li> <li>5. Mostre o código do componente e seu valor total.</li> </ol> <p>( ) 1-E, 2-E, 3-E, 4-P, 5-S. ( ) 1-S, 2-S, 3-S, 4-S, 5-E. ( ) 1-P, 2-P, 3-P, 4-P, 5-S. ( ) 1-E, 2-E, 3-E, 4-S, 5-P. ( ) 1-P, 2-P, 3-P, 4-E, 5-S.</p>
---

**Exercícios Mecânica**  
Arquivo: EM EXER1.TXT

Identifique no algoritmo, para cada passo, os dados de entrada (E), processamento (P) e saída (S), e responda a alternativa correta:

1. Receba o código da peça;
2. Receba o valor da peça;
3. Receba a quantidade de peças;
4. Calcule o valor total da peça (quantidade \* valor);
5. Mostre o código da peça e seu valor total.

- ( ) 1-E, 2-E, 3-E, 4-P, 5-S.  
 ( ) 1-S, 2-S, 3-S, 4-S, 5-E.  
 ( ) 1-P, 2-P, 3-P, 4-P, 5-S.  
 ( ) 1-E, 2-E, 3-E, 4-S, 5-P.  
 ( ) 1-P, 2-P, 3-P, 4-E, 5-S.

**Exercícios Produção**  
Arquivo: EP EXER1.TXT

Identifique no algoritmo, para cada passo, os dados de entrada (E), processamento (P) e saída (S), e responda a alternativa correta:

1. Receba o código do insumo;
2. Receba o valor do insumo;
3. Receba a quantidade de insumos;
4. Calcule o valor total do item (quantidade \* valor do insumo);
5. Mostre o código do insumo e seu valor total.

- ( ) 1-E, 2-E, 3-E, 4-P, 5-S.  
 ( ) 1-S, 2-S, 3-S, 4-S, 5-E.  
 ( ) 1-P, 2-P, 3-P, 4-P, 5-S.  
 ( ) 1-E, 2-E, 3-E, 4-S, 5-P.  
 ( ) 1-P, 2-P, 3-P, 4-E, 5-S.

## 8.2 Operadores básicos

**Exemplos Genéricos**  
Arquivo: GN EXEM1.TXT

Para a elaboração de um composto específico na indústria, um engenheiro utiliza uma determinada quantidade de insumos. Faça um algoritmo que receba a quantidade do composto utilizado diariamente (fixo) com unidade de medida em gramas. Calcule e escreva a quantidade deste composto que serão gerados (em Kg) após uma semana.

```
inicio
  real qtde_gramas, total_gramas
  ler qtde_gramas
  total_gramas <- qtde_gramas * ( 7 / 1000 )
  escrever total_gramas
fim
```

**Exemplos CIVIL**  
Arquivo: EC EXEM1.TXT

Desenvolva um algoritmo para ler as dimensões de um ambiente destinado a um laboratório retangular qualquer (comprimento, largura e altura). Calcular e escrever a quantidade de caixas de azulejos para se colocar em todas as suas paredes (considere que não será descontada a área ocupada por portas e janelas). Cada caixa de azulejos preenche 2,3 m<sup>2</sup>.

```

inicio
    real larg, comp, alt, qtde_caixa

    ler larg, comp, alt
    qtde_caixa <- ( ( larg * alt ) * 2 ) + ( ( comp * alt ) * 2 ) / 2.3
    escrever qtde_caixa
fim

```

#### Exemplos ELÉTRICA

Arquivo: EL EXEM1.TXT

Escreva um programa para calcular e imprimir o número de lâmpadas necessárias para iluminar um determinado cômodo de uma residência. Dados de entrada: a potência da lâmpada utilizada (em watts), as dimensões (largura e comprimento, em metros) do cômodo. Considere que a potência necessária é de 18 watts por metro quadrado.

```

inicio
    inteiro pot_lamp
    real larg, comp, qtde_lamp

    ler pot_lamp
    ler larg, comp
    qtde_lamp <- ( ( larg * comp ) * 18 ) / pot_lamp
    escrever qtde_lamp
fim

```

#### Exemplos MECÂNICA

Arquivo: EM EXEM1.TXT

Elabore um algoritmo que aborde a lei da conservação de energia. Desenvolva um código que leia as variáveis referentes para o cálculo e, ao final, calcule e escreva a energia mecânica, de acordo com a fórmula do princípio da conservação de energia mecânica.

Utilize a fórmula:  
 $E = U + K$  (Energia mecânica = Energia potencial + Energia cinética).

```

inicio
    real energia_mec, u, k

    ler u, k
    energia_mec <- u + k
    escrever energia_mec
fim

```

#### Exemplos PRODUÇÃO

Arquivo: EP EXEM1.TXT

Desenvolva um algoritmo que calcule o comprimento de onda. Cabe ao usuário informar a velocidade desta onda (em m/s) e sua frequência em Hz.

