



UNIVERSIDADE FEDERAL DO CEARÁ - UFC  
CAMPUS CRATEÚS  
CIÊNCIA DA COMPUTAÇÃO

**Disciplina:** Projeto Integrador II  
Prof. Ítalo Mendes da Silva Ribeiro

## Documento Integrador

### Sumário

<b>1</b>	<b>Projetos</b>	<b>2</b>
1.1	Projeto 1 - Show do Milhão . . . . .	2
1.2	Projeto 2 - Controle de Gastos . . . . .	7
1.3	Projeto 3 - Gerenciador de Campeonato . . . . .	11
1.4	Projeto 4 - Leitor de PDF . . . . .	15
1.5	Projeto 5 - Gerenciador de Arquivos . . . . .	18
1.6	Projeto 6 - Jogo de Conscientização sobre o Aedes aegypti . . . . .	21
1.7	Projeto 7 - Troca de Mensagens Instantâneas . . . . .	25
1.8	Projeto 8 - Armazenamento de Arquivos em Nuvem . . . . .	29
1.9	Projeto 9 - Simulador da Transmissão de Pacotes na Rede . . . . .	33
<b>2</b>	<b>Regras Gerais</b>	<b>37</b>

## Projeto 1

## Show do Milhão

## Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

## Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

## Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

## Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

## Competências

Comunicação

Planejamento e organização

Proatividade  
Integridade  
Trabalho em equipe

## Habilidades

---

Prototipação  
Programação de computadores  
Estruturação de informações em Banco de Dados

## Descrição da Temática

---

Jogo de perguntas e respostas onde o conteúdo é gerenciado pela comunidade de jogadores. O jogo permite aos usuários ampliação e teste dos seus conhecimentos, de forma simples e divertida. O gerenciamento das perguntas pela comunidade manterá as questões atuais e corretas, como ocorre em outros conteúdos geridos por comunidades com o Wikipédia. Jogos no estilo são famosos e fazem sucesso em programas televisivos como o [Show do Milhão](#) e [Who Wants to Be a Millionaire?](#)

A equipe envolvida no projeto praticará o desenvolvimento de interface e interação para jogos, que normalmente permitem elementos visuais mais coloridos e com aparência diferente dos demais programas de computador. As informações utilizadas no jogo deverão ser estruturadas. A fim de organizar o desenvolvimento é importante estruturar as classes e pacotes do sistema, além de estabelecer a ordem de execução das ações no sistema. Por isso, a documentação do sistema em diagramas UML é um passo fundamental para organizar e padronizar a comunicação para toda equipe, além de encontrar possíveis problemas antes do desenvolvimento do sistema. Padrões de projeto também são necessários para uma boa organização do código e facilitar a implementação.

## Produto

---

Jogo de perguntas, em que o objetivo é conseguir o prêmio de 1 milhão de reais. As perguntas são de múltipla escolha com quatro alternativas. Em cada pergunta, o jogador poderá: responder a pergunta; parar de responder; errar a resposta; pedir a eliminação de duas respostas erradas; ou denunciar a pergunta por enunciado, alternativas ou resposta incorreta. O jogador poderá pedir a eliminação de duas respostas erradas uma única vez. Se responder corretamente, o jogador passa para a próxima pergunta. De acordo com a pergunta em que o jogador estiver, ele receberá um prêmio em dinheiro se o jogador responder corretamente, parar ou errar a resposta de acordo com a tabela:

Nº da Pergunta	Acertar	Parar	Errar
1	R\$ 1 mil	R\$ 0	R\$ 0
2	R\$ 5 mil	R\$ 1 mil	R\$ 500
3	R\$ 50 mil	R\$ 5 mil	R\$ 2.5 mil
4	R\$ 100 mil	R\$ 50 mil	R\$ 25 mil
5	R\$ 300 mil	R\$ 100 mil	R\$ 50 mil
6	R\$ 500 mil	R\$ 300 mil	R\$ 150 mil
7	R\$ 1 milhão	R\$ 500 mil	R\$ 0

O usuário precisa de uma conta para jogar. A conta do usuário possui as informações de:

- Nome;
- Nickname;
- Avatar (foto);
- Lista de perguntas adicionadas e aceitas;
- Lista de perguntas adicionadas e não aceitas;
- Número de partidas jogadas;
- Número total de perguntas respondidas;
- Premiação total de todas as partidas jogadas;
- Quantidade de utilizações da eliminação de duas alternativas;
- Número de derrotas por erro da resposta;
- Número de derrotas por pedido de parada;

O programa tem as seguintes funcionalidades:

1. Criação de conta;
2. Login;
3. Deslogar;
4. Edição de conta;
5. Remoção de conta;
6. Adição de pergunta. A pergunta é aceita quando cinco outros usuários escolhidos aleatoriamente aprovam-na;
7. Lista de perguntas adicionadas aceitas e não aceitas do usuário;
8. Edição de perguntas do usuário que foram adicionadas e **não aceitas**;
9. Revisão de pergunta. Quando uma pergunta é denunciada por dois usuários, a pergunta é enviada para cinco usuários verificarem o enunciado, alternativas e resposta correta;
10. Resposta das perguntas do jogo (necessárias telas de vitória e derrota do jogador);
11. Eliminação de duas respostas erradas;
12. Embaralhar as respostas em cada vez que pergunta é mostrada;
13. Hall da fama, com lista dos 10 jogadores com as maiores premiações totais do jogo;

14. Armazenamento das informações em banco de dados.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

É permitida utilização de qualquer framework para manipulação do BD e emprego de qualquer SGBD. Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 9.

## Metodologia

- Definição dos requisitos do jogo;
- Elaboração dos diagramas UML;
- Elaboração do modelo das tabelas do banco de dados;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Estrutura das tabelas no banco de dados;
5. Normalização das tabelas do banco de dados;
6. Utilização dos princípios SOLID;

## Cronograma

Ver 13º artigo das regras gerais

## Projeto 2

### Controle de Gastos

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

## Descrição da Temática

---

O controle de gastos é uma atividade importante para o controle financeiro de uma pessoa, família ou empresa, sendo importante para ajudar na educação financeira. Programas que armazenam as informações de receitas e gastos são importantes para facilitar o gerenciamento financeiro, auxiliando uma tarefa que para muitos é tediosa e complicada, dificultando ou mesmo impossibilitando o controle de gastos. Com as informações precisas dos gastos é muito fácil para o usuário encontrar as principais despesas, e assim equilibrar melhor suas finanças.

A equipe desenvolverá um programa de controle financeiro, onde o usuário informará suas receitas e despesas organizando-as objetivando melhorar suas finanças, e permitindo uma maior economia no final do mês.

A equipe envolvida no projeto praticará o desenvolvimento de interface e interação para criação de um sistema de fácil utilização. A fim de organizar o desenvolvimento é importante estruturar as classes e pacotes do sistema, além de estabelecer a ordem de execução das ações no sistema. Por isso, a documentação do sistema em diagramas UML é um passo fundamental para organizar e padronizar a comunicação para toda equipe, além de encontrar possíveis problemas antes do desenvolvimento do sistema. Padrões de projeto também são necessários para uma boa organização do código e facilitar a implementação.

## Produto

---

Programa de controle financeiro. É fornecida a receita do usuário para o mês, e a cada nova despesa adicionada, o valor disponível de receita no mês pelo usuário é atualizado. Uma conta de usuário é necessária para usar o programa. A conta terá nome completo, email e avatar.

Uma despesa tem os seguintes dados:

- Valor;
- Data;
- Descrição;
- Categoria (casa, carro, alimentação, animal de estimação, etc);
- Forma de pagamento (cartão de débito, cartão de crédito ou dinheiro);
- Número de parcelas (apenas para cartão de crédito);
- Se está paga ou não.



As informações dos cartões de crédito e débito seguirão o exemplo da tabela seguinte:

NÚMERO	Tipo	Bandeira	Limite (R\$)	Valor Atual (R\$)	Dia Fechamento
1122 1833	Crédito	Cielo	3000	0	28
2347 9326	Crédito	Mastercard	1500	0	24
5558 0991	Débito	Visa	Receita do Mês	-	-
0933 4967	Débito	Hipercard	Receita do Mês	-	-

As funcionalidades do programa são:

1. Criação de conta;
2. Login;
3. Logout;
4. Edição de conta;
5. Remoção de conta;
6. CRUD de receita;
7. CRUD de cartão de crédito;
8. CRUD de cartão de débito;
9. CRUD de categoria. Uma categoria tem apenas o nome;
10. CRUD de despesa;
  - Na adição de despesa verificações são obrigatórias. Se a despesa for em cartão de crédito, é necessário verificar se a despesa não ultrapassará o valor limite do cartão e se não ultrapassou o dia de fechamento da fatura, para inclusão da despesa na fatura do mês atual ou do mês seguinte. Caso a despesa seja em cartão de débito, deve-se verificar se ainda existe saldo no cartão (receita do mês);
  - Nas despesas pagas em cartão de crédito, o valor da prestação será incluído automaticamente nas faturas seguintes de acordo com o número de prestações em que a despesa é dividida;
11. Nas listagens de cada CRUD deverão existir opções para reordenação ou filtragem dos itens. Por exemplo, as despesas listadas podem ser reordenadas pelo valor, data, categoria, etc. Além disso, as despesas são filtráveis pelo mês, categoria, etc;

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

É permitida utilização de qualquer framework para manipulação do BD e emprego de qualquer SGBD. Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 11.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Elaboração do modelo das tabelas do banco de dados;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Estrutura das tabelas no banco de dados;
5. Normalização das tabelas do banco de dados;
6. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)

## Projeto 3

### Gerenciador de Campeonato

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

## Descrição da Temática

---

Atividades esportivas são importantes para a saúde, ajudando na prevenção e no combate de várias doenças como a depressão. É muito comum a realização de práticas esportivas em grupo, as vezes os amigos gostam de competir para tornar a atividade esportiva mais interessante. A organização e o acompanhamento de uma competição esportiva pode ser mais prática com um programa para gerenciar o torneio. Assim, os integrantes da equipe desenvolverão um programa para organizar um campeonato esportivo, em que o modo de competição e a modalidade esportiva será decidido pela equipe.

