



**PLANO DE ENSINO**

para o Quadrimestre Suplementar (QS) – Resolução CONSEPE 240/2020

ANO LETIVO	QUADRIMESTRE	TURNO	CAMPUS
2021	Q2	Noturno	Santo André

CÓDIGO	NOME
MCTA018-13SA	Programação Orientada a Objetos
Turma	RECOMENDAÇÃO
NBMCTA018-13SA	Processamento da Informação Programação Estruturada

**EMENTA**

Conceitos básicos: classes, objetos, mensagens, encapsulamento, herança, polimorfismo. Programação orientada a objetos utilizando uma linguagem de programação orientada a objetos (Java). Análise e projeto orientados a objetos. UML. Padrões de projeto de software.

**COMPETÊNCIA**

A disciplina deve permitir que o aluno seja capaz de compreender os conceitos básicos da programação orientada a objetos e aplicar adequadamente os conceitos e técnicas de programação orientada a objetos no projeto e desenvolvimento de sistemas computacionais.

**SOBRE A DISCIPLINA**

A disciplina será conduzida no ambiente virtual de aprendizagem (AVA) Google Classroom:

- <https://classroom.google.com/c/MzQxNjM1NTc3MjQ5>

Curso: **NBMCTA018-13SA - Programação Orientada a Objetos - Prof. Paulo Meirelles - 2021.2**

**Fique atento aos comunicados realizados no Google Classroom.**

## Aulas e Atividades

**Docente:** Paulo Roberto Miranda Meirelles ([paulo.meirelles@ufabc.edu.br](mailto:paulo.meirelles@ufabc.edu.br))

**Horário semanal:** segunda-feira das 21h às 23h | quarta-feira das 19h às 21h

### Plano:

Semana	Horas	Tópicos	Atividades
01 24/05/2021 a 30/05/2021	4h	<ul style="list-style-type: none"><li>● Apresentação da disciplina</li><li>● Software Livre</li><li>● Controle de versão</li><li>● Ambiente de desenvolvimento</li></ul>	<ul style="list-style-type: none"><li>● Assistir vídeo aulas (gravadas)</li><li>● Ler material</li><li>● Quiz para responder</li></ul>
02 31/05/2021 a 06/06/2021	4h	<ul style="list-style-type: none"><li>● Introdução à Orientação à Objetos</li><li>● Classes e objetos</li><li>● Métodos e atributos</li><li>● Membros estáticos e de instância</li></ul>	<ul style="list-style-type: none"><li>● Assistir vídeo aulas (gravadas)</li><li>● Ler material</li><li>● Exercício (Lab) para submissão</li><li>● Quiz para responder</li></ul>
03 07/06/2021 a 13/06/2021	4h	<ul style="list-style-type: none"><li>● Testes automatizados</li><li>● UML (Diagrama de Classes: classe, associação, agregação e composição)</li></ul>	<ul style="list-style-type: none"><li>● Assistir vídeo aulas (gravadas)</li><li>● Ler material</li><li>● Exercício (Lab) para submissão</li><li>● Quiz para responder</li></ul>
04 14/06/2021 a 20/06/2021	4h	<ul style="list-style-type: none"><li>● Herança</li><li>● Modificadores de acesso</li><li>● Construtores</li><li>● Encapsulamento</li><li>● Métodos de acesso</li><li>● Classes abstratas</li><li>● UML (Diagrama de Classes: herança)</li></ul>	<ul style="list-style-type: none"><li>● Assistir vídeo aulas (gravadas)</li><li>● Ler material</li><li>● Exercício (Lab) para submissão</li><li>● Quiz para responder</li></ul>
05 21/06/2021 a 27/06/2021	4h	<ul style="list-style-type: none"><li>● Polimorfismo</li><li>● Interface</li><li>● UML (Diagrama de Classes: interface)</li></ul>	<ul style="list-style-type: none"><li>● Assistir vídeo aulas (gravadas)</li><li>● Ler material</li><li>● Exercício (Lab) para submissão</li><li>● Quiz para responder</li></ul>