Fórmula: Compr = v/f

```

inicio
    real v, f

    ler v, f
    escrever "Comprimento da onda = ", v / f, "m"
fim

```

## 9 Estruturas de controle: Decisão

Exercícios Genéricos  
Arquivo: GN EXER2.TXT

Qual será o resultado a ser escrito na tela após a execução do algoritmo abaixo? Realize um teste de mesa.

```

inicio
    inteiro A, B, C
    texto D

    A <- 2
    B <- -10
    C <- 5
    D <- "UDESC"
    se ( A > B ) e ( C >= A ) entao
        escrever "Minicurso"
    fimse
    se ( D == "Joinville" ) ou ( C <= B ) entao
        escrever "Algoritmos"
    senao
        escrever "CCT"
    fimse
    se ( C >= A ) e ( C == -5 ) e ( D == "UDESC" ) entao
        se ( B * -1 ) = 10 entao
            escrever "UDESC"
        senao
            escrever "PPGECMT & PPGCA"
        fimse
    fimse
fim

```

- a) Minicurso Algoritmos UDESC
- b) Minicurso CCT UDESC
- c) Algoritmos CCT UDESC
- d) Minicurso CCT PPGECMT & PPGCA
- e) Algoritmos CCT UDESC

Exercícios Civil  
Arquivo: EC EXER2.TXT

Analisando as linhas do algoritmo abaixo, qual será o resultado da execução deste trecho de programa?

```

inicio
    inteiro a,b,c
    a <- 5
    b <- 3
    c <- 3
    se ( a > 3 ) e ( b >= c ) entao
        escrever "ENGENHARIA"
    fimse
    se ( c == a ) entao
        escrever "CIVIL"
    fimse
fim

```

- a) ENGENHARIA CIVIL
- b) ENGENHARIA
- c) Não escreverá nada
- d) CIVIL
- e) CIVIL ENGENHARIA

**Exercícios Elétrica**  
Arquivo: EL EXER2.TXT

Watt É uma medida de potência, que é calculada multiplicando a tensão pela corrente. Com 12 volts e 30 amperes temos 360 watts, por exemplo.

Quantos Watts obteremos através das linhas de código abaixo?

```

inicio
    inteiro volt, amp, watt

    volt <- 9
    amp <- 2
    se volt <= 0 ou amp <= 0 entao
        escrever "Valor inválido informado"
    senao
        watt <- volt * amp
    fimse
    escrever watt
fim

a) 4.5w
b) 18w
c) 9w
d) 180w
e) 2 w

```

**Exercícios Mecânica**  
Arquivo: EM EXER2.TXT

Faça um algoritmo que, dada uma força F, aplicada a um bloco de 15kg que desliza sobre um superfície onde o coeficiente de atrito dinâmico é 0,25. O corpo tem aceleração constante de 1m/s<sup>2</sup>.

Qual a força aplicada no corpo?

Nota: Considere a aceleração da gravidade = 10 m/s<sup>2</sup> na terra e 1.26 m/s<sup>2</sup> na lua

Fórmula:  $F - u \cdot m \cdot g = m \cdot a$

Onde:

F = Força

u = coeficiente de atrito

g = aceleração da gravidade

m = massa do corpo

a = aceleração do corpo

inicio

real F, u, m, g, a

u <- 0.25

m <- 15

g <- 10

a <- 1

se ( g = 10 ) entao

escrever "Força na gravidade terrestre \n"

F <- u \* m \* g + m \* a

fimse

se ( g = 1.26 ) entao

escrever "Força na gravidade na lua \n"

F <- u \* m \* g + m \* a

fimse

escrever F, " N"

fim

- a) Força na gravidade terrestre 52.5 N
- b) Força ba gravidade na lua 19.725 N
- c) 52.5 N
- d) 19.725 N
- e) 3.750 N

#### Exercícios Produção

Arquivo: EP EXER2.TXT

Analizando as linhas do algoritmo abaixo e considerando as informações como sendo os três lados de um triângulo, qual será o resultado da execução deste trecho de programa?

```
inicio
    inteiro a,b,c
```

```
    a <- 7
    b <- 6
    c <- 5
    se ( a = b ) e ( a = c ) entao
        escrever "equilátero"
    fimse
    se ( a /= b ) e ( a /= c ) e ( b /= c ) entao
        escrever "escaleno"
    fimse
fim
```

- a) escaleno
- b) equilátero escaleno
- c) Não escreverá nada
- d) 7 6 5
- e) 5 6 7

#### 9.1 Estrutura de decisão simples

##### Exemplos Genéricos

Arquivo: GN EXEM2.TXT

Desenvolva um algoritmo que leia um valor em Megabytes e converta em Kilobytes ou Gigabytes conforme a opção do usuário.

```
inicio
    inteiro mb, opcao

    escrever "Digite o valor em Megabytes "
    ler mb
    escrever "Informe a unidade para qual deseja converter: 1) Kb, 2) Gb "
    ler opcao
    escrever "Conversão = "
    se opcao = 1 entao
        escrever mb * 1024
    fimse
    se opcao = 2 entao
        escrever mb / 1024
    fimse
fim
```

##### Exemplos Civil

Arquivo: EC EXEM2.TXT

O saco de cimento de 50kg custa R\$21,50 se comprados menos de 10 unidades e R\$20,00 comprando-se 10 ou mais. Escreva um Algoritmo que leia a quantidade de sacos comprados, calcule e escreva o valor total da compra.

```

inicio
    inteiro qtde

    ler qtde
    se qtde < 10 entao
        escrever qtde * 21.50
    senao
        escrever qtde * 20
    fimse
fim