A equipe envolvida no projeto praticará o desenvolvimento de interface e interação para criação de um sistema de fácil utilização. A fim de organizar o desenvolvimento é importante estruturar as classes e pacotes do sistema, além de estabelecer a ordem de execução das ações no sistema. Por isso, a documentação do sistema em diagramas UML é um passo fundamental para organizar e padronizar a comunicação para toda equipe, além de encontrar possíveis problemas antes do desenvolvimento do sistema. Padrões de projeto também são necessários para uma boa organização do código e facilitar a implementação.

## Produto

---

Programa para gerenciamento de uma competição esportiva. A modalidade esportiva é definida pela equipe, abrangendo desde esportes tradicionais e eSports. A forma de realização da competição também será decidida pela equipe: mata-mata, pontos corridos, grupos seguida de mata-mata, etc. Uma conta de usuário é necessária para usar o programa. A conta terá nome completo, email, avatar e torneios criados.

Cada competição terá os seguintes dados:

- Nome;
- Descrição;
- Quantidade de times;
- Premiação;
- Forma de competição.

Uma partida terá:

- Data;

- Horário;
- Local;
- Dois times;
- Placar;
- Momento da pontuação (competidor que marcou o ponto e tempo da partida, como um gol, ganho de um set, etc.);
- Ao menos 4 estatísticas (nome e valor). Por exemplo: cartões amarelos, chutes ao gol, aces, bloqueios, etc.

Um time possuirá:

- Nome;
- Imagem do escudo;
- Abreviação;
- Nome dos competidores do times.

As funcionalidades do programa são:

1. Criação de conta;
2. Login;
3. Logout;
4. Edição de conta;
5. Remoção de conta;
6. CRUD de time;
7. CRUD de competição;
8. CRUD de partida;
9. Sorteio automático dos times, grupos, partidas, etc. de acordo com a forma de competição. Após a conclusão do sorteio, o resultado pode ser alterado, como times, grupos, etc;
10. Exibição da forma de organização da competição, como os grupos, mata-mata, tabela de pontos, etc; de acordo com a forma de competição;
11. Nas listagens de cada CRUD deverão existir opções para reordenação ou filtragem dos itens. Por exemplo, os times listados podem ser reordenadas ou filtrados pelo nome.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

É permitida utilização de qualquer framework para manipulação do BD e emprego de qualquer SGBD. Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 11.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Elaboração do modelo das tabelas do banco de dados;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Estrutura das tabelas no banco de dados;
5. Normalização das tabelas do banco de dados;
6. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)

## Projeto 4

### Leitor de PDF

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

## Descrição da Temática

---

Arquivos no formato PDF (*Portable Document Format*) são amplamente utilizados para documentos digitais, os quais deseja-se que o conteúdo seja visto, mas não editado, como por exemplo livros, escrituras, contratos, etc. Porém, em certas situações o usuário pode necessitar adicionar informações e comentários ao documento, e muitos editores de PDF não possuem funcionalidades para adição de comentários, e os editores que possuem, as vezes são limitadas ou são pagos. Assim, um visualizador de PDF leve, multiplataforma e que permita a incorporação de comentários ao documento terá uma boa chance de êxito.

O grupo desenvolverá habilidades para manipular um determinado formato de arquivo muito utilizado, além de praticar a prototipação de interface com boa interação e usabilidade. A fim de organizar o desenvolvimento é importante estruturar as classes e pacotes do sistema, além de estabelecer a ordem de execução das ações no sistema. Por isso, a documentação do sistema em diagramas UML é um passo fundamental para organizar e padronizar a comunicação para toda equipe, além de encontrar possíveis problemas antes do desenvolvimento do sistema. Padrões de projeto também são necessários para uma boa organização do código e facilitar a implementação. Além disso, as anotações com formas geométricas utilizarão alguns conceitos de transformações geométricas de Computação Gráfica.

## Produto

---

Visualizador e editor de comentários de documentos PDF. O software deverá ser desktop e multiplataforma.

