FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO, TECNOLOGIA E EDUCAÇÃO

### CURSO DE SISTEMAS DE INFORMAÇÃO

**BRUNO MACHADO ESTEVES**

**BRUNO MOREIRA MACHADO**

**MATHEUS MACHADO GUERZONI DUARTE**

**ALGOCODE**

**SANTA RITA DO SAPUCAI - MG**

**2016FAI – CENTRO DE ENSINO SUPERIOR EM GESTÃO, TECNOLOGIA E EDUCAÇÃO**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

**BRUNO MACHADO ESTEVES**

**BRUNO MOREIRA MACHADO**

**MATHEUS MACHADO GUERZONI DUARTE**

**ALGOCODE**

**Projeto do 2º. Ano de Sistemas de Informação apresentado como requisito parcial para aprovação na disciplina de Linguagem de Programação sob a orientação do Prof. Dr. Fábio Gavião Avelino de Méllo.**

**SANTA RITA DO SAPUCAI - MG**

**2016**

**LISTA DE REVISÃO**

| Data | Versão | Autor | Descrição |
| --- | --- | --- | --- |
| 16/04/2016 | 0.2 | Equipe | Revisão do Documento |
| 28/04/2016 | 0.5 | Equipe | Revisão do Projeto |
| 20/05/2016 | 0.5 | Equipe | Revisão do Documento |
| 15/06/2016 | 0.9 | Equipe | Revisão do Projeto |
| 21/07/2016 | 0.7 | Equipe | Revisão do Documento |
| 21/09/2016 | 1.0 | Equipe | Revisão do Projeto |
| 24/09/2016 | 1.0 | Equipe | Revisão do Documento |

**AGRADECIMENTOS**

**RESUMO**

O AlgoCode baseia-se na tentativa de facilitar a programação. A ideia é fazer um gerador de códigos, para que o desenvolvedor possa criar seus algoritmos sem ter que os passar para uma linguagem específica que por vezes pode ser mais complicada, e a transforma em linguagem c.

**Palavras-chave: Gerador de Códigos, Algoritmos.**

**LISTA DE FIGURAS**

**LISTA DE QUADROS**

[Quadro 1– Níveis de decisão e grupos funcionais 13](#_Toc413665571)

**LISTA DE SIGLAS E ABREVIATURAS**

|  |  |
| --- | --- |
| FAI- | Centro de Ensino Superior em Gestão, Tecnologia e Educação |
| FAITEC - | Feira de Tecnologia da FAI |
|  |  |
|  |  |

**SUMÁRIO**

[1 INTRODUÇÃO 9](#_Toc420480327)

[2 DESCRIÇÃO DO PROJETO 10](#_Toc420480328)

[2.1 FORMULAÇÃO DO PROBLEMA 10](#_Toc420480329)

[2.2 JUSTIFICATIVA 10](#_Toc420480330)

[2.3 OBJETIVOS 10](#_Toc420480331)

[2.4 PÚBLICO-ALVO 10](#_Toc420480332)

[2.5 TRABALHOS RELACIONADOS 10](#_Toc420480333)

[2.6 METODOLOGIA DE TRABALHO 11](#_Toc420480334)

[2.7 ENTROPIA NEGATIVA 11](#_Toc420480335)

[3 PLANEJAMENTO 12](#_Toc420480336)

[3.1 RECURSOS DE HARDWARE 12](#_Toc420480337)

[3.2 RECURSOS DE SOFTWARE 12](#_Toc420480338)

[3.3 RECURSOS HUMANOS 12](#_Toc420480339)

[3.4 OUTROS RECURSOS 12](#_Toc420480340)

[3.5 CRONOGRAMA DE ATIVIDADES 12](#_Toc420480341)

[4 DESCRIÇÃO TÉCNICA 13](#_Toc420480342)

[4.1 NÍVEIS DE DECISÃO E GRUPOS FUNCIONAIS 13](#_Toc420480343)

[4.2 FUNCIONALIDADES PRINCIPAIS 13](#_Toc420480344)

[4.3 REQUISITOS DE HARDWARE 13](#_Toc420480345)