```

**Exemplos Elétrica**  
Arquivo: EL EXEM2.TXT

Lâmpadas de led de 9w custam R\$ 9,70 cada se forem compradas menos de uma dúzia, e R\$ 9,00 se forem compradas em quantidade de 12 unidades ou mais. Escreva um Algoritmo que leia o número de lâmpadas compradas, calcule e escreva o custo total da compra.

```

inicio
    inteiro qtde

    ler qtde
    se qtde < 12 entao
        escrever qtde * 9.70
    senao
        escrever qtde * 9.00
    fimse
fim

```

**Exemplos Mecânica**  
Arquivo: EM EXEM2.TXT

Implemente um algoritmo que leia um valor referente a coleta e aferição do PH de uma fonte de água potável. Ao final, escreva "PH alcalino" caso o valor lido for igual ou superior a 7 e "PH ácido" se o valor lido for inferior a 7.

```

inicio
    real ph

    ler ph
    se ph >= 7 entao
        escrever "PH alcalino"
    senao
        escrever "PH ácido"
    fimse
fim

```

**Exemplos Produção**  
Arquivo: EP EXEM2.TXT

Escreva um algoritmo que leia o valor de uma temperatura em Celsius em uma câmara fria e escreva o valor lido é positivo ou negativo. Considere o valor zero como positivo.

```

inicio

```

```

real temp

ler temp
se temp >= 0 entao
    escrever "Temperatura positiva"
senao
    escrever "Temperatura negativa"
fimse
fim

```

## 9.2 Estrutura de decisão composta

Exemplos Genéricos  
Arquivo: GN EXEM3.TXT

Elabore um programa que calcule o teste de hipóteses estatístico. O usuário deve informar a hipótese nula e a alternativa, o nível de significância desejado, as médias amostral e populacional, o desvio padrão populacional e o número de indivíduos que contém a amostra. O teste de hipóteses deve considerar a tabela de distribuição normal, ou seja, a tabela Z. O usuário deve fornecer o valor da tabela ao programa para que sejam feitos os cálculos. O programa deve testar se foi aceita a hipótese nula ou a alternativa. Após a realização do cálculo, o programa deve apresentar qual das duas alternativas foi aceita.

```

inicio
    real nivel_significancia, H0, H1, P, z_calc, media_amostra, media_pop, desvio_padrao
    inteiro n
    escrever "Digite a hipótese nula, H0: "
    ler H0
    escrever "Digite a hipótese alternativa, H1 (qualquer valor diferente de H0): "
    ler H1
    escrever "Digite o nível de significância a ser considerado(em decimais): "
    ler nivel_significancia
    escrever "Digite a média da amostra e a média da população, respectivamente!"
    ler media_amostra, media_pop
    escrever "Digite o desvio padrão pertencente à população:"
    ler desvio_padrao
    escrever "Digite o número de indivíduos pertencentes à amostra: "
    ler n
    z_calc <- ( raiz ( n ) * ( media_amostra - media_pop ) ) / desvio_padrao
    escrever "Informe o valor crítico associado a: ", z_calc
    ler P
    se ( z_calc < 0 ) e ( P > nivel_significancia ) ou ( z_calc > 0 ) e ( P < nivel_significancia ) entao
        escrever "Aceitamos a hipótese nula H0 e rejeitamos a hipótese alternativa H1.\n"
    senao
        escrever "Aceitamos a hipótese alternativa H1 e rejeitamos a hipótese nula H0.\n\n"
    fimse
fim

```

Exemplos Civil  
Arquivo: EC EXEM3.TXT

Um tijolo de 6 furos (residencial) mede respectivamente 9x14x19cm. Já o tijolo de 8 furos (predial) mede 9x19x19cm. Faça um algoritmo que solicite ao usuário o tipo do tijolo (R ou P) em pé a ser usado e o tamanho da parede (em m<sup>2</sup>) a ser feita. Ao final, escreva a quantidade de tijolos necessários para a obra.

```

inicio
    real metragem, qtde
    caracter tipo

```

```

ler tipo
ler metragem
se tipo = "R" entao
    qtde <- metragem / ( ( 14 * 19 ) / 1000 )
fimse
se tipo = "P" entao
    qtde <- metragem / ( ( 19 * 19 ) / 1000 )
fimse
escrever qtde
fim

```

#### Exemplos Elétrica

Arquivo: EL EXEM3.TXT

Desenvolva um algoritmo que leia os 3 lados de um triângulo e escreva se este triângulo é Equilátero (3 lados iguais) ou Escaleno (3 lados diferentes).

```

inicio
    real l1, l2, l3

    ler l1, l2, l3
    se ( l1 = l2 ) e ( l2 = l3 ) entao
        escrever "Equilátero"
    fimse
    se (l1 != l2 ) e ( l2 != l3 ) e ( l1 != l3 ) entao
        escrever "Escaleno"
    fimse
fim

```

#### Exemplos Mecânica

Arquivo: EM EXEM3.TXT

Uma esteira de transporte de produtos possui configurações de velocidade que variam em:

- até 0,5 m/s = normal
- acima de 0,6m/s = rápido

Faça um algoritmo que leia o valor correspondente a velocidade desta esteira e escreva a sua classificação

```

inicio
    real velocidade

    ler velocidade
    se velocidade <= 0.5 entao
        escrever "Velocidade normal"
    senao
        escrever "Velocidade rápida"
    fimse
fim

