



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
ESCOLA POLITÉCNICA

CURSO DE ENGENHARIA DE COMPUTAÇÃO  
PROGRAMAÇÃO ORIENTADA A OBJETOS

Quando os ventos da mudança sopram,  
algumas pessoas constroem barreiras,  
outras, moinhos de vento.  
Érico Veríssimo

## Form de upload do Trabalho 2 – Controle de Biblioteca pessoal

### Equipe

---

Estudante 1: João Gabriel Klein Lopes

Estudante 2: Mateus Ferro Antunes de Oliveira

Estudante 3: Milena Heloisa de Amorim Silvério

### Link para o vídeo

---

Importante:

- O vídeo precisa estar **público**
- Pode estar **não listado**: não aparece no canal, apenas quem tem o link acessa.
- Faça um teste com o link em um navegador não logado no YouTube.

Link para o Vídeo: [https://youtu.be/gm9JSa\\_BrRU](https://youtu.be/gm9JSa_BrRU)

### Autoavaliação

---

Para cada um dos itens da tabela da próxima página, **escolha marcando um X aquele que mais se adapta ao resultado de sua implementação.**

A soma total de todos os itens resulta em 11 pontos, anote abaixo a soma alcançada.

A soma dos pontos é: 11

*Comentários a respeito da implementação (o que gostaria de destacar de interessante, algo extra que foi feito e merece uma observação, alguma dificuldade que foi resolvida/aprendida etc.)*

O trabalho foi um grande desafio para todos do grupo. Isso nos ajudou ainda mais para entender a programação orientada à objetos em JAVA.

Todos do grupo participaram e auxiliaram na elaboração do projeto.

Ficamos animados em desenvolver o programa, pois nos desafiou a aprender e fixar os conhecimentos passados pelo professor durante o semestre.

## Grade de autoavaliação do trabalho em relação a especificação

<b>Implementa as classes X e Y empregando herança</b> (X) 1.0 – Implementa as duas classes livres para especializar outros tipos de itens (subclasses de Item) ou altera a hierarquia, porém introduz duas classes novas conforme escolha da equipe mantendo algum relacionamento de herança ( ) 0.6 – Tem duas classes, mas não usa herança ou apenas uma classe com herança (implementa parcialmente o solicitado) ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, tem classes extras, mas sem herança). ( ) 0.0 – Não há esta possibilidade (classes não foram implementadas)
<b>Funcionalidade cadastrar item</b> (X) 1.0 – Opera corretamente, lendo os dados necessários conforme a categoria do item, bem como usando idItem com código adequado gerado automaticamente e incluindo em na Biblioteca ( ) 0.6 – Le os dados corretamente, porém não utiliza o idItem conforme especificado ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo). ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade cadastrar amigo</b> (X) 1.0 – Opera corretamente, criando um objeto Amigo e incluindo na ListaAmigos ( ) 0.6 – Opera com alguma funcionalidade diferente (exemplo: idAmigo não é gerado automaticamente) ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo). ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade empréstimos de item</b> (X) 1.0 – Opera corretamente perguntando o código do item e o código do amigo efetuando o empréstimo adicionando com os atributos corretos (incluindo dataEmprestimo) na ListaEmprestimos ( ) 0.6 – Funciona, porém de outra forma levemente diferente, por exemplo, não empregando os códigos como descrito na especificação ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo) ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade devolver item</b> (X) 1.0 – Opera corretamente perguntando o código do item e efetuando a devolução (incluindo dataDevolução) do objeto corretor na ListaEmprestimos ( ) 0.6 – Funciona, porém de outra forma levemente diferente, por exemplo, não usando o código correto ou então com problemas na data ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo) ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade empréstimos atuais</b> (X) 1.0 – Opera corretamente empregando o enum dispItem para verificar os itens emprestados ( ) 0.6 – Funciona, porém de outra forma levemente diferente, por exemplo, não empregando o enum solicitado ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo) ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade histórico de empréstimos</b> (X) 1.0 – Opera corretamente conforme especificado, a partir do código de um item lista todas as movimentações que já ocorreram com este item da biblioteca ( ) 0.6 – Funciona, porém de outra forma levemente diferente, por exemplo, não mostra todos os atributos ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo) ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade listar biblioteca</b> (X) 1.0 – Opera corretamente conforme especificado, listando todos os itens da biblioteca em ordem de alfabética de título apresentando seu nome e sua última situação ( ) 0.6 – Funciona, porém de outra forma levemente diferente, por exemplo, não mostra todos os atributos ou não ordena ( ) 0.4 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo) ( ) 0.0 – Não há esta possibilidade
<b>Funcionalidade alterar estado</b> (X) 1.0 – Opera corretamente conforme especificado, permitindo mudar o estado de um item a partir do seu código ( ) 0.5 – Opera diferente do especificado (funciona de outro jeito do esperado, foi adaptado para outro modelo) ( ) 0.0 – Não há esta possibilidade
<b>Persistência de objetos em disco (serialização das classes)</b> (X) 1.0 – Salva em disco os dados conforme as classes vão sendo manipuladas e ao abrir o programa procurar pelo arquivo lendo os dados serializados (ou seja, garante a persistência dos dados do sistema Biblioteca) ( ) 0.5 – Efetua algumas operações, apenas salva ou salva parcialmente ( ) 0.0 – Não há esta possibilidade
<b>Código está documentado com tags JavaDoc</b> (X) 1.0 – Sim, todas as classes e métodos estão corretamente documentados com tags JavaDoc. ( ) 0.5 – As classes estão parcialmente documentadas (faltam algumas classes ou métodos), ou a documentação foi feita apenas com comentários tradicionais, sem uso de tags JavaDoc ( ) 0.0 – Não há documentação com comentários ou tags