

**CI-APP: UM SISTEMA DE COMUNICADO INTERNO PARA A  
FACULDADE ASSER**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO  
1. PERÍODO 2019**

Prof. Dr. Erik Aceiro Antonio  
Prof. Esp. Cristiano J. Cecanho

# 1. Introdução

As Atividades Multidisciplinares do Curso de Graduação em Sistemas de Informação da Escola Superior de Tecnologia e Educação de Rio Claro (ESRC) são partes integrantes do currículo e estão previstas no Projeto Pedagógico, aprovado no âmbito do Colegiado de Curso, sendo esta versão vigente a partir do mês de fevereiro de 2018 (vide respectivo regimento disponível em [1]).

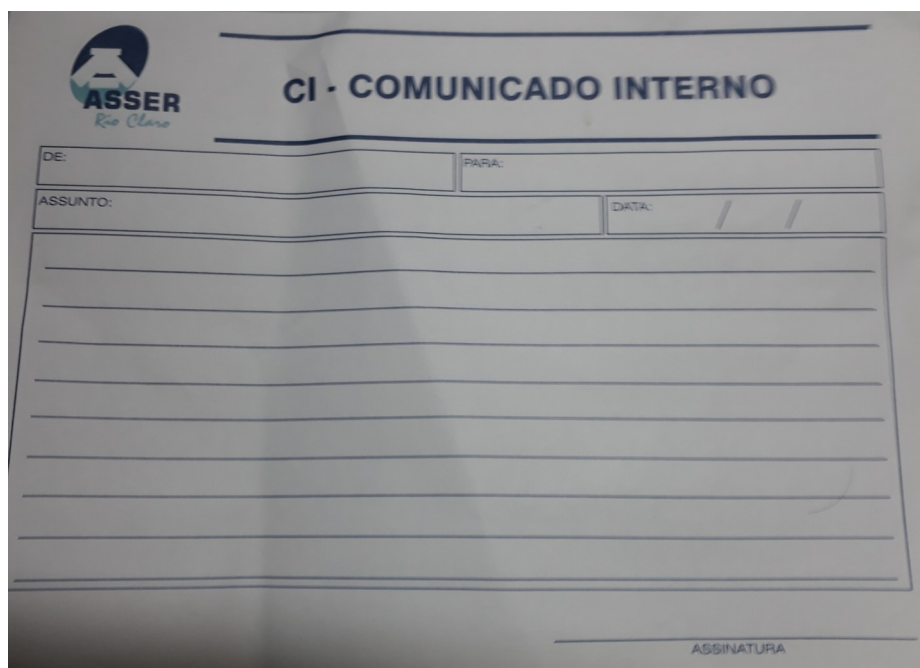
Essa atividade será desenvolvida nos moldes de uma Webquest [2] que originalmente foi proposta por Bernie Dodge, professor estadual da Califórnia (EUA) tendo como foco uma metodologia baseada no da Internet de forma criativa. A Webquest é uma atividade investigativa onde as informações com as quais os alunos interagem provêm da internet.

Portanto, para desenvolver as atividades siga as Sessões seguintes para poder compreender a atividade que será desenvolvida.

Em síntese, essa Webquest envolve as seguintes disciplinas do semestre atual de 2019: (i) Cálculo Diferencial e Integral II; (ii) Programação Orientada a Objetos; (iii) Programação de Computadores II; (iv) Economia; (v) Circuitos Lógicos; (vi) Álgebra Linear; (vii) Relações Étnicas Raciais.

Esse projeto visa o desenvolvimento de um sistema para controle de comunicação interna da Faculdade realizada através do formulário – **CI (COMUNICADO INTERNO)**. Para isso, serão realizadas algumas atividades

A **Figura 1** ilustra o formulário básico que deverá ser utilizado para auxiliar o desenvolvimento do CI-APP permitindo a automatização do processo.



O formulário é intitulado "CI - COMUNICADO INTERNO" e possui o logo da ASSER Rio Claro no canto superior esquerdo. Ele contém campos para "DE:", "PARA:", "ASSUNTO:" e "DATA:". A data é preenchida com " / /". Abaixo desses campos, há uma área com linhas horizontais para o texto do comunicado. No canto inferior direito, há uma linha para a "ASSINATURA".

