

 <b>UnB</b>   CIC	<b>UNIVERSIDADE DE BRASÍLIA</b> <b>CAMPUS DARCY RIBEIRO</b>	
Departamento:	Ciência da Computação	
Disciplina:	Técnicas de Programação 1	CIC0197
Professora:	Roberta Barbosa Oliveira	Valor: 10 pts.

“As grandes ideias surgem da observação dos pequenos detalhes.” (Augusto Cury)

## TRABALHO PRÁTICO

Conceitos envolvidos:

Programação Orientada a Objetos, Tratamento de Exceção, Técnicas de Design  
Herança, Polimorfismo, Modularização, Distribuição e Concorrência

### OBJETIVO DO TRABALHO

O objetivo deste trabalho é aplicar, de forma prática, técnicas de programação baseadas em conceitos de orientação a objetos. Busca-se proporcionar aos alunos a experiência de percorrer etapas de um processo de desenvolvimento de software, desde a definição do domínio e regras de negócio até a implementação final, contemplando os seguintes aspectos fundamentais:

- Modelagem orientada a objetos, aplicando herança, polimorfismo, encapsulamento e abstração;
- Implementação de interfaces gráficas, possibilitando interação efetiva com os usuários
- Aplicação de boas práticas de programação, com separação em camadas e organização modular;
- Persistência de dados e tratamento de exceções, assegurando confiabilidade e robustez do software;
- Colaboração em grupo, com divisão de responsabilidades e integração das partes desenvolvidas.

Assim, o trabalho não apenas reforça os fundamentos teóricos da disciplina, mas também desenvolve competências práticas essenciais para a formação em computação, promovendo a transição do aprendizado conceitual para a aplicação em um software. Com isso, espera-se que os alunos desenvolvam competências técnicas relacionadas à implementação de software orientado a objeto, bem como habilidades de trabalho em equipe, análise, planejamento e apresentação de soluções computacionais.

## **ESPECIFICAÇÃO DO TRABALHO A SER DESENVOLVIDO**

Cada grupo deverá desenvolver um software considerando um dos temas disponibilizado pela professora no Ambiente da Disciplina. O projeto deve ser implementado utilizando a linguagem de programação orientada a objetos trabalhada em aula e aplicando, de forma adequada, as técnicas de programação estudadas na disciplina. Os seguintes conceitos devem contemplados no desenvolvimento:

- Programação Orientada a Objetos: construção do software a partir de classes, objetos e abstrações;
- Tratamento de Exceções: implementação de mecanismos para lidar com erros e garantir a robustez do software;
- Técnicas de Design: aplicação de boas práticas de organização, reutilização e manutenção de código;
- Herança e Polimorfismo: utilização dos princípios de generalização, especialização e flexibilidade no comportamento das classes;
- Modularização: separação em camadas e pacotes, promovendo clareza e independência entre os componentes;
- Distribuição e Concorrência (opcional): implementação de funcionalidades que permitam execução paralela e integração eficiente entre diferentes partes do software.

O trabalho será desenvolvido em quatro etapas progressivas, além da definição inicial do grupo e apresentação final. Cada aluno terá responsabilidades em todas as fases, devendo seguir as instruções fornecidas no ambiente virtual da disciplina e contribuir para a integração das partes do projeto.

## **ENTREGAS E APRESENTAÇÕES**

O processo do desenvolvimento do trabalho contará com cinco entregas e uma apresentação final:

- **Definição do Grupo de Trabalho;**
- **Etapas 1: Estrutura Inicial;**
- **Etapas 2: Design e Modelagem;**
- **Etapas 3: Implementação da Lógica do Software;**
- **Etapas 4: Integração e Navegabilidade;**
- **Apresentação Final do Trabalho.**

Todas as entregas são obrigatórias, inclusive a apresentação final. Para cada etapa deverá ser enviado o documento atualizado (em PDF), adicionando a descrição das atividades realizadas, seguindo o modelo disponibilizado no ambiente virtual, bem como a pasta do projeto contendo a implementação correspondente e todos os arquivos necessários para sua execução. As entregas deverão ser submetidas nas atividades específicas da disciplina, dentro dos prazos definidos no cronograma, e todas são sujeitas à validação pela professora. Ao final, será realizada a apresentação do trabalho, na qual o grupo deverá expor as funcionalidades implementadas, o processo de desenvolvimento e os conceitos aplicados, conforme as orientações e prazos estabelecidos.

## Definição do Grupo de Trabalho

A definição do grupo de trabalho constitui a primeira etapa do projeto. Os grupos deverão ser formados, preferencialmente, por quatro integrantes no mínimo, sendo permitido um quinto integrante quando necessário. Excepcionalmente, o trabalho poderá ser realizado individualmente, mediante solicitação prévia e aprovação da professora, com eventuais adaptações no tema para adequar a carga de atividades.

Todos os integrantes do grupo deverão assumir responsabilidade conjunta pelo desenvolvimento do projeto. Em cada etapa, as tarefas serão divididas entre os membros, de forma que cada um seja responsável por uma parte específica, cabendo ao grupo como um todo a integração das partes desenvolvidas. A composição dos grupos e a identificação de seus integrantes deverão ser registradas no ambiente virtual da disciplina, dentro do prazo estabelecido no cronograma.