As funcionalidades do programa são:

1. Visualização de arquivo PDF;
2. Salvar arquivo PDF com os comentários feitos no arquivo;
3. Adição de imagens sobre o conteúdo do PDF;
4. Inclusão de comentários em formato de texto, onde o texto terá as seguintes opções:
  - tamanho da fonte;
  - cor da fonte.
5. Adição de forma geométrica sobre o conteúdo do PDF, com as seguintes opções:
  - forma geométricas possíveis: linha, retângulo, elipse/círculo;



- As formas podem ser apenas as bordar ou não;
- cor da forma;
- transformações das formas: translação, rotação e escala.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 3.

## Metodologia

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Utilização dos princípios SOLID;

## Cronograma

[Ver 13º artigo das regras gerais](#)

## Projeto 5

### Gerenciador de Arquivos

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

## Descrição da Temática

---

Gerenciador de arquivo é um software importante para visualizar e gerenciar os arquivos no dispositivo. Assim, as funcionalidades e as maneiras de visualização dos arquivos presentes no gerenciador, contribuem para a execução das tarefas do usuário.

O grupo desenvolverá habilidades de prototipação de interface com boa interação e usabilidade. Objetivando organizar o desenvolvimento é importante estruturar as classes e pacotes do sistema, além de estabelecer a ordem de execução das ações no sistema. Por isso, a documentação do sistema em diagramas UML é um passo fundamental para organizar e padronizar a comunicação para toda equipe, além de encontrar possíveis problemas antes do desenvolvimento do sistema. Padrões de projeto também são necessários para uma boa organização do código e facilitar a implementação.

## Produto

---

Gerenciador de arquivos. O software deverá ser desktop e multiplataforma.

As funcionalidades do programa são:

1. exibição dos arquivos de um diretório;
2. abrir arquivo com software adequado;
3. renomear arquivo;
4. remover arquivo para a lixeira, após confirmação pelo usuário;
5. copiar arquivo;
6. colar arquivo;
7. renomear diretório;
8. remover diretório e todos seus os arquivos para a lixeira, após confirmação pelo usuário;
9. copiar diretório e todos seus os arquivos;
10. colar diretório e todos seus os arquivos;
11. abas, onde cada aba poderá exibir um diretório diferente;
12. painel lateral com atalho para arquivos e diretórios favoritos.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;

- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 12.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Crterios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)

## Projeto 6

### Jogo de Conscientização sobre o *Aedes aegypti*

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

## Descrição da Temática

---

Todo ano, o Brasil tem milhares de casos de dengue, zika e chikungunya. Em 2020, de acordo com [boletim epidemiológico do Ministério da Saúde](#), foram notificados 987.173 casos prováveis de Dengue, 82.419 de chikungunya e 7.387 de zika. Além de muitas mortes, as três doenças causam também várias problemas as pessoas que se recuperam dos vírus, como microcefalia congênita. As três doenças são causadas por vírus e transmitidas ao ser humano pelo mosquito *Aedes aegypti*. Por isso a prevenção contra o mosquito é fundamental para o controle das doenças. Uma forma lúdica e prática de conscientização e prevenção são jogos eletrônicos.

Assim, o grupo criará um jogo para mostrar como o mosquito se propaga, os impactos da propagação do mosquito na vizinhança próxima ao foco da doença e métodos para impedir a multiplicação do *Aedes aegypti*. Assim, o jogo objetivará mostrar a importância da participação de cada um, no combate ao *Aedes aegypti* e as doenças causadas por ele.

O grupo praticará a prototipação de interface com boa interação e usabilidade, para que o jogo seja de fácil aprendizado e contribuir para a prevenção das doenças. A fim de organizar o desenvolvimento é importante estruturar as classes e pacotes do sistema, além de estabelecer a ordem de execução das ações no sistema. Por isso, a documentação do sistema em diagramas UML é um passo fundamental para organizar e padronizar a comunicação para toda equipe, além de encontrar possíveis problemas antes do desenvolvimento do sistema. Padrões de projeto também são necessários para uma boa organização do código e facilitar a implementação. Além disso, vários conceitos de Computação Gráfica serão usados como transformações geométricas, projeção de câmera, texturas, modelos 3D, etc.

## Produto

---

Jogo de simulação de vizinhança e propagação do *Aedes aegypti*. Cada residência terá de 1 a 5 moradores, onde cada morador será de três tipos, de acordo com a probabilidade de realizar alguma ação de prevenção contra o *Aedes aegypti*.

- consciente: 98% de chance de realizar ação preventiva;
- indiferente: 50% de chance de realizar ação preventiva;
- negacionista: 1% de chance de realizar ação preventiva.

O tipo de morador e a quantidade de moradores em cada residência são definidos aleatoriamente no início do jogo. Se o morador ou um conhecido do morador fica doente de dengue, zika ou

chikungunya, o morador aumenta em 50% a sua chance de realizar alguma ação preventiva, até um limite de 100%. Qualquer morador pode ficar doente, independente da porcentagem de tomar ação correta. Caso o morador fique doente, ele terá 25% de chance de morrer.

