**ESCOLA SUPERIOR DE TECNOLOGIA E EDUCAÇÃO DE RIO CLARO**

**ASSOCIAÇÃO DAS ESCOLAS REUNIDAS**

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

|  |  |
| --- | --- |
| Nome: Alisson Daniel Rodrigues dos Santos | RA: 8100213 |
| Nome: Amanda dos Santos | RA: 8100223 |
| Nome: Andrew Vianna Carrazzone | RA: 8100217 |
| Nome: Danilo Alves da Silva | RA: 8100211 |
| Nome: Gismar Pereira Barbosa | RA: 8100215 |
| Nome: Mileide Cristina Loureiro | RA: 8100209 |

**RIO CLARO**

**JUNHO / 2019**

**ESCOLA SUPERIOR DE TECNOLOGIA E EDUCAÇÃO DE RIO CLARO**

**ASSOCIAÇÃO DAS ESCOLAS REUNIDAS**

|  |  |
| --- | --- |
| Nome: Alisson Daniel Rodrigues dos Santos | RA: 8100213 |
| Nome: Amanda dos Santos | RA: 8100223 |
| Nome: Andrew Vianna Carrazzone | RA: 8100217 |
| Nome: Danilo Alves da Silva | RA: 8100211 |
| Nome: Gismar Pereira Barbosa | RA: 8100215 |
| Nome: Mileide Cristina Loureiro | RA: 8100209 |

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

Trabalho da disciplina de Programação Orientada a Objetos, ministrada na Escola Superior de Tecnologia e Educação de Rio Claro – ASSER entregue para avaliação.

Orientador: Prof. Dr. Erik Aceiro Antonio

**RIO CLARO**

**DEZEMBRO / 2019**

**RESUMO**

Utilizando-se a integração prática das disciplinas do semestre atual do curso de Sistemas de Informação: Cálculo Diferencial e Integral II, Programação Orientada a Objetos, Programação de Computadores II, Economia, Circuitos Lógicos, Álgebra Linear e Relações Étnicas Raciais e com base na Atividade Multidisciplinar proposta, foi desenvolvido uma aplicação denomimnada como CI-APP que visa uma melhoria de automação para o atual Formulário de Comunidade Interno da Faculdade Asser de Rio Claro.

**SUMÁRIO**

[1. OBJETIVO DO TRABALHO 5](#_Toc12611321)

[1.1 Matriz de Rastreabilidade 6](#_Toc12611322)

[1.2 Organização do Projeto 7](#_Toc12611323)

[2. TIPO DE ESTRUTURA DE DADOS UTILIZADA. 9](#_Toc12611324)

[2.1 Vetores(vector) 9](#_Toc12611325)

[3. ESTUDOS REALIZADOS PARA A CONSTRUÇÃO DO PROJETO. 10](#_Toc12611326)

[CONCLUSÃO 12](#_Toc12611327)

[REFERÊNCIAS 13](#_Toc12611328)

# OBJETIVO DO TRABALHO

Atualmente, a comunicação interna da Faculdade Asser de Rio Claro é realizada por meio do preenchimento de um formulário básico. Neste formulário o usuário precisa preencher os seguintes campos:

* De: Remetente da Mensagem;
* Para: Destinatário da Mensagem;
* Assunto: Sobre o que se trata a Mensagem;
* Data: O dia que a mensagem foi redigida;
* Mensagem: Campo destinado para a redação da mensagem;

Pensando na automatização deste procedimento de comunicação interna, foi desenvolvido um sistema que através da entrada de dados é possível: Criação, consulta, remoção e impressão dos comunicados internos. Desta forma, o usuário adquire novas funcionalidades, das quais, o procedimento no papel, não o permitia.

A atividade foi desenvolvida utilizando-se a linguagem de programação C, e baseando-se nos moldes de uma Webquest, (atividade investigativa onde as informações com as quais os alunos interagem provêm da internet). Para controle de versão de arquivos do código utilizamos o sistema GitHub.

Para a organização e desenvolvimento da atividade foi utilizado o método Kanban, que é um sistema de gerenciamento de tarefas que mostra em um quadro o fluxo de trabalho onde é possível visualizar quais tarefas estão em desenvolvimento (doing), quais já foram finalizadas (done) e as que ainda não foram realizadas (to do). O método foi adaptado ao “Issues” no GitHub, onde é possível realizar o controle das tarefas estipuladas pelo grupo:

## Matriz de Rastreabilidade

|  |  |
| --- | --- |
| **Matriz de Rastreabilidade - Módulo de Testes** |  |
| **Main:** | **QATest.cpp** |
| **Classe:** | **QAModule.hpp** |
|  | **QAModule.cpp** |
| **Teste:** | **Uso da função "assert"** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***QAModule::createTest()*** | | ***QAModule::selectTest()*** | ***QAModule::updateTest()*** | | | ***QAModule::deleteTest()*** | ***QAModule::printTest()*** |
| ***Form::auxValidateInput()*** | ***Form::formCreate()*** | ***Form::formSelect()*** | ***Form::validateInputToUpdate()*** | ***Form::returnInputToUpdate()*** | ***Form::formUpdate()*** | ***Form::formDelete()*** | ***Form::formToPrint()*** |
| **Inserir Dados** | X | X |  |  |  |  |  |  |
| **Consultar Dados** |  |  | X |  |  |  |  |  |
| **Atualizar Dados** | X |  | X | X | X | X |  |  |
| **Excluir Dados** |  |  | X |  |  |  | X |  |
| **Imprimir Dados** |  |  | X |  |  |  |  | X |

Figura 1: Matriz de Rastreabilidade

## 1.2 Organização do Projeto

A organização dos arquivos do projeto foi organizada seguindo as diretrizes conforme imagem abaixo:

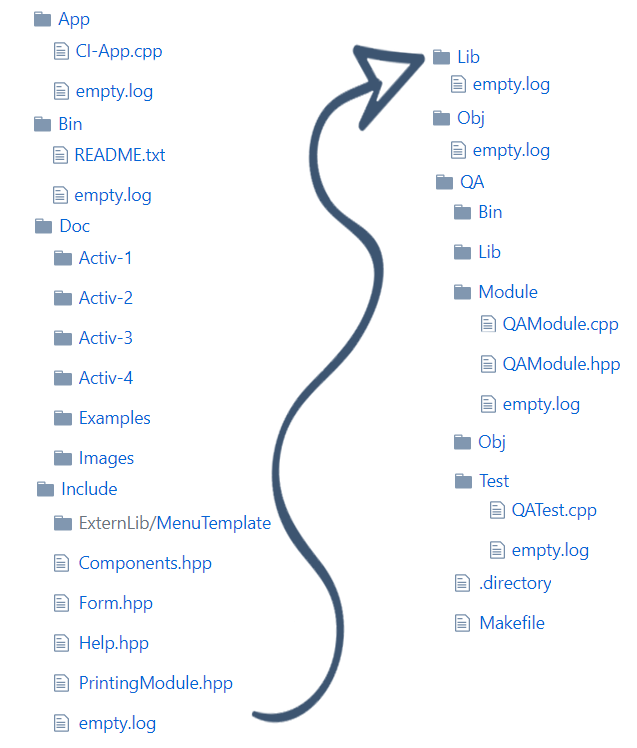


Figura 2: Organização do Projeto

Estrutura dos arquivos, módulos e classes do projeto:

|  |  |  |
| --- | --- | --- |
| ESTRUTURA PROJETO - CI-App | | |
|  | | |
| ARQUIVO | **MÓDULO** | **CLASSE** |
| CI-App.cpp | Main | --- |
| Form.hpp | Formulário (CRUD) | Form |
| FormCreate.cpp | Formulário (CRUD) | Form |
| FormSelect.cpp | Formulário (CRUD) | Form |
| FormUpdate.cpp | Formulário (CRUD) | Form |
| FormDelete.cpp | Formulário (CRUD) | Form |
| FormData.cpp | Formulário (CRUD) | Form |
| FormToPrint.cpp | Formulário (CRUD) | Form |
| Help.hpp | Ajuda | Help |
| Help.cpp | Ajuda | Help |
| Components.hpp | Componentes | Components |
| Components.cpp | Componentes | Components |
| PrintingModule.hpp | Impressão | PrintingModule |
| PrintingModule.cpp | Impressão | PrintingModule |
| QAModule.hpp | Teste | QAModule |
| QAModule.coo | Teste | QAModule |
| QATest.cpp | Teste (Main) | --- |
| MenuTemplate.hpp | Externo (Menu) | MenuTemplate |
| MenuTemplate.cpp | Externo (Menu) | MenuTemplate |
| Entry.hpp | Externo (Menu) | Entry |
| Entry.cpp | Externo (Menu) | Entry |
| getchar.hpp | Externo (Menu) | MenuTemplate |
| getchar.cpp | Externo (Menu) | MenuTemplate |

Figura 3: Estrutura do Projeto

# TIPO DE ESTRUTURA DE DADOS UTILIZADA.

Para a eficiência das buscas necessárias do programa, manipulação do grande volume de dados e a facilidade da leitura do código, a estrutura de dados de dados utilizada no projeto foi a *vector*.

## Vetores(vector)

O *vector* é um arranjo do qual possui uma capacidade de redimensionamento conforme a inserção ou remoção de um elemento, ele pode variar dinamicamente, ou seja, se o espaço reservado estive cheio, e, for necessário espaço adicional, a alocação ocorrerá de forma automática.