```

#### Exemplos Produção

Arquivo: EP EXEM3.TXT

Andréia possui uma consultoria em Engenharia da Qualidade. Sua empresa irá aumentar o salário dos empregados, porém não para todos. Escreva um algoritmo que leia o salário de um empregado e mostre o seu salário reajustado a partir das seguintes regras:

- Funcionários que recebem valor inferior a R\$ 5.000,00 obterão um reajuste de 10% no salário;
- Funcionários com holerite entre R\$ 5.000,00 e R\$ 20.000,00 terão um reajuste de 5%;

- Funcionários com salário superior a R\$ 20.000,00 não terão direito ao aumento.

```

inicio
    real salario, reajuste

    ler salario
    se salario < 5000 entao
        reajuste <- salario + ( salario * ( 10/100 ) )
    fimse
    se salario >= 5000 e salario < 20000 entao
        reajuste <- salario + ( salario * ( 5/100 ) )
    fimse
    se salario >= 20000 entao
        reajuste <- salario
    fimse
    escrever reajuste
fim

```

## 10 Estrutura de controle: Laços (repetição)

Exercícios Genéricos

Arquivo: GN EXER3.TXT

Para implementar um algoritmo que calcule e escreva a tabuada do número 2, quais seriam os valores correspondentes para as variáveis multiplicando, multiplicador e t no código fonte abaixo?

```

inicio
    inteiro multiplicando, multiplicador, t

    multiplicando <- ?
    multiplicador <- ?
    para t de ? ate multiplicador passo 1
        escrever multiplicando * t, "\n"
    proximo
fim

```

- a) multiplicando <- 2, multiplicador <- 10, t = 1
- b) multiplicando <- 1, multiplicador <- 10, t = 1
- c) multiplicando <- 2, multiplicador <- 10, t = 2
- d) multiplicando <- 2, multiplicador <- 1, t = 10
- e) multiplicando <- 2, multiplicador <- 10, t = 10

Exercícios Civil

Arquivo: EC EXER3.TXT

(Adaptado do conto dos grãos de cereal em um tabuleiro de xadrez)

Um pedreiro erudito, logo no início da obra de um edifício de 40 andares, comentou com o Engenheiro responsável que, dado o tamanho do empreendimento e os longos anos de serviço até a entrega final do imóvel, trabalharia "de graça" em troca de alguns tijolos para a reforma de sua humilde casa. O esperto Engenheiro logo achou interessante pois a estimativa de trabalho era de não menos do que 4 anos portanto, acreditou estar em vantagem em relação ao seu trabalhador e aceitou a proposta.

Tal proposta consistia em pegar um tabuleiro de xadrez comum e colocar um tijolo na primeira casa, dois na segunda, quatro na terceira e assim por diante, ir dobrando a quantidade de tijolos até a última casa (64<sup>a</sup>). No final, este seria a quantidade de tijolos que o trabalhador receberia em troca do seu trabalho.

Dado o algoritmo que representa a passagem responda quantos tijolos o trabalhador receberá na 8<sup>a</sup>, 16<sup>a</sup> e 64<sup>a</sup> casas respectivamente?

```

inicio
    real tijolos

```

```

inteiro i
tijolos <- 1
para i de 1 ate 64 passo 1
    tijolos <-(tijolos*2)
    se i = 8 entao
        escrever tijolos, "\n"
    fimse
    se i = 16 entao
        escrever tijolos, "\n"
    fimse
    se i = 64 entao
        escrever tijolos, "\n"
    fimse
proximo
fim

```

- a) 8, 16, 64
- b) 64, 128, 256
- c) 256, 65536, 1.8446744073709552E19
- d) 0
- e) 256, 1024, 65536

#### Exercícios Elétrica

Arquivo: EL EXER3.TXT

De acordo com o código fonte abaixo, o valor a ser escrito pelo algoritmo será:

```

inicio
    inteiro vetor [ 10 ]
    inteiro cont, num, x

    num <- 5
    para cont de 1 ate 10 passo 1
        vetor [ cont ] <- vetor [ cont ] * num
    proximo
    escrever vetor [ 4 ]
fim

```

- a) 4
- b) 10
- c) 20
- d) 40
- e) 5

#### Exercícios Mecânica

Arquivo: EM EXER3.TXT

Faça um algoritmo que, dada uma força F, aplicada a um bloco de 15kg que desliza sobre um superfície onde o coeficiente de atrito dinâmico é 0,25. O corpo tem aceleração constante de 1m/s<sup>2</sup>.

Qual a força aplicada no corpo?

Nota: Considere a aceleração da gravidade = 10 m/s<sup>2</sup> na terra e 1.26 m/s<sup>2</sup> na lua

Fórmula:  $F - u \cdot m \cdot g = m \cdot a$

Onde:

F = Força

u = coeficiente de atrito

g = aceleração da gravidade

m = massa do corpo

```

a = aceleração do corpo

inicio
    real F, u, m, g, a

    u <- 0.25
    m <- 15
    g <- 10
    a <- 1
    se ( g = 10 ) entao
        escrever "Força na gravidade terrestre \n"
        F <- u * m * g + m * a
    fimse
    se ( g = 1.26 ) entao
        escrever "Força na gravidade na lua \n"
        F <- u * m * g + m * a
    fimse
    escrever F, " N"
fim