Quando um mosquito nasce, ele voa para uma residência aleatória do bairro, e caso algum local da residência esteja propício para desenvolvimento das larvas, o mosquito colocará seus ovos. São locais propícios:

- garra com água;
- pneu velho com água;
- vaso de planta com água no pratinho da base;
- caixa d'água destampada;
- piscina com pouca água.

Depois de 2 dias, o ovo virá larva. Após 2 dias, a larva se tornará um mosquito que voará para uma residência próxima.

Aleatoriamente, em alguns momentos durante a semana, cada morador andará por sua residência por um caminho aleatório, ao passar próximo a algum local propício, o jogador poderá clicar no local propício, o morador pode ou não realizar alguma ação preventiva, de acordo com sua porcentagem de realização. Caso o morador execute a ação de prevenção, o local deixará de ser propício. Dessa forma, o local mudará da seguinte maneira:

- garra ficará vazia e com a boca virada para baixo;
- pneu velho ficará sem água;
- vaso de planta terão terra no pratinho da base;
- caixa d'água será tampada;
- piscina ficará sem água.

No início do jogo, de 1 a 3 mosquitos *Aedes aegypti* voarão e colocarão ovos em local propício. Cada 6 segundos equivalem a 1h no jogo.

O jogador perderá quando 10 moradores morrerem.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)



## Projeto 7

### Troca de Mensagens Instantâneas

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

CRT0038 - Redes de Computadores

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

Antonio Emerson Barros Tomaz

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

Comunicação através de redes de computadores

## Descrição da Temática

---

A troca de mensagens instantâneas é amplamente realizada por pessoas e empresas, permitindo uma troca rápida de conteúdo, desde textos, imagens, áudios, animações e vídeos. Assim, uma vasta possibilidades de comunicação é possível, tanto para um único contato, como para um grupo de pessoas. Dessa forma, os softwares de mensagem instantânea fazem parte do dia-a-dia de muitas pessoas e empresas, sendo imprescindíveis para o negócio de muitas empresas, como para diversão de muitas pessoas.

O grupo implementará um software de envio de mensagens instantâneas, onde os membros da equipe praticarão habilidades de documentação e organização do software com diagramas UML, antes de início do desenvolvimento. Além disso, o grupo praticará o funcionamento e comunicação em redes de computadores, que envolvem conhecimentos e habilidades específicas de softwares que realizam comunicação em redes, criando uma estrutura bem diferente, de softwares sem comunicação em redes. Os alunos também desenvolverão habilidades em linguagens funcionais e a construção de interfaces que empregará conhecimentos de Computação Gráfica.

## Produto

---

Software de envio de mensagens instantâneas. O usuário enviará mensagens no formato de texto, imagem, animação, vídeo e áudio. Uma conta de usuário é necessária para utilização do programa. A conta terá nome, email, avatar e uma descrição.

Uma mensagem tem os seguintes dados:

- data de envio;
- horário de envio;
- data de recebimento;
- horário de recebimento;
- emissor;
- destinatário;

- status: enviada, recebida e vista;
- conteúdo: texto, imagem, animação, video ou áudio.

As funcionalidades do programa são:

1. Criação de conta;
2. Login;
3. Logout;
4. Edição de conta;
5. Remoção de conta;
6. Envio e recebimento de mensagem;
7. visualização de mensagem;
8. remoção de mensagem;
9. busca por mensagem;
10. Grupo de usuários para troca de mensagens, onde um ou mais usuários serão administradores do grupo e poderão adicionar ou remover usuários do grupo.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

É permitida utilização de qualquer framework para manipulação do BD e emprego de qualquer SGBD.

O grupo deve usar apenas as funções básicas de comunicação em redes disponíveis na linguagem de programação utilizada. Não é permitida utilização de qualquer framework, API ou biblioteca já pronta com recursos para comunicação em redes de computadores.

Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 10.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Elaboração do modelo das tabelas do banco de dados;
- Criação do protótipo;
- Testes com usuários;

- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Estrutura das tabelas no banco de dados;
5. Normalização das tabelas do banco de dados;
6. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)

## Projeto 8

### Armazenamento de Arquivos em Nuvem

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

CRT0038 - Redes de Computadores

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

Antônio Emerson Barros Tomaz

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade

Trabalho em equipe

## Habilidades

---

Prototipação

Programação de computadores

Estruturação de informações em Banco de Dados

Comunicação através de redes de computadores

## Descrição da Temática

---

O armazenamento de arquivos em serviço de nuvens, se tornou indispensável para evitar a perda de arquivos, seja por problemas em hardware ou mesmo vírus. O serviço também permite o compartilhamento do arquivo de maneira prática. Grandes empresas de tecnologia, observando a demanda pelo serviço de armazenamento de arquivos em nuvens oferecem softwares para o armazenamento, como o Dropbox, Google Drive, OneDrive, etc.