06	4h	<ul style="list-style-type: none"> <li>• Orientação a objetos em Python</li> <li>• Definição de projeto da disciplina</li> </ul>	<ul style="list-style-type: none"> <li>• Assistir vídeo aulas (gravadas)</li> <li>• Ler material</li> <li>• Exercício (Lab) para submissão</li> <li>• Enviar proposta de projeto da disciplina</li> </ul>
28/06/2021 a 04/07/2021			
07	4h	<ul style="list-style-type: none"> <li>• Tipos genéricos</li> <li>• Coleções</li> </ul>	<ul style="list-style-type: none"> <li>• Assistir vídeo aulas (gravadas)</li> <li>• Ler material</li> <li>• Exercício (Lab) para submissão</li> <li>• Quiz para responder</li> </ul>
05/07/2021 a 11/07/2021			
08	4h	<ul style="list-style-type: none"> <li>• Tratamento de exceções</li> </ul>	<ul style="list-style-type: none"> <li>• Assistir vídeo aulas (gravadas)</li> <li>• Ler material</li> <li>• Exercício (Lab) para submissão</li> <li>• Quiz para responder</li> </ul>
12/07/2021 a 18/07/2021			
09	4h	<ul style="list-style-type: none"> <li>• Análise e projeto orientados a objetos</li> <li>• Andamento do projeto da disciplina</li> </ul>	<ul style="list-style-type: none"> <li>• Assistir vídeo aulas (gravadas)</li> <li>• Ler material</li> <li>• Exercício (Lab) para submissão</li> <li>• Relatório parcial do projeto (e vídeo ou apresentação ao vivo)</li> </ul>
19/07/2021 a 25/07/2021			
10	4h	<ul style="list-style-type: none"> <li>• Padrões de Projeto</li> </ul>	<ul style="list-style-type: none"> <li>• Assistir vídeo aulas (gravadas)</li> <li>• Ler material</li> <li>• Exercício (Lab) para submissão</li> <li>• Quiz para responder</li> </ul>
26/07/2021 a 01/08/2021			
11	4h	<ul style="list-style-type: none"> <li>• Padrões de Projeto</li> </ul>	<ul style="list-style-type: none"> <li>• Assistir vídeo aulas (gravadas)</li> <li>• Ler material</li> <li>• Exercício (Lab) para submissão</li> <li>• Quiz para responder</li> </ul>
02/08/2021 a 08/08/2021			

12	4h	<ul style="list-style-type: none"> <li>• Entrega/apresentação dos projetos da disciplina</li> <li>• Atividade(s) substitutiva(s)</li> </ul>	<ul style="list-style-type: none"> <li>• Relatório final do projeto (e vídeo ou apresentação ao vivo)</li> </ul>
09/08/2021 a 14/08/2021			

Onde:

- **Horas:** a coluna horas na tabela é a soma de T+P. Contudo, será necessário tempo adicional de dedicação para estudo, conforme indicado no T-P-I da disciplina.
- **Assistir vídeo aulas (gravadas):** serão disponibilizados links para vídeo aulas sobre o conteúdo da disciplina (até antes do início do horário de cada aula), no Google Classroom. É necessário um navegador web com capacidade de executar vídeos armazenados no Google Drive. Durante as vídeo aulas, além de apresentar o conteúdo, podem ser realizados exemplos práticos de implementação dos conceitos.
- **Ler material:** será disponibilizado material para leitura (por exemplo, slides da aula em PDF ou outro material sobre o conteúdo da disciplina). É recomendável ler o material, além de assistir às vídeo aulas.
- **Exercícios (Lab) para submissão:** serão disponibilizados exercícios ao longo do quadrimestre. Todos os exercícios são individuais (devem ser resolvidos pelo próprio discente). Os exercícios deverão ser submetidos via Github Classroom (teremos uma vídeo aula específica sobre controle de versão e o uso do Git, e o/a discente deve criar um usuário no Github).
  - Cada exercício pode ter pontuação máxima e prazo de entrega diferentes (essas informações estarão disponíveis no enunciado).
  - Os exercícios serão anunciados em Java poderão ter como modelo um código em Java, podendo estar acompanhado dos testes automatizados. Aqueles discentes que desejarem, podem submeter a solução dos exercícios em Python 3, de forma que os testes automatizados devem ser escritos pelo/a discente, baseados nos testes disponibilizados pelo professor em Java.
  - Os repositórios com os códigos usados da disciplina estarão disponíveis em <https://github.com/UFABC-OO>.
- **Quiz para responder:** ao longo do quadrimestre, serão disponibilizados questionários que tem como objetivo fazer uma avaliação dos conhecimentos teóricos dos alunos sobre os assuntos tratados durante uma determinada semana. Todos os questionários devem ser respondidos individualmente (devem ser resolvidos pelo próprio discente). Os questionários estarão integrados às atividades disponíveis em nossa sala no Google Classroom e deverão ser respondidos e submetidos exclusivamente via Google Classroom. Cada *quiz* pode ter pontuação máxima e prazo de entrega diferentes (essas informações estarão disponíveis no enunciado).

### Avaliação dos exercícios (Lab) e questionários (Quiz):

1. Poderá ser utilizado um sistema de correção automática para os exercícios (Lab). A princípio, a correção automática (caso esteja disponível) aplicará o seguinte critério:
  - 1.1. Exercícios com pontuação máxima até 1 ponto: é necessário passar em todos os casos de teste para obter 1 ponto. Caso contrário, ficará com zero.
  - 1.2. Exercícios com pontuação máxima superior a 1 ponto: se não passar em todos os casos de teste, estes exercícios podem receber pontuação parcial, limitando-se a 60% da pontuação do exercício.