[4.4 REQUISITOS DE SOFTWARE 14](#_Toc420480346)

[4.5 PARADIGMA ORIENTADO A OBJETOS 14](#_Toc420480347)

[5 CONCLUSÃO 15](#_Toc420480348)

[REFERÊNCIAS 16](#_Toc420480349)

[APÊNDICE A – OBRAS CONSULTADAS 17](#_Toc420480350)

[APÊNDICE B – *PROJECT DESCRIPTION* 18](#_Toc420480351)

# 1 INTRODUÇÃO

O projeto AlgoCode é um gerador de código desenvolvido como parte da disciplina de Linguagem de Programação e ilustra, na pratica, os conhecimentos em linguagem C e C++ orientada a objetos. O gerador pretende alcançar um público mais especifico, uma vez que seu tema é mais focado em programação, pois transforma o algoritmo em linguagem C. Tem o objetivo de facilitar a criação de projetos de uma maneira dinâmica e simples.

Visto que a área de Sistemas de Informação vem crescendo bastante hoje em dia, a ideia é tornar mais fácil uma matéria considerada por muitos como difícil em algo simples. De maneira comum entre os algoritmos, o desenvolvedor cria sua ideia e o gerador a transforma em linguagem C.

# 2 DESCRIÇÃO DO PROJETO

## 2.1 FORMULAÇÃO DO PROBLEMA

A ideia principal do projeto é fazer com que uma matéria considerada complexa e difícil se torne algo simples e fácil. Utilizando o gerador de códigos AlgoCode, o usuário pode criar seus algoritmos e desfrutar de suas ideias na programação em C.

## 2.2 JUSTIFICATIVA

A maneira convencional de tratar assuntos técnicos traz o livro didático e a apostila como as principais ferramentas de ensino e aprendizagem.

Diante deste cenário, que pode ser comprovado com uma simples conversa com alunos de sistemas de informação, idealizou- se um gerador de base lógica (Linguagem C++ Orientada a Objetos) que pudesse, de modo simples, fácil e dinâmico, facilitar os estudantes e talvez até mesmo os professores.

## 2.3 OBJETIVOS

O objetivo geral do projeto é aplicar conteúdos apreendidos na disciplina de Linguagem de programação de modo prático, ou seja, desenvolvendo projetos orientados a objetos a partir de algoritmos

O objetivo especifico do gerador é favorecer o ensino e aprendizagem de modo simples. A ideia é facilitar, ajudar e ensinar.

## 2.4 PÚBLICO-ALVO

Pretende-se que possam se interessar por este projeto pessoas que sejam curiosas a respeito de conhecimentos na área de Sistemas de Informação, bem como, professores, alunos e pesquisadores dessa área. Portanto, o público alvo deste projeto caracteriza-se por faixas etárias diversas e, até mesmo, instituições de ensino, que possam utilizá-lo de forma pedagógica.

## 2.5 TRABALHOS RELACIONADOS

O projeto tem como inspiração algumas ideias das feiras de tecnologia da FAI anteriores, o grupo pesquisou e foi montando sua própria ideia.

## 2.6 METODOLOGIA DE TRABALHO

Entende-se por metodologia, grosso modo, o caminho a ser percorrido para o desenvolvimento adequado de uma pesquisa ou projeto. Assim, para que (nome do projeto) fosse criado, foi necessário mapear as ações e executá-las no tempo estimado. Primeiro, ocorreu uma pesquisa para busca de fundamentos teóricos em Linguagens de Programação, disciplina do Professor Fábio Gavião. O segundo procedimento metodológico foi o desenvolvimento em laboratório das etapas de programação em Linguagem C++.

## 2.7 ENTROPIA NEGATIVA

Apesar de ser de um gerador de codigos atual, sempre pode-se fazer mudanças, adicionando novas ideias. Diversificando a linguagem usada, evitando a monotonia para que não seja cansativo ao usuário. Acatando novas ideias sugeridas pelo público, sendo possível tal sugestão ser executada. Não deixando de manter o projeto sempre a atualizado junto das tecnologias do momento. Sendo assim, mantendo o entretenimento do público com o Projeto.