A equipe implementará um software de armazenamento e compartilhamento de arquivos em nuvem, onde os membros da equipe praticarão habilidades de documentação e organização do software com diagramas UML, antes de início do desenvolvimento. Além disso, o grupo praticará o funcionamento e comunicação em redes de computadores, que envolvem conhecimentos e habilidades específicas de softwares que realizam comunicação em redes, criando uma estrutura bem diferente, de softwares sem comunicação em redes. Os alunos também desenvolverão habilidades em linguagens funcionais e a construção de interfaces que empregará conhecimentos de Computação Gráfica.

## Produto

---

Software de armazenamento e compartilhamento de arquivos em nuvem. O usuário indicará um diretório raiz no seu dispositivo, onde todos os arquivos e pastas dentro do diretório raiz serão enviados para o servidor na nuvem. Qualquer tipo de arquivo é aceito para armazenamento. O usuário necessitará de uma conta para utilização do serviço. A conta terá nome, email, avatar e lista dos dispositivos conectados.

Um mesmo arquivo deve possuir uma cópia, em cada dispositivo que o usuário tiver o cliente conectado ao serviço. Quando um arquivo ou pasta é adicionado, removido ou modificado dentro do diretório raiz, a atualização deve ser enviada para a nuvem, e em seguida passada aos clientes conectados, sem a necessidade de uma ação ou comando adicional pelo usuário. Cada arquivo deve possuir ao menos:

- data da última modificação;
- lista de dispositivos do cliente, os quais ainda não foi atualizado com a última versão do arquivo.

As funcionalidades do programa são:

1. Criação de conta;
2. Login;
3. Logout;
4. Edição de conta;
5. Remoção de conta;
6. Envio de arquivo para nuvem;
7. Atualização de arquivo nos clientes;
8. Compartilhamento de arquivo com um link.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

É permitida utilização de qualquer framework para manipulação do BD e emprego de qualquer SGBD.

O grupo deve usar apenas as funções básicas de comunicação em redes disponíveis na linguagem de programação utilizada. Não é permitida utilização de qualquer framework, API ou biblioteca já pronta com recursos para comunicação em redes de computadores.

Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 8.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Elaboração do modelo das tabelas do banco de dados;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Estrutura das tabelas no banco de dados;
5. Normalização das tabelas do banco de dados;
6. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)



## Projeto 9

### Simulador da Transmissão de Pacotes na Rede

#### Identificação

**Componente Curricular:** Projeto Integrador

**Semestre Letivo:** 2021.2

**Professor Responsável pelo PI:** Ítalo Mendes da Silva Ribeiro

#### Componentes Curriculares Integralizados

CRT0025 - Programação Funcional

CRT0034 - Análise e Projeto de Sistemas

CRT0388 - Computação Gráfica

CRT0038 - Redes de Computadores

#### Professores das Disciplinas Integralizadas

Luiz Alberto do Carmo Viana

Bruno Castro Honorato Silva

Arnaldo Barreto Vila Nova

Antonio Emerson Barros Tomaz

#### Linhas de Extensão

Desenvolvimento tecnológico

Metodologias e estratégias de ensino/aprendizagem

Tecnologia da Informação

#### Competências

Comunicação

Planejamento e organização

Proatividade

Integridade  
Trabalho em equipe

## Habilidades

Prototipação  
Programação de computadores  
Estruturação de informações em Banco de Dados  
Comunicação através de redes de computadores

## Descrição da Temática

As redes de computadores são muito empregadas em várias atividades de empresas e pessoas, nos mais diversos tipos de dispositivos eletrônicos. Assim, o aprendizado de como as informações trafegam nas redes é importante para definição da comunicação, estruturação e prevenção de problemas dos sistemas. Dessa forma, uma maneira de simular e visualizar o transporte dos pacotes de dados entre origem e destino nas redes de computadores, e informações relacionadas ao transporte, são úteis para auxiliar o aprendizado dos conceitos de movimentação dos pacotes na rede.

O grupo implementará um software para simular o transporte de pacotes em redes de computadores, onde os membros da equipe praticarão habilidades de documentação e organização do software com diagramas UML, antes de início do desenvolvimento. Além disso, o grupo praticará conceitos de comunicação em redes de computadores, que envolvem conhecimentos e habilidades específicas de softwares que realizam comunicação em redes, criando uma estrutura bem diferente, de softwares sem comunicação em redes. Os alunos também desenvolverão habilidades em linguagens funcionais e a construção de interfaces que empregará conhecimentos de Computação Gráfica.

