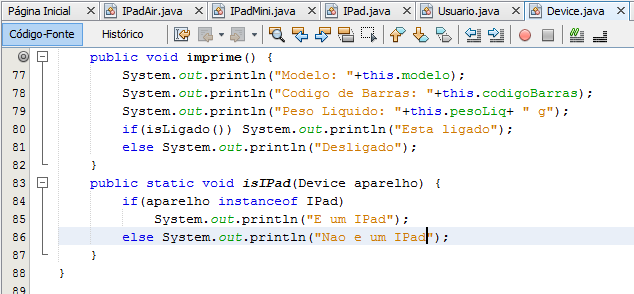
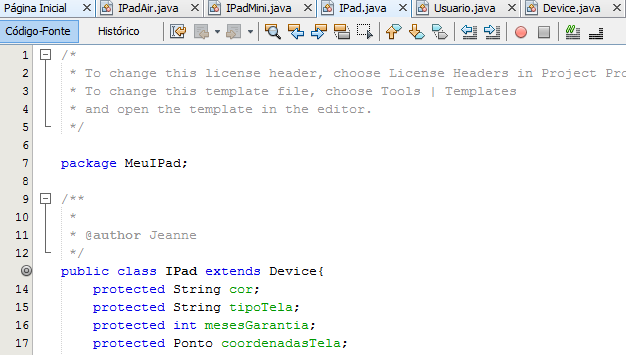
***Importante***: Para item abaixo deve ser copiado trechos do código que cumprem o requisito e explicado, se não for aparente, o porquê o requisito é cumprido. Sejam bem explícitos. Deve ser indicado também o arquivo da classe em que está o trecho do código. Eu avaliarei o código do Github a partir desse documento para confirmá-lo e também para detectar possíveis erros. **Quem não seguir o que está indicado aqui, não terá o projeto avaliado e perderá a atividade.**

**IPad**

Implementação de uma classe IPad em java

Dentre as outras funcionalidades propostas por voce deve um codigo usado para diferenciar um IPad de outros devices. Deve tambem armazenar as coordenadas x e y apontadas na tela pelo usuario.

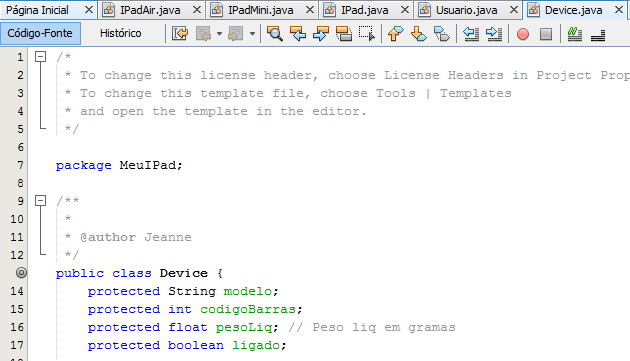