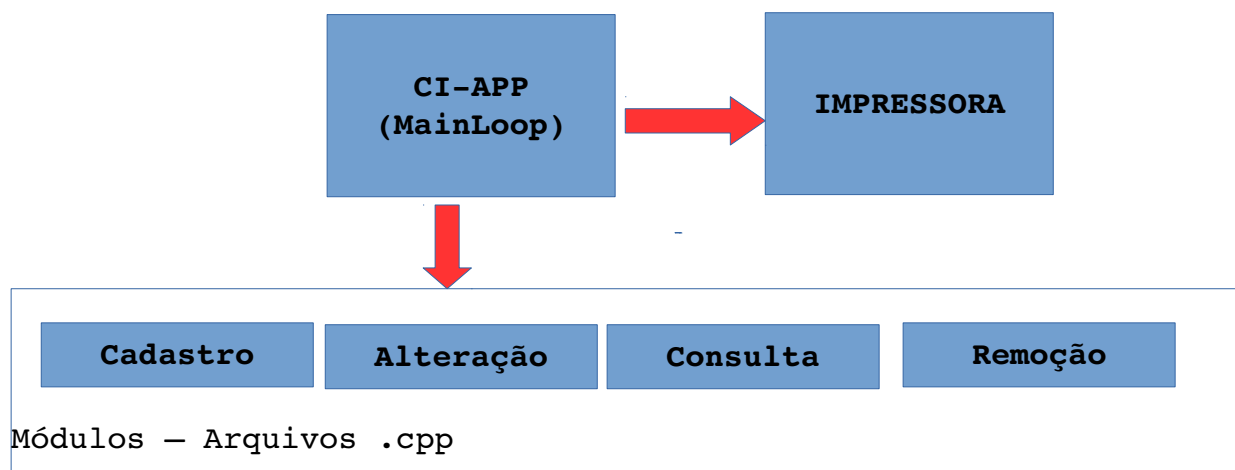
**Figura 1: Formulário de CI**

## 2. Tarefa

A partir da Figura 1 deve-se implementar uma aplicação em C++ com os seguintes requisitos básicos:

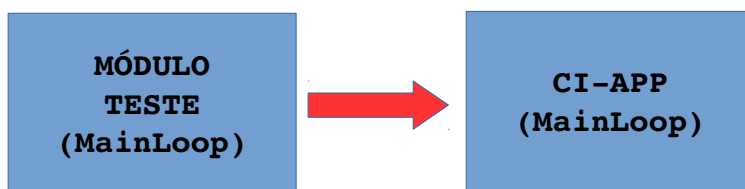
ID	REQUISITO
<b>RF1</b>	O sistema deverá gerenciar o CI contendo - entrada e saída de dados - conforme campos disponíveis na <b>Figura 1</b>
<b>RF2</b>	O sistema deve permitir o <b>cadastro</b> de entrada de dados para um <b>REGISTRO de CI</b> contendo os seguintes campos. <b>1) DE</b> (Quem está solicitando a atividade). Este campo deve ser do tipo texto e permitir no máximo 100 caracteres para indicar o nome <b>2) PARA</b> (Para quem é o comunicado). Este campo deve ser do tipo texto e permitir no máximo 100 caracteres para indicar o nome <b>3) DATA</b> (Data da geração da comunicação). Este campo é gerado automaticamente pelo sistema. A data automática deve ser gerada conforme o padrão <b>DD/MM/XXXX</b> <b>4) TEXTO</b> . Campo texto para entrada da informação a ser solicitada. Deve-se permitir no máximo 500 caracteres.
<b>RF3</b>	O sistema deve permitir a <b>atualização</b> de um <b>REGISTRO de CI</b>
<b>RF4</b>	O sistema deve permitir a <b>remoção</b> de um <b>REGISTRO de CI</b>
<b>RF5</b>	O sistema deve permitir a <b>consulta</b> de um <b>REGISTRO de CI</b>
<b>RF6</b>	O sistema deve permitir a <b>impressão</b> de um registro de CI informado pelo filtro de identificador (ID). Nesse caso, o usuário deverá informar o ID e então a Aplicação deverá enviar o registro para uma impressora. Essa impressora deve ser uma impressora Térmica
<b>RNF7</b>	O sistema deve ser validado conforme a especificação funcional de um módulo de teste auxiliar
<b>RNF8</b>	O sistema deve ser implementado seguindo os conceitos de modularização de código.

A **Figura 2** ilustra o fluxo de operação do CI-APP desde o processo inicial - que é a entrada de dados até a impressão final



**Figura 2: Módulos do CI**

A **Figura 3** ilustra o Módulo de Teste que também deverá ser desenvolvido de forma que os alunos possam assegurar a validade da aplicação segundo os critérios estabelecidos em testes funcionais (testes de unidade).



**Figura 3: Teste no Moódulo CI**

### 3. ATIVIDADE 1

Nessa primeira atividade os alunos deverão implementar a solução abstrata seguindo os conceitos com as disciplinas abaixo:


**(i) Programação Orientada a Objetos (36 horas)** – Nesse caso, a disciplina irá apoiar o conhecimento básico de POO bem como a separação em módulos do projeto. Também serão usados os conceitos vistos em aula - tais como *objetos*, *classes*, *agrupamento por arquivos (módulos)*, *instâncias* e *testes*.