### Etapa 1: Estrutura Inicial

O objetivo desta etapa é **configurar o ambiente de desenvolvimento e criar os protótipos das telas principais do software**, possibilitando a navegação básica entre elas, ainda sem implementação das regras de negócio.

Para a configuração do ambiente, deverão ser realizadas atividades como a definição da IDE utilizada pelo grupo para o desenvolvimento da aplicação, a criação do projeto inicial com a devida estruturação dos pacotes proposta e a preparação do repositório de controle de versão (como por exemplo o GitHub), com as configurações necessárias. Opcionalmente, poderão ser incluídas bibliotecas externas, por exemplo pela adição de JARs ou do uso do Maven para gerenciamento de dependências. Além disso, recomenda-se a realização de testes iniciais para assegurar que o projeto esteja executando corretamente em todos os ambientes dos integrantes do grupo.

Em relação aos protótipos funcionais, espera-se a criação das telas iniciais do software de acordo com as especificações do tema do projeto, com a definição de layout gráfico (como textos, campos, botões e tabelas) e a implementação da navegação simples entre elas, sem lógica de negócio ou persistência de dados.

### Etapa 2: Design e Modelagem

O objetivo desta etapa é realizar a **modelagem do domínio**, estruturando-o em termos de classes, atributos, métodos e relacionamentos, além de iniciar a definição dos principais casos de uso. Essa etapa tem como foco a construção da base conceitual e lógica, usando uma ferramenta de modelagem, que sustentará o desenvolvimento posterior do projeto.

No processo de modelagem, deverão ser criadas as classes de acordo com o tema do projeto, aplicando conceitos de orientação a objetos como herança, polimorfismo, encapsulamento e abstração. Além disso, os métodos principais deverão ser definidos, garantindo clareza na responsabilidade de cada classe.

Paralelamente, será iniciado o preparo para os casos de uso do sistema, por meio da criação de controladores que organizam a lógica básica das operações, ainda sem a integração com a interface gráfica ou com a persistência de dados.

### **Etapa 3: Implementação da Lógica do Software**

O objetivo desta etapa é transformar a modelagem definida anteriormente em um sistema funcional, **implementando as regras de negócio**, os **mecanismos de persistência de dados (arquivos)** e o **tratamento de exceções**.

As atividades desta fase envolvem a codificação das regras de negócio estabelecidas para o domínio, aplicando herança, polimorfismo e demais conceitos de orientação a objetos de forma prática. Também deve ser realizada a integração com mecanismos de persistência, garantindo o armazenamento e a recuperação das informações do sistema. Outro ponto essencial é a implementação do tratamento de exceções, assegurando a robustez do software frente a erros e situações inesperadas.

### **Etapa 4: Integração e Navegabilidade**

O objetivo desta etapa é consolidar todas as partes do sistema, **integrando a lógica implementada à interface gráfica** e garantindo a navegabilidade entre as telas. Este é o momento de tornar o software utilizável, com interação completa entre os usuários e as funcionalidades implementadas.

As atividades incluem a programação dos eventos da interface gráfica, conectando botões, menus e formulários às operações da lógica de negócio. Além disso, deve-se assegurar a correta navegação entre telas, com fluxo de uso consistente e intuitivo. A comunicação entre interface, regras de negócio e persistência deve estar plenamente estabelecida, promovendo a integração final do projeto.

### **Apresentação Final do Trabalho**

A apresentação deve demonstrar a **execução do software e suas funcionalidades**, destacando as técnicas de programação utilizadas. O programa apresentado deve corresponder à versão final entregue na tarefa, sem alterações.

Cada grupo terá 20 minutos para a apresentação, seguidos de 15 minutos para arguição. Para grupos com 5 integrantes, a apresentação terá 25 minutos. Todos os integrantes do grupo devem participar da apresentação, dividindo o tempo de forma equilibrada. Durante a arguição, cada integrante deve demonstrar conhecimento sobre todo o trabalho desenvolvido pelo grupo.

Caso algum integrante não possa participar na data marcada, mediante apresentação de documento comprobatório, o grupo deverá apresentar em outra data, previamente acordada com a professora. Nessa situação, o grupo deve entrar em contato com a professora até o dia da apresentação, conforme o cronograma estabelecido.

## MÉTODO DE AVALIAÇÃO

A pontuação máxima do trabalho será de **10 pontos**. Todas as etapas são obrigatórias e o trabalho será avaliado apenas se o grupo cumprir todas as etapas, com entregas e apresentações realizadas nas datas estabelecidas no cronograma.

A avaliação do trabalho considerará três categorias:

- **Relatório final (2 pontos):** Avaliado em grupo, deve apresentar a descrição completa de todas as etapas do trabalho, seguindo a estrutura disponibilizada no ambiente da disciplina;
- **Apresentação final (3 pontos):** Avaliada individualmente, considerando a demonstração da compreensão e assimilação do conteúdo da disciplina;
- **Implementação final (5 pontos):** A nota será composta por 3 pontos individuais para cada integrante e 2 ponto de colaboração em grupo. A avaliação da implementação levará em conta as funcionalidades do software, a correta aplicação do conteúdo disciplinar e a solução proposta, considerando a utilização das técnicas de programação para atender aos requisitos de qualidade de software, como reutilização, manutenibilidade, confiabilidade, flexibilidade, extensibilidade, eficiência e clareza do código.