```

- a) Força na gravidade terrestre 52.5 N
- b) Força ba gravidade na lua 19.725 N
- c) 52.5 N
- d) 19.725 N
- e) 3.750 N

Exemplo Produção  
Arquivo: EP EXER3.TXT

Analizando as linhas do algoritmo abaixo e considerando as informações como sendo os três lados de um triângulo, qual será o resultado da execução deste trecho de programa?

```

inicio
    inteiro a,b,c

    a <- 7
    b <- 6
    c <- 5
    se ( a = b ) e ( a = c ) entao
        escrever "equilátero "
    fimse
    se ( a != b ) e ( a != c ) e ( b != c ) entao
        escrever "escaleno "
    fimse
fim

a)      escaleno
b)      equilátero escaleno
c)      Não escreverá nada
d)      7 6 5
e)      5 6 7

```

### 10.1 Laço while (enquanto)

Exemplos Genérico  
Arquivo: EG EXEM4.TXT

Em ano eleitoral os alunos da UDESC foram solicitados para realizarem uma pesquisa de intenção de votos para os candidatos A, B e C. Leia os votos para os candidatos até que seja digitado X como candidato. Ao final, calcule e escreva o percentual que de votos que cada candidato recebeu.

```

inicio
    inteiro total, totA, totB, totC
    caracter A, B, C, opcao

    escrever "Escolha o seu candidato: A, B, C ou X para sair... \n"
    ler opcao
    enquanto ( opcao = "A" ) ou ( opcao = "B" ) ou ( opcao = "C" ) faz
        total <- total + 1
        se ( opcao = "A" ) entao
            totA <- totA + 1
        fimse
        se ( opcao = "B" ) entao
            totB <- totB + 1
        fimse
        AB  se ( opcao = "C" ) entao
            totC <- totC + 1
        fimse
        ler opcao
    fim enquanto
    escrever ( totA * 100 ) / total, "% A\n"
    escrever ( totB * 100 ) / total, "% B\n"
    escrever ( totC * 100 ) / total, "% C\n"
fim

```

#### Exemplos Civil

Arquivo: EC EXEM4.TXT

O pedreiro Paulo já rebocou 350m<sup>2</sup> e reboca em média 6,5m<sup>2</sup> de parede por hora. José começou horas depois e já fez 127m<sup>2</sup> de reboco a uma taxa média de 8,3m<sup>2</sup>/h.  
Em quantas horas José alcançará Paulo?

```

inicio
    real paulo, jose
    inteiro horas

    paulo <- 350
    jose <- 127
    horas <- 0
    enquanto paulo >= jose faz
        paulo <- paulo + 6.5
        jose <- jose + 8.3
        horas <- horas + 1
    fim enquanto
    escrever horas
fim

```

#### Exemplos Elétrica

Arquivo: EL EXEM4.TXT

Em um circuito em série, a resistência total equivale à soma de todas as resistências. Sendo assim, implemente um algoritmo que leia o valor de cada resistência até que seja informado um valor negativo. Ao final, escreva o total da resistência obtida em ohms.

```

inicio
    real resistencia, total

    ler resistencia
    enquanto resistencia >= 0 faz
        total <- total + resistencia

```

```

    ler resistencia
    fim enquanto
    escrever total
fim

```

#### Exemplos Mecânica

Arquivo: EM EXEM4.TXT

Fazer um algoritmo que leia as temperaturas em uma câmara fria. As amostras serão lidas até ser digitado o valor 999 como temperatura. Ao final, calcular e escrever as seguintes estatísticas: A quantidade de amostras coletadas, a média das temperaturas, a maior e a menor temperatura lida.

```

inicio
    inteiro cont
    real amostra, maior, menor, acumulador

    menor <-999
    ler amostra
    enquanto amostra /= 999 faz
        acumulador <- acumulador + amostra
        cont <- cont + 1
        se amostra >= maior entao
            maior <- amostra
        fimse
        se amostra <= menor entao
            menor <- amostra
        fimse
        ler amostra
    fim enquanto
    escrever cont, "\n"
    escrever acumulador / cont, "\n"
    escrever maior, "\n"
    escrever menor, "\n"
fim

```

#### Exemplos Produção

Arquivo: EP EXEM4.TXT

A medição de poluentes despejados em rios pelas indústrias determina o tamanho da multa a ser paga em prejuízo ao meio ambiente incluindo até a paralização total das atividades desta empresa.

Os valores de referência são:

- de 0 a 0,25 mg/L = isento
- de 0,26 a 0,35 mg/L = R\$ 100.000,00
- de 0,36 a 0,50 mg/L = R\$ 500.000,00
- acima de 0,50 mg/L = interdição

Faça um algoritmo que leia a quantidade de poluentes despejados (em mg/L) e a penalidade a ser aplicada até que seja digitado um valor negativo para a leitura destes poluentes.

```

inicio
    real qtde

    ler qtde
    enquanto qtde >= 0 faz
        se qtde >= 0 e qtde <= 0.25 entao
            escrever "isento \n"
        fimse
        se qtde > 0.26 e qtde <= 0.35 entao
            escrever "Multa de R$ 100.000,00 \n"
        fimse
        se qtde > 0.36 e qtde <= 0.50 entao

```

```

    escrever "Multa de R$ 500.000,00 \n"
fimse
se qtde > 0.50 entao
    escrever "Empresa interditada \n"
fimse
ler qtde
fim enquanto
fim

```

### 10.3 Laço for (para)

Exemplos Genérico

Arquivo: EG EXEM5.TXT

Elabore um algoritmo que calcule o fatorial associado a um número, fornecido pelo usuário (considere apenas números inteiros e positivos). O fatorial é calculado através da multiplicação dos números anteriores até 1.

Exemplo:  $6! = 6 * 5 * 4 * 3 * 2 * 1$ .

```

inicio
    inteiro n, i, fatorial

    ler n
    fatorial <- 1
    para i de 1 ate n passo 1
        fatorial<-fatorial*i
    proximo
    escrever fatorial
fim

```

Exemplos Civil

Arquivo: EC EXEM5.TXT

Desenvolva um algoritmo que calcule a multiplicação de dois números inteiros sem utilizar o operador de multiplicação "\*".

```

inicio
    inteiro multiplicando, multiplicador, resultado, cont

    ler multiplicando
    ler multiplicador
    para cont de 1 ate multiplicador passo 1
        resultado <- resultado + multiplicando
    proximo
    escrever resultado
fim

```

Exemplos Elétrica

Arquivo: EL EXEM5.TXT

Um dispositivo de 1 Watt consome 1 Joule de energia a cada segundo. Se você multiplicar o número de watts pelo número de segundos, terá joules como resultado. Desenvolva um algoritmo que solicite a quantidade de watts de uma lâmpada e o tempo (em segundos). Ao final, calcule a quantidade de Joules para 5 diferentes lâmpadas e intervalos de tempo após este período de acordo com a fórmula:  $J = W * s$

```

inicio
    real W
    inteiro cont, s

```

```

para cont de 1 ate 5 passo 1
    ler W
    ler s
    escrever W * s, " Joules \n"
    proximo
fim

```

#### Exemplos Mecânica

Arquivo: EM EXEM5.TXT

Considere a massa de um objeto suspenso por uma corda fixada em uma viga. Esta corda será usada para elevar esse peso a uma aceleração de 1 m/s<sup>2</sup>. Dado a força da gravidade (9,8m/s<sup>2</sup>), informe o peso de 5 diferentes objetos (em Kg), calcule e escreva a quantidade de força necessária para suspender cada objeto ( $F = Fg + m + a$ ).

```

inicio
    real m
    inteiro cont

    para cont de 1 ate 5 passo 1
        ler m
        escrever 98 + m * 1, " Newtons \n"
        proximo
fim

```

#### Exemplos Produção

Arquivo: EP EXEM5.TXT

Em um frigorífico há 30 bois. Para cada boi existe escrito em seu dorso um número que o identifica e seu peso. Faça um algoritmo que leia as informações de cada boi e a seguir, mostre o número de identificação e o peso do boi mais magro e do boi mais gordo.

```

inicio
    inteiro num, cont, num_gordo, num_magro
    real peso, mais_gordo, mais_magro

    mais_magro <- 9999
    para cont de 1 até 30 passo 1
        ler num
        ler peso
        se peso >= mais_gordo entao
            mais_gordo <- peso
            num_gordo <- num
        fimse
        se peso <= mais_magro entao
            mais_magro <- peso
            num_magro <- num
        fimse
    proximo
    escrever "Boi mais gordo ", num_gordo , " com peso ", mais_gordo, "\n"
    escrever "Boi mais magro ", num_magro , " com peso ", mais_magro
fim

```

## 11 Vetores e matrizes

#### Exercícios Genéricos

Arquivo: GN EXER4.TXT

De acordo com o código fonte abaixo, o valor a ser escrito pelo algoritmo será:

```

inicio
    inteiro vetor [ 10 ]
    inteiro cont, num, x

    num <- 5
    para cont de 0 ate 9 passo 1
        vetor [ cont ] <- vetor [ cont ] * num
    proximo
    escrever vetor [ 4 ]
fim

a) 4
b) 10
c) 20
d) 40
e) 5

```

Exercícios Civil  
Arquivo: EC EXER4.TXT

Qual será o resultado da execução do algoritmo abaixo?

```

inicio
    texto matriz[2] [2]

    matriz[0][0] <- "areia "
    matriz[0][1] <- "cimento "
    matriz[1][0] <- "cal "
    matriz[1][1] <- "agua "
    escrever matriz[0][1] + matriz[0][0] + matriz[1][1]
fim

a) matriz[0][1] + matriz[0][0] + matriz[1][1]
b) cal
c) concreto
d) cimento areia agua
e) não há resposta correta/erro

```

Exercícios Elétrica  
Arquivo: EL EXER4.TXT

O resultado da execução do algoritmo a seguir será:

```

inicio
    texto vetor [6]

    vetor [0] <- "Derivada "
    vetor [1] <- "d/dx(a) = 0 "
    vetor [2] <- "Trigonometria "
    vetor [3] <- "tg x = sen x/cos x = 1/cotg x "
    vetor [4] <- "d/dx*(a*u) = a * u' "
    vetor [5] <- "sen^2x + cos^2x = 1 "
    escrever vetor [2] + vetor [5] + vetor [3]
fim

a) Trigonometria sen^2x + cos^2x = 1 tg x = sen x/cos x = 1/cotg x
b) Trigonometria
c) Derivada

```

- d) vetor [2] + vetor [5] + vetor [3]  
e) Trigonometria  $\operatorname{tg} x = \operatorname{sen} x / \operatorname{cos} x = 1 / \operatorname{cotg} x$   $\operatorname{sen}^2 x + \operatorname{cos}^2 x = 1$

Exercícios Mecânica  
Arquivo: EM EXER4.TXT

O resultado da execução do algoritmo a seguir será:

```
inicio
    texto vetor [6]

        vetor [0] <- "Derivada "
        vetor [1] <- "d/dx(a) = 0 "
        vetor [2] <- "Trigonometria "
        vetor [3] <- "tg x = sen x/cos x = 1/cotg x "
        vetor [4] <- "d/dx*(a*u) = a * u' "
        vetor [5] <- "sen^2x + cos^2x = 1 "
        escrever vetor [2] + vetor [5] + vetor [3]
fim
```

- a) Trigonometria  $\operatorname{sen}^2 x + \operatorname{cos}^2 x = 1$   $\operatorname{tg} x = \operatorname{sen} x / \operatorname{cos} x = 1 / \operatorname{cotg} x$   
b) Trigonometria  
c) Derivada  
d) vetor [2] + vetor [5] + vetor [3]  
e) Trigonometria  $\operatorname{tg} x = \operatorname{sen} x / \operatorname{cos} x = 1 / \operatorname{cotg} x$   $\operatorname{sen}^2 x + \operatorname{cos}^2 x = 1$

Exemplo Produção  
Arquivo: EP EXER4.TXT

Em uma indústria de auto-peças, a quantidade de itens descartados mensalmente por conta de falhas no processo produtivo pode ser verificado através do vetor no algoritmo a seguir onde cada posição representa um mês no calendário.

```
inicio
    inteiro vetor [12]

        vetor [0] <- 154
        vetor [1] <- 20
        vetor [2] <- 43
        vetor [3] <- 58
        vetor [4] <- 87
        vetor [5] <- 90
        vetor [6] <- 122
        vetor [7] <- 68
        vetor [8] <- 45
        vetor [9] <- 98
        vetor [10] <- 82
        vetor [11] <- 18
        escrever vetor [2] + vetor [5] + vetor [9]
fim
```

O somatório da quantidade total de peças perdidas nos meses de fevereiro, maio e setembro correspondem a:

- a) 231  
b) 152  
c) 43, 90, 98  
d) 178  
e) 16

**11.1 Vetores**

Exemplos Genérico

Arquivo: EG EXEM6.TXT

Para o melhor desenvolvimento de frangos em um aviário, a temperatura média ao longo do dia precisa ser de 25°C. Desenvolva um algoritmo que leia e armazene a média das temperaturas durante 24h. Ao final, escreva quais as horas do dia em que essa temperatura variou em mais de +- 10% (21.6°C e 26,4°C).

```

inicio
    real temp
    inteiro hora[ 24 ],
    inteiro cont

    para cont de 0 ate 23
        ler hora[cont]
    proximo
    para cont de 0 ate 23
        se hora[cont] < 21.6 entao
            escrever "Temperatura baixa na hora ", cont, "\n"
        fimse
        se hora[cont] > 26.4 entao
            escrever "Temperatura alta na hora ", cont, "\n"
        fimse
    proximo
fim

```

Exemplos Civil

Arquivo: EC EXEM6.TXT

Em um prédio comercial de 20 andares estão instaladas diferentes atividades comerciais. Sabe-se que o tráfego de visitantes por andar não é regularmente distribuído através dos andares, portanto, construa um algoritmo que leia a quantidade diária de pessoas que circulam em cada andar, calcule e escreva a média de pessoas que passaram pelo prédio e o andar que mais recebeu visitantes.

```

inicio
    real media
    inteiro andar[ 20 ]
    inteiro cont, qtde

    para cont de 0 ate 19 passo 1
        ler andar[cont]
        media <- media + andar[cont]
        se qtde < andar[cont] entao
            qtde <- andar[cont]
        fimse
    proximo
    escrever media / 20
    escrever qtde
fim

```

Exemplos Elétrica

Arquivo: EL EXEM6.TXT

Desenvolva um algoritmo que leia 12 tensões obtidas pela leitura de um sensor em uma sala durante 24h. Armazene todos estes valores em um vetor e calcule a média de todos os valores no final do dia.

```

inicio
    real media
    real tensao[ 12 ]

```

```

inteiro cont

para cont de 0 ate 11
    ler tensao[cont]
    media <- media + tensao[cont]
proximo
escrever media / 12
fim

```

**Exemplos Mecânica**  
Arquivo: EM EXEM6.TXT

Uma montadora automobilística está conduzindo uma bateria de testes com um determinado modelo de sua fábrica. O teste em questão é o de aceleração onde serão aferidas 10 tomadas de tempo. Partindo-se do princípio de que o carro se encontra em repouso (parado), calcule a média das 10 acelerações obtidas e armazenadas em um vetor.

Fórmula:  $a = \Delta V / \Delta T = (v_f - v_i) / (t_f - t_i)$

```

inicio
real Vf, Vi, Tf, Ti, media, vetor [10]
inteiro cont

para cont de 0 ate 9 passo 1
    ler Vf
    Vi <- 0
    ler Tf
    Ti <- 0
    vetor [cont] <- ( Vf - Vi ) / ( Tf - Ti )
    media <- media + vetor [cont]
proximo
escrever media / cont, "m/s²"
fim

```

**Exemplos Produção**  
Arquivo: EP EXEM6.TXT

Desenvolva um algoritmo que leia o peso de 10 embalagens obtidas pela leitura de uma balança eletrônica durante 5 min. Armazene todos estes valores em um vetor e calcule e escreva a média de peso destas embalagens.

```

inicio
real media
real embalagem[ 10 ]
inteiro cont

para cont de 0 ate 9
    ler embalagem[cont]
    media <- media + embalagem[cont]
proximo
escrever media / 10
fim

```

## 11.2 Matrizes

**Exemplos Genérico**  
Arquivo: EG EXEM7.TXT

Dados duas matrizes A e B de ordem 3x2, crie um algoritmo que leia os valores para todas as posições destas matrizes e, ao final, calcule e escreva a soma e o produto entre elas.

```

inicio
    real A[3] [2], B [3] [2]
    inteiro i,j

    para i de 0 ate 2 passo 1
        para j de 0 ate 1 passo 1
            ler A[i][j]
            ler B[i][j]
            proximo
        proximo
        escrever "A + B\n"
    para i de 0 ate 2 passo 1
        para j de 0 ate 1 passo 1
            escrever A[i][j] + B[i][j], "\n"
        proximo
    proximo
    escrever "A * B\n"
    para i de 0 ate 2 passo 1
        para j de 0 ate 1 passo 1
            escrever A[i][j] * B[i][j], "\n"
        proximo
    proximo
fim

```

#### Exemplos Civil

Arquivo: EC EXEM7.TXT

Crie um algoritmo que calcule o determinante associado a uma matriz 3x3. O usuário deve digitar a matriz pelo teclado e os valores informados devem pertencer ao domínio dos números inteiros. Utilizando a regra de Sarrus, estudada na disciplina de álgebra linear, calcule o e escreva o seu determinante.

```

inicio
    inteiro matriz[3] [3], determinante
    inteiro i,j

    para i de 0 ate 2 passo 1
        para j de 0 ate 2 passo 1
            ler matriz[i][j]
        proximo
    proximo
    determinante <- matriz[0][0] * matriz[1][1] * matriz[2][2] + matriz[0][1] * matriz[1][2] * matriz[2][0]
    + matriz[0][2] * matriz[1][0] * matriz[2][1] - matriz[0][1] * matriz[1][0] * matriz[2][2] + matriz[0][0] *
    matriz[1][2] * matriz[2][1] + matriz[0][2] * matriz[1][1] * matriz[2][0]
    escrever determinante
fim

```

#### Exemplos Elétrica

Arquivo: EL EXEM7.TXT

Faça um algoritmo que, informadas 2 matrizes de ordem 2 (A e B) de números inteiros, realize a soma de seus elementos e armazene em uma matriz C.

```

inicio
    inteiro i,j, A[2] [2], B[2] [2], C[2] [2]

    para i de 0 ate 2 passo 1
        para j de 0 ate 2 passo 1
            ler A[i][j]
            ler B[i][j]

```

```

    proximo
proximo
para i de 0 ate 2 passo 1
    para j de 0 ate 2 passo 1
        C[i][j] <- A[i][j] + B[i][j]
    proximo
proximo
fim

```

#### Exemplos Mecânica

Arquivo: EM EXEM7.TXT

Crie um algoritmo que calcule o determinante associado a uma matriz 3x3. O usuário deve digitar a matriz pelo teclado e os valores informados devem pertencer ao domínio dos números inteiros. Utilizando a regra de Sarrus, estudada na disciplina de álgebra linear, calcule o e escreva o seu determinante.

```

inicio
    inteiro matriz[3] [3], determinante
    inteiro i,j

    para i de 0 ate 2 passo 1
        para j de 0 ate 2 passo 1
            ler matriz[i][j]
        proximo
    proximo
    determinante <- matriz[0][0] * matriz[1][1] * matriz[2][2] + matriz[0][1] * matriz[1][2] * matriz[2][0]
+ matriz[0][2] * matriz[1][0] * matriz[2][1] - matriz[0][1] * matriz[1][0] * matriz[2][2] + matriz[0][0] *
matriz[1][2] * matriz[2][1] + matriz[0][2] * matriz[1][1] * matriz[2][0]
    escrever determinante
fim

```

#### Exemplos Produção

Arquivo: EP EXEM7.TXT

Em uma confecção, a quantidade de peças de uma determinada coleção é apontada por hora durante 24h de um dia. Crie um algoritmo com uma matriz contendo 2 colunas (representando antes do meio dia e depois do meio dia) e 12 linhas (que representam as horas do período). Faça as 24 leituras, calcule e escreva em que período a fábrica é mais produtiva.

```

inicio
    real matriz[2] [12], am, pm
    inteiro i,j

    para i de 0 ate 1 passo 1
        para j de 0 ate 11 passo 1
            ler matriz[i][j]
        se i = 1 entao
            am <- am + matriz[i][j]
        senao
            pm <- pm + matriz[i][j]
        fimse
        proximo
    proximo
    se am > pm entao
        escrever "O turno da manhã é mais produtivo"
    senao
        escrever "O turno da tarde é mais produtivo"
    fimse
fim

```