**(ii) Programação de Computadores II (72 horas)** – Nesse caso, a disciplina irá apoiar os conceitos de programação básica como por exemplo, tipos de estrutura de dados que será usada, forma de entrada e saída de dados.

**(iii) Álgebra Linear (72 horas)** – Nesse caso, a disciplina irá fornecer subsídios tais como conceitos de Matrizes e Rastreabilidade.

Portanto, nessa atividade sugere-se o mapeamento através de uma Matriz de Rastreabilidade (podendo ser feita em Excel) dos principais Requisitos e seus Casos de Testes que serão utilizados (ou desenvolvidos) – indicando qual Requisito está relacionado com o Teste (vide Figura abaixo)

Matriz de rastreabilidade				
	$t_1$	$t_2$	...	$t_M$
Caso de uso 1	✓	✓		
Caso de uso 2		✓		
...				
Caso de uso N		✓		✓



Eliane Martins - Instituto de Computação - UNICAMP

MARTINS, E. Testes de Regreção. Acesso em 08 de abril de 2019. Disponível em <<https://slideplayer.com.br/slide/352225/>>

Além da Matriz de Rastreabilidade também é necessário a criação e organização do projeto em módulos usando os conceitos de POO. Para isso, sugere-se a decomposição do problema em partes menores seguindo o conceito de Dividir e Conquistar – Veja o exemplo e definição no link abaixo:

[https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/divide-and-conquer.html](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/divide-and-conquer.html)

Dessa forma para esse projeto do CI-APP serão necessários organizar e estruturar o problema em pedaços menores. Comente em grupo e estruture os arquivos, módulos e classes que serão necessários abaixo:

Arquivo	Módulo	Classe
CIApp.cpp	Módulo Principal	----
....	....	...

Para a **ATIVIDADE 1** o(s) aluno(s) deverão entregar os seguintes itens em formato ABNT e em código C/C++.

A Entrega deve ser feita através do repositório do GIT da disciplina ATM\_2019 disponível no endereço eletrônico ([https://github.com/aceiro/atm\\_2019](https://github.com/aceiro/atm_2019)).

As atividades devem ser enviadas no mesmo arquivo - **Atividades.docx conforme itens abaixo:**

- Criar no projeto do GitHub ([https://github.com/aceiro/atm\\_2019](https://github.com/aceiro/atm_2019)) as **Issues** necessárias para o desenvolvimento do projeto. Para isso, siga a estrutura básica de um projeto Kanban criando as Issues no seguinte formato:

<b>ATM2019: Descrição da Atividade a ser desenvolvida</b>
---

Crie os rótulos – **TODO / DOING / DONE** para indicar em qual fase a atividade está.

Modifique as atividades para que reflitam o status de cada uma das atividades e seu curso de desenvolvimento no processo Kanban. Siga o link abaixo

<a href="https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/">https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/</a>
---

- Para essa atividade - Enviar 1 arquivo em ABNT – **Atividades.docx** contendo:
  - Capa
  - Nome do Integrantes do grupos
  - Objetivo do Trabalho
  - Matriz de Rastreabilidade definida
- Enviar o projeto inicial em C++ com os módulos / módulo de teste inicial / módulo princila CRUD na pasta raiz. Organizar o projeto seguindo as diretrizes disponíveis no link abaixo.

<https://medium.com/heuristics/c-application-development-part-1-project-structure-454b00f9eddc>

Como sugestão, siga o exemplo do seguinte link abaixo:

#### Nome do Projeto

```

|---- README.txt
|---- include
|      |---- Nome do Projeto (arquivo públicos)
|      |      |---- public_header(s).h
|---- src
|      |---- private_header(s).h (arquivo privados ao projeto)
|      |---- code(s).cpp
|---- libs (opcional se usar alguma biblioteca externa)
|      |---- A
|      |---- B
|---- tests

```



## 4. ATIVIDADE 2

Nessa segunda atividade os alunos deverão buscar a melhor solução para a implementação do CRUD (**Create-Retrieve-Update-Delete**) do Sistema de CI. Para tanto, as seguintes disciplinas estão relacionadas

**(i) Programação Orientada a Objetos (36 horas)** – a disciplina irá apoiar o desenvolvimento e inclusão de classes que deverão ser utilizado para criar o(s) objeto(s) do projeto. Além disso, será utilizado uma biblioteca de terceiro em C++ para manipulação do menu de ações.