## Produto

Software para simulação de transporte de pacotes em redes de computadores, como a Internet. O tamanho da mensagem que será enviada na rede é definida pelo usuário. Em seguida, a mensagem é dividida em pacotes, e os pacotes são enviados para o roteador mais próximo. Em cada roteador, é realizada uma avaliação para definição da rota que o pacote seguirá na rede. A avaliação consiste em determinar o menor caminho para o destino, para em seguida, verificar com a tabela de roteamento, o congestionamento no menor caminho. Se o menor caminho estiver congestionado, outra rota é determinada utilizando a tabela de roteamento. Os pacotes são transmitidos entre os roteadores até o destino, onde a mensagem original é recriada.

Cada pacote terá as seguintes informações:

- endereço da origem;
- endereço do destino;

- tamanho;
- tempo de cada um dos atrasos;
- número da ordenação do pacote.

Cada roteador da rede terá os seguintes dados:

- tabela de roteamento;
- fila de pacotes.

Desde a divisão da mensagem na origem até a reconstrução da mensagem no destino, deve-se adicionar os seguintes tempos de atrasos:

- atraso de processamento nodal;
- atraso de fila;
- atraso de transmissão;
- atraso de propagação.

O tempo de cada um dos quatro atrasos é determinado pelo usuário.

As funcionalidades do programa são:

1. Criação da mensagem para envio na rede;
2. Divisão da mensagem em pacotes;
3. Recebimento dos pacotes no destino;
4. Recriação da mensagem original no destino;
5. Determinação da menor rota para o destino;
6. Obtenção do caminho menos congestionado;
7. Destaque do menor caminho na visualização;
8. Exibição da tabela de roteamento ao selecionar um roteador. A exibição da tabela de roteamento, não deve atrapalhar a visualização das ligações dos roteadores da rede;
9. Apresentação do tempo de cada um e da soma de todos os atrasos de cada pacote, ao selecionar um pacote;
10. Pausa/retomada da simulação;
11. Visualização da rede formada pelos dispositivos: origem, destino, roteadores. Além de linhas ligando os dispositivos que estão conectados entre si.

O grupo deve elaborar os seguintes diagramas UML do sistema:

- classe;
- pacote;
- caso de uso;
- sequência;
- máquina de estados ou transição de estados.

Em seguida, implementar (ou reestruturar) o código do sistema utilizando os princípios SOLID da orientação a objetos.

Caso o sistema seja implementado usando programação funcional, a equipe não precisa implementar a funcionalidade 10.

## Metodologia

---

- Definição dos requisitos do programa;
- Elaboração dos diagramas UML;
- Criação do protótipo;
- Testes com usuários;
- Ajustes no protótipo;
- Correções de erros e problemas.

## Critérios do Produto

---

Ao final da disciplina a nota atribuída ao projeto será de acordo com as funcionalidades concluídas e os seguintes itens:

1. Diagramas UML;
2. Funcionalidades implementadas;
3. Interface e interação do programa;
4. Utilização dos princípios SOLID;

## Cronograma

---

[Ver 13º artigo das regras gerais](#)

- Art. 1º** – Qualquer linguagem de programação, framework, biblioteca ou plugin são permitidos para desenvolvimento do projeto, desde que não descumpram nenhuma regra específica do projeto do grupo.
- Art. 2º** – Cada grupo terá no máximo quatro integrantes. Se um aluno, dois ou três alunos restarem sem grupos, o(s) aluno(s) podem fazer o trabalho sozinho ou se organizarem para formarem um grupo.
- Art. 3º** – Cada grupo escolherá um dos projetos do documento integrador. O grupo postará em um fórum no SIGAA, o **nome completo de TODOS OS INTEGRANTES** e o projeto escolhido, seguindo o exemplo abaixo:

---

Integrantes: Pedro Henrique Silva Amorim, Antônio Jardim Silva Pedrosa e Valeria Batista Ferreira  
Projeto 2

---

Caso o nome completo dos integrantes não seja informado, o grupo perderá 0.5 ponto na nota final da disciplina.

- Art. 4º** – É aconselhado e incentivado que os grupos da disciplina de Projeto Integrador I se mantenham com os mesmos integrantes e continuem o mesmo projeto, a fim de diminuir o esforço para conclusão das atividades de Projeto Integrador II.
- Art. 5º** – Não existe um prazo máximo para inclusão dos integrantes do grupo no fórum do SIGAA, mas quanto antes o grupo seja formado, maiores são as chances de conclusão do projeto.
- Art. 6º** – Os arquivos necessários para conclusão da atividade devem ser armazenados em um sistema de controle de versão como GitHub, Bitbucket, etc. O acesso aos arquivos do projeto deve ser permitido para os professores da disciplina, para análise do código.
- Art. 7º** – O link para acessar o sistema Web ou a última versão do arquivo executável do projeto devem estar no repositório do controle de versão usado pelo grupo, na data de entrega do marco.
- Art. 8º** – Problemas com os arquivos do trabalho como perda, problemas no computador, no pendrive, em algum mídia de armazenamento ou posse dos arquivos estar com apenas um dos membros, não será motivo para evitar penalização por atraso na entrega da atividade. O grupo é responsável por manter cópias dos arquivos para evitar a perda ou impossibilidade de entrega.
- Art. 9º** – Nos Marcos I e II, os grupos obrigatoriamente apresentarão o andamento das atividades do seu projeto, e os marcos não contarão para nota da disciplina.
- §1º A não apresentação do grupo nos Marcos I ou II ocasionará reprovação do grupo.
- §2º Ausência de comprovações de atividades ou ações no desenvolvimento do projeto do grupo nos Marcos I ou II causará reprovação do grupo.
- Art. 10º** – No Marco III o grupo apresentará a versão final do seu projeto. Para ser aprovado o grupo

deve ter cumprido os critérios de avaliação do projeto, além de principalmente apresentar um sistema completo, ou seja, um sistema faltando apenas alguns poucos detalhes ou pequenas partes das funcionalidades do sistema, ao critério do professor da disciplina.

§1º Se alguma das disciplinas envolvidas no projeto estiver com o conteúdo atrasado, as funcionalidades relacionadas a disciplina poderão ser removidas ou reduzidas, ao critério do professor.

**Art. 11º** – Caso a versão final não seja aprovada no Marco III, o grupo estará de Prova Final e tem até o último dia da semana de provas finais (18/02) para apresentação da versão atualizada do seu sistema.

**Art. 12º** – A data da apresentação dos Marcos pode ser anterior ou até 2 dias úteis depois da data especificada no cronograma do [artigo 13º](#), em dia e horário a combinar com o professor. Assim, mesmo grupos com alunos matriculados com choque de horário, podem marcar a apresentação para um horário sem choque com outras disciplinas.

**Art. 13º** – O cronograma da disciplina é:

Cronograma	
5 / 11	Marco I
10 / 12	Marco II
4 / 2	Marco III
18 / 2	Apresentação Prova Final

**Art. 14º** – Caso uma ou mais disciplinas não estejam dentre as aprovadas ou matriculadas considerando os integrantes do grupo, as atividades do projeto relacionadas a disciplina não deverão ser feitas pelo grupo.

**Art. 15º** – O nome dos componentes do sistema como arquivos, classes, atributos, etc; devem estar de acordo com os dados que serão armazenados ou organizados. Uma variável que armazena a idade deverá se chamar de idade, ou qualquer nome que mostre que a variável esteja armazenando a informação da idade. Caso os nomes dos componentes não estejam de acordo, o grupo poderá ser penalizado com reprovação na disciplina.

**Art. 16º** – O código do programa deve estar indentado, ou seja, com os recuos adequados. Se o código possuir problemas de indentação, o grupo poderá ser reprovado na disciplina.

**Art. 17º** – Elementos dos projetos como código ou interface por exemplo, que sejam iguais de grupos diferentes serão considerados cópias, e os grupos receberão nota zero.

**Art. 18º** – É proibido o uso do comando **goto** em qualquer situação ou quantidade no código. Em caso de utilização do comando **goto** o grupo será reprovado na disciplina.

**Art. 19º** – É proibido que o projeto aborde os seguintes tópicos:

- Referências a drogas ilícitas, tabaco, marcas de produtos e temas questionáveis (ao critério do professor);
- Referência ou apologia a violência ou crimes (ao critério do professor);
- Material relacionado a atividades ilegais como jogos de azar, empresas, serviços ou produtos que promovem ou estimulam tais jogos;

- d) Material difamatório, obsceno, profano, vulgar, repulsivo, ofensivo ou sugestivo que descreva ou represente funções fisiológicas, enfermidades ou qualquer assunto que uma pessoa de bom senso julgaria socialmente inaceitável (ao critério do professor);
- e) Publicidade para produtos ou sites pornográficos;
- f) Materiais que contenham marcas registradas, conteúdo protegido por direitos autorais ou outras formas de propriedade intelectual sem o consentimento do proprietário, ou qualquer elemento que possa submeter o professor ou a universidade em processos por infração ou apropriação indevida;
- g) Material que possa ofender qualquer grupo, aluno, pessoa, entidade ou produto.

**Art. 20º** – O desconhecimento da existência do trabalho ou datas das atividades do trabalho não serão aceitos como justificativa para não realização ou atraso da atividade. É obrigação do grupo buscar informações no SIGAA, com o professor ou demais alunos sobre atividades da disciplina.

**Art. 21º** – A não compreensão da descrição e realização da atividade não será aceita como justificativa para realização incompleta ou incorreta das atividades do trabalho. É obrigação do aluno sanar dúvidas em relação as atividades.

**Art. 22º** – A descrição dos projetos e suas exigências podem sofrer alterações no decorrer da disciplina, as quais serão comunicadas ao grupo.