Para o projeto, a forma de inserção dos dados ser no final, como acontece em vetores, é viável, uma vez que, como não é possível definir uma quantidade máxima de comunicados internos, temos que ter uma estrutura dinâmica, que nos possibilite a extensão do *array* quando necessário.

A escolha da utilização do *vector*, se deu, devido ao fato de: ser um dos tipos de estrutura mais simples e com menos sobrecarga de memória em seu armazenamento, somados também suas características de manipulação dos elementos.

Em comparação com as outras estruturas de dados dinâmica (deques, listas e forward\_lists), os *vectors* são muito eficientes na questão de acesso dos seus elementos (como em matrizes). Porém a desvantagem dessa estrutura está nas operações de inserir ou remover elementos em outras posições além do final, o *vector* nestes casos não apresenta um bom desempenho, e possui iteradores e referências menos consistentes comparados com a estrutura tipo lista, por exemplo.

# ESTUDOS REALIZADOS PARA A CONSTRUÇÃO DO PROJETO.

Para o desenvolvimento do projeto, ao longo do semestre, o grupo teve o aprendizado das seguintes matérias estudadas no semestre atual: Cálculo Diferencial e Integral II, Programação Orientada a Objetos, Programação de Computadores II, Economia, Circuitos Lógicos, Álgebra Linear e Relações Étnicas Raciais. Cada qual possui uma importância significativa para a conclusão do projeto.

A matéria de Programação Orientada a Objetos, além de ser a base principal para a construção do projeto, ela também, possibilitou, a criação do código de maneira “inovadora”, no sentido de que, os desenvolvedores da aplicação apresentada, obtiveram como estudo do semestre anterior a este, a linguagem C, a qual é uma linguagem que não possui a orientação a objetos, por isto os desenvolvedores tiveram a inovação mencionada ao se depararem com a linguagem C++, que é orientada a objetos e possui várias vantagens, como por exemplo, o reuso do código de uma forma mais lógica e produtiva.

Em Programação de Computadores II, o estudo também foi com a linguagem C++. No decorrer do semestre foi apresentado teorias fundamentais para um bom programador, foi estudado: algoritmos, pilhas, filas, ordenação, Tabela Hash, Grafo e Árvores. Dentro do projeto utilizamos o muito, em especifico, o conceito de Fila.

Dentro da matéria de Cálculo Diferencial e Integral II foi apresentado conceitos e cálculos de derivadas e integrais. Com o estudo do calculo integral foi constatado como usar a integral para resolver problemas relativos a áreas, volumes, comprimentos de curva e etc. Assim, os desenvolvedores obtiveram um ganho na capacidade lógica, o que contribuiu para a execução do projeto.

Álgebra Linear e Circuitos Lógicos apresentaram conceitos extremamente relevantes dentro do mundo da programação:

* Álgebra Linear: Matrizes, Sistemas Lineares e Espaços Vetoriais.
* Circuitos Lógicos: conceitos lógicos, circuitos básicos, aritmética binária, funções booleanas, circuitos sequenciais, circuitos digitais.

Dentro dos conceitos apresentados, no projeto, muitas construções de funções foram baseadas os conceitos lógicos e funções booleanas.

# CONCLUSÃO

Com base na ATM2s2019 foi possível elaborar um projeto envolvendo a integralização dos conceitos tratados nas disciplinas do semestre. Ressalta-se também que a ATM2s2019 foi colaborativa entre todos os alunos integrantes da equipe.

O Formulário de comunicado interno da faculdade Asser que foi estudado neste projeto, já oferece e atende a necessidade da comunicação interna da faculdade, porém foi observado que seria possível através de uma aplicação melhorar a comunicação interna da faculdade.

O projeto teve como objetivo trazer uma automatização do processo, visando atender de forma mais segura, rápida e eficiente as mensagens, uma vez que, a forma convencional, formulário de papel, apresenta fragilidade, no sentido de que, o papel oferece o risco de perda, estrago ou até mesmo, por algum descuido de falha humana, que o mesmo não chegue ao remetente desejado, ocorrendo assim, uma falha na comunicação, prejudicando o destinatário e o remetente da mensagem.

Com isso, a aplicação CI-APP, a segurança da mensagem foi assegurada, pois através do armazenamento em memória o usuário pode consultar, atualizar, salvar e imprimir a mensagem.

Desta maneira, o programa apresenta uma solução que garante aos usuários envolvidos uma melhor experiência.

# REFERÊNCIAS

<http://calhau.dca.fee.unicamp.br/wiki/images/0/01/EstruturasDados.pdf>

<https://www.geeksforgeeks.org/vector-in-cpp-stl/>

<http://www.cplusplus.com/reference/vector/vector/>