# 3 PLANEJAMENTO

## 3.1 RECURSOS DE HARDWARE

## 3.2 RECURSOS DE SOFTWARE

RS 1 - Visual Studio 2013.

RS 2 - Microsoft Word 2013.

## 3.3 RECURSOS HUMANOS

Desde o começo do projeto a equipe precisou se empenhar e aprender a utilizar a Windows Forms, junto do monitor Marcelo.

Ajuda em dúvidas em Programação com o Professor Fábio Gavião, que ministra a disciplina Linguagens de Programação.

Trabalho escrito, teoria no primeiro ano com a professora Daniella Mont’Alvão, que ministra a disciplina Metodologia de Pesquisa Científica.

## 3.4 OUTROS RECURSOS

Smartphones, Tablet, Notebooks, Desktops.

## 3.5 CRONOGRAMA DE ATIVIDADES

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome** | **Atividades** | **Jan** | **Fev** | **Mar** | **Abr** | **Mai** | **Jun** | **Jul** | **Ago** | **Set** | **Out** | **Nov** | **Dez** |
| Bruno Moreira Machado | Programação/Trabalho Escrito (PCD) | x | x |  |  |  |  |  |  |  | x | x | x |
| Matheus Machado Guerzoni Duarte | Programação / Design Gráfico | x | x |  |  |  |  |  |  |  | x | x | x |

# 4 DESCRIÇÃO TÉCNICA

## 4.1 FUNCIONALIDADES PRINCIPAIS

Arquivo – Manipulação do documento como salvar e abrir projetos

Ferramentas – Analisador léxico de palavras

Configurações – Opções para trocar a cor e fonte dos fundos e letras

Ajuda – Auxilia o usuário e mostra as informações do programa

## 4.2 REQUISITOS DE HARDWARE

Requisitos Mínimos:

**Processador:** Intel® Core™ 2 Duo CPU E4500

**Velocidade do processador:** 2.20GHz

**Memória RAM:** 2 GB

**Memória de vídeo:** 512 Mb

**Espaço:** 100 Mb

Requisitos Recomendados:

**Processador:** Intel® Core™ i3-4160

**Velocidade do processador:** 3.6 GHz

**Memória RAM:** 4 GB

**Memória de vídeo:** 2 GB

**Espaço:** 500 Mb

## 4.3 REQUISITOS DE SOFTWARE

Requisitos Mínimos:

**Sistemas Operacionais:** Windows Vista, Windows 7, Windows 8 , Windows 10

Requisitos Recomendados:

**Sistemas Operacionais:** Windows 7, Windows 8, Windows 10

## 4.4 PARADIGMA ORIENTADO A OBJETOS

**4.4.1 Análise**

Fazer uma lista com os nomes das classes descobertas na fase de análise e explicar a finalidade de cada uma de forma bem objetiva.

Fazer o diagrama de classes e mostra-lo nesta seção após a lista das classes. Este diagrama deverá mostrar os relacionamentos principais apenas: dependência, herança e composição.

Apenas as classes principais deverão ser mostradas nesta seção.

**4.4.2 Projeto**

Listar os protótipos dos principais métodos das classes listadas na seção anterior identificando sua finalidade.

**4.4.3 Codificação**

O padrão seguido foi usar o nome dos objetos e o nome de cada ação para a sua identificação em inglês.

Ex: getStatus, ConvertAlgorithmToC, AddLineNumbers.

**4.4.4 Testes previstos**

Descreva os testes que foram realizados para os principais métodos de algumas classes selecionadas mostrando o que é dado de entrada e uma comparação entre a saída esperada e a ocorrida.

Testes de espaçamento, testes de capitalização, tentando ao máximo encontrar erros para evitar conversões incorretas e conflitos com o código.

Teste de espaçamento: Todos os possíveis espaçamentos que o usuário pode colocar foram testados para que não houvesse interferência no resultado.

Teste de capitalização: Foi testado os possíveis erros de capitalização de caracteres para que as verificações pudessem ser realizadas com precisão

4.5 MANUAL DO USUÁRIO

Faça o manual do usuário, como um documento separado e coloque-o em Apêndice.