- 1.3. Alteração manual da pontuação automática: em alguns casos, o professor poderá atribuir nota manualmente nos exercícios e alterar a nota atribuída pelo sistema de correção automática também (por exemplo, devido a pontos que o corretor automático não pode identificar ou problemas/erros na avaliação).
2. Todos os questionários (Quiz) terão correção automática, com pontuação proporcional aos acertos das respostas.
3. Todos os exercícios (Lab) e questionários (Quiz) serão disponibilizados na sala do Google Classroom com antecedência de, no mínimo, 72 horas.

## Avaliação do período letivo regular

A média final será composto da seguinte forma:

$$MF = (Nota\_Projeto*5 + Nota\_Lab*4 + Nota\_Quiz*1)/10$$

1. **Questionários (Quiz) sobre os conteúdos das aulas e Exercícios de programação (Labs):** cada uma dessas tarefas pode ter pontuação máxima diferente assim como prazos de entrega diferentes. As tarefas não entregues no prazo receberão nota zero. Todas as tarefas são individuais (devem ser resolvidas pelo próprio discente). Para os exercícios de programação, uma ferramenta para identificação de submissões com alta similaridade pode ser usada.
2. **Projeto** que demonstre a aplicação adequada dos conceitos e técnicas de programação orientada a objetos no projeto e desenvolvimento do sistema a ser proposto pelo/a discente até a sexta semana do quadrimestre (os critérios serão descritos em um documento a ser disponibilizado na primeira semana de aula).
  - 2.1. O projeto deve ser desenvolvido, preferencialmente, de forma individual. Em casos que os/as discentes apresentem um projeto com mais requisitos e complexidade, explicando a divisão das tarefas, o projeto pode ser desenvolvido em dupla.
  - 2.2. Alternativamente, até a sexta semana do quadrimestre, o aluno pode escolher ter como projeto de disciplina um **projeto de software livre**, de forma que faça contribuições relevantes para o projeto escolhido (os critérios e exemplos estarão disponíveis em um documento complementar e um vídeo será gravado sobre o assunto para melhor orientar os alunos sobre esta alternativa).

**Observação:** Para ser aprovado, o/a discente deve ter no mínimo nota 5,0 no projeto. Caso a nota do projeto da disciplina for inferior a 5, as notas dos Lab e Quiz não serão contabilizadas, de maneira que a média final será:

$$MF = (Nota\_Projeto*5)/10$$

## Alternativa para alunos com muita experiência de programação:

Durante a primeira semana de aula, o professor disponibilizará um questionário para que o/as discentes respondam voluntariamente. Aqueles que indicarem que têm boa experiência com alguma linguagem de programação e que conhecem o processo de desenvolvimento de um software livre, poderão ter sua média final composta por contribuições semanais para um projeto de software livre:

### MF = Nota\_Projeto

De forma que o/a discente deve:

- A. Fazer um post semanal no mural da disciplina no Google Classroom (ou compartilhar link de um post semanal em blog pessoal) sobre a contribuição feita e o link para o pull/merge-request no repositório (ou do email com o patch enviado para a lista) do projeto de software livre que está contribuindo.
- B. Ao final, entregar o relatório final descrevendo as contribuições, explicar os conceitos e técnicas de orientação a objetos que foram trabalhados e a importância das contribuições enviadas para o projeto de software livre escolhido.
- C. Começar a contribuir com o projeto a partir da segunda semana do quadrimestre até a décima segunda semana. Caso o rendimento não esteja sendo satisfatório, ao professor avaliar as postagens semanais, o/a aluno/a poderá refazer o seu plano e também deverá fazer as atividades previstas na disciplina (avaliando com o professor a continuidade das contribuições ao projeto de software livre escolhido como projeto de disciplina).

## Conceito final

A partir da média final (MF), o conceito final (CF) será obtido por meio da seguinte forma:

- **A** =  $MF \geq 9$
- **B** =  $9 > MF \geq 7,5$
- **C** =  $7,5 > MF \geq 6$
- **D** =  $6 > MF \geq 5$
- **F** =  $MF < 5$
- **O** = reprovação por ausência de avaliação

## Atividade Substitutiva (SUB)

Os/as discentes que não puderem realizar algum dos Lab ou Quiz dentro do prazo, em razão das justificativas mencionadas na Resolução CONSEPE 227 de 23 de abril de 2018, poderão realizar a entrega após o prazo como forma de avaliação substitutiva (para até no máximo de duas tarefas não entregues no prazo).