**(ii) Programação de Computadores II (72 horas)** – a disciplina irá apoiar a estrutura básica de implementação em C/C++ para os registros do CRUD

Em grupos, recomenda-se a leitura dos seguintes materiais visando identificar a melhor estratégia para a implementação do CRUD do Sistema de CI.

Para a **ATIVIDADE 2** o(s) aluno(s) deverão entregar os seguintes itens:

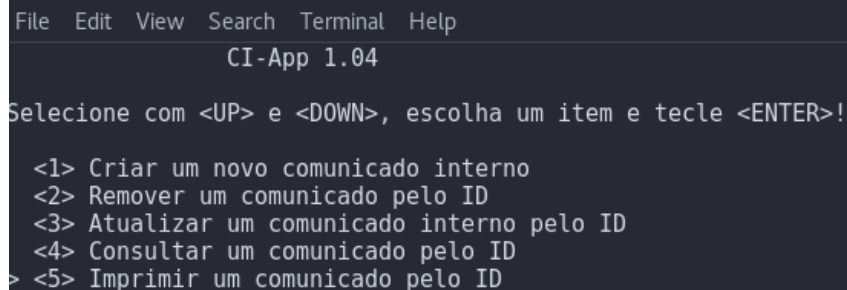
- Documento com a **Justificativa sobre qual tipo de estrutura de dados será utilizada para implementação em C/C++**. Para isso, sugerimos o uso da biblioteca STL (Standard Template Library) pois nela será possível utilizar algumas das principais estruturas de dados implementadas. Dentre elas destacamos
  - Vetores (vector)
  - Listas Ligadas (list)
  - Pilhas (queue)
  - Algoritmos de Ordenação (sort)Acesse o link abaixo para maiores informações sobre as principais estruturas de dados

<https://www.geeksforgeeks.org/the-c-standard-template-library-stl/>

- Módulo principal do sistema com o Menu. Para isso, deve-se usar a biblioteca MenuTemplate disponível no seguinte endereço eletrônico

<https://github.com/Hasenfresser/MenuTemplate>

- Deve-se incluir a biblioteca MenuTemplate ao projeto inicialmente criado e em seguida modificar e/ou adicionar as entradas para o menu de forma apropriada. Veja o exemplo abaixo:



```
File Edit View Search Terminal Help
CI-App 1.04
Selecione com <UP> e <DOWN>, escolha um item e tecle <ENTER>!
<1> Criar um novo comunicado interno
<2> Remover um comunicado pelo ID
<3> Atualizar um comunicado interno pelo ID
<4> Consultar um comunicado pelo ID
> <5> Imprimir um comunicado pelo ID
```

- Módulo principal do sistema CRUD com a estrutura de dados definida anteriormente.
  - Nesse caso, deverão ser implementadas as rotinas para CRIAR – i.e. armazenar um item na memória; R-Recupera um item da memória; U-Atualizar (do inglês, Update) e D – Apagar (do inglês, Delete)
- Módulo de teste com os principais testes identificados na **Atividade 1**.

## 5. ATIVIDADE 3

Nessa segunda atividade deverão ser implementados e integrados o restante dos módulos do CI-App. Ainda relacionado com as disciplinas de **(i) Programação Orientada a Objetos (36 horas); e (ii) Programação de Computadores II (72 horas)**.

Nessa atividade os alunos deverão entregar os seguintes itens:

- Implementação do Módulo de Criação de Registro do Comunicado Interno. Para isso os alunos deverão usar os recursos de classe, instância e de Estrutura de Dados. Cada registro criado deve ser alocado em memória. No final, é permitido a geração em disco dos registros através de um modelo de impressão que posteriormente pode ser enviado para a impressora (através de um agendamento em Linux ou Windows)
- Implementação dos outros módulos de Consulta, Remoção, Atualização do Sistema de CI-App
- Deve-se entregar os códigos fontes validados e testado através de um módulo externo

## 5. ATIVIDADE 4

Nessa segunda atividade os alunos deverão redigir um relatório final contendo o entendimento, relacionamento e importância de cada uma das disciplinas do semestre atual. Nesse caso, é importante relacionar além das disciplinas básicas de programação, as disciplinas de núcleo comum e também de áreas humanísticas.

- (i) Cálculo Diferencial e Integral II;
- (ii) Programação Orientada a Objetos;
- (iii) Programação de Computadores II;
- (iv) Economia;
- (v) Circuitos Lógicos;
- (vi) Álgebra Linear; e (vii) Relações Étnicas Raciais.

## 6. Critério de Avaliação

O trabalho será avaliado através de média ponderada com os seguintes pesos apresentados abaixo.

Item	Peso
ATIVIDADE 1	25%
ATIVIDADE 2	25%

ATIVIDADE 3	25%
ATIVIDADE 4	25%
<b>Total</b>	<b>100%</b>