- A solicitação da avaliação substitutiva deve ser realizada pelo/a discente antes do dia 04/08/2021 (quarta-feira) para viabilizar a realização dentro do período do quadrimestre.

## Mecanismo de Recuperação (REC)

Estarão habilitados para o mecanismo de recuperação, os alunos que obtiverem conceito final D ou F, obedecendo às regras indicadas na Resolução CONSEPE no. 182 de 23 de outubro de 2014.

Como mecanismo de recuperação, os elegíveis, caso desejem, poderão melhorar o seus projetos de

disciplina, até o dia 11/08 (quarta-feira), entregando um novo relatório, para compor o Conceito Final Recuperado. No caso de quem optou por contribuições em um projeto de software livre, poderá fazer uma nova rodada de contribuições, também enviando um novo relatório final.

Após a substituição da nota do projeto, a média final com recuperação (MR) será obtida com a mesma fórmula usada para a média final. Com a média final com recuperação, o conceito final com recuperação (CFR) será no máximo C, obtido por meio da seguinte forma:

- **C** =  $MR \geq 6$
- **D** =  $6 > MR \geq 5$
- **F** =  $MR < 5$

### **Análise contra Plágio**

Caso seja identificado plágio em qualquer exercício de programação entregue ou mesmo no projeto da disciplina, será atribuído Conceito Final igual a **O** (reprovação por ausência de avaliação) para todos os envolvidos. Algumas consequências que o aluno poderá sofrer estão descritas no Código de Honra anexo a este plano.

### **Ambiente de desenvolvimento e ferramentas de apoio**

- Computador com acesso à internet
- Caixa de som ou fone de ouvido
- Navegador web compatível com o Google Classroom e Google Meet
  - deve permitir a execução de vídeos a partir do Google classroom e Drive
- Java Development Kit
  - recomendado o Open Java Development Kit (OpenJDK) 11
- Python 3
- IDE Eclipse (recomendando) ou Editor de sua preferência
- Recomenda-se o uso do sistema operacional GNU/Linux (Debian ou Ubuntu, por exemplo)

### **Atividades de apoio e atendimentos**

As aulas gravadas serão disponibilizadas até o horário previsto para o início da aula do dia, ou seja, segundas, antes das 21h, e quartas, antes das 19h. Recomenda-se que os alunos assistam as vídeo aulas durante a primeira hora prevista e possam discutir suas dúvidas com o professor em seguida.

O professor estará disponível nos seguintes dias/horários para atendimento aos alunos (dúvidas sobre o conteúdo da disciplina):

1. Segundas: 21h às 23h (Google Meet)
2. Quartas: 19h às 21h (Google Meet)
3. Quintas: 15h (Google Meet)

Link do Google Meet: <https://meet.google.com/lookup/fb74psu4rm>

## Bibliografia

### Básica:

1. BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro, RJ: Campus, 2005.
2. DEITEL, H. M.; DEITEL, P. J. Java: como programar. 6. ed. Porto Alegre, RS: Bookman, 2005.
3. GUEDES, G. T. A. UML 2: uma abordagem prática. São Paulo, SP: Novatec, 2009.

### Bibliografia complementar:

1. BARNES, D. J.; KOLLING, M. Programação orientada a objetos com Java. 4. ed. São Paulo, SP: Pearson Prentice Hall, 2009.
2. BRUEGGE, B.; DUTOIT, A. H. Object-oriented software engineering: using UML, patterns, and Java. 2. ed. Upper Saddle River, NJ: Prentice Hall, 2003.
3. FLANAGAN, D. Java: o guia essencial. 5. ed. Porto Alegre, RS: Bookman, 2006. • FOWLER, M. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre, RS: Bookman, 2005.
4. LARMAN, C. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre, RS: Bookman, 2007.

## Material de Apoio

1. Introdução a Ciência da Computação com Java e Orientação a Objetos
  - o <http://ccsl.ime.usp.br/files/publications/files/2008/intro-java-cc.pdf>
2. Java e Orientação a Objetos - Caelum Apostila FJ11
  - o <https://www.caelum.com.br/download/caelum-java-objetos-fj11.pdf>

## Agradecimento

A organização, formato e parte do conteúdo deste plano de ensino foram baseados nos planos de ensino dos professores Paulo Henrique Pisani e Monael Pinheiro Ribeiro, ambos do CMCC/UFABC.

## Revisões

Em 27/05/2021, dois ajustes foram feitos neste plano de ensino:

1. Todas as atividades avaliativas serão disponibilizadas com antecedência de, no mínimo, 72 horas
2. Atendimento, além dos horários das aulas, quintas, 15h. Todos os atendimentos via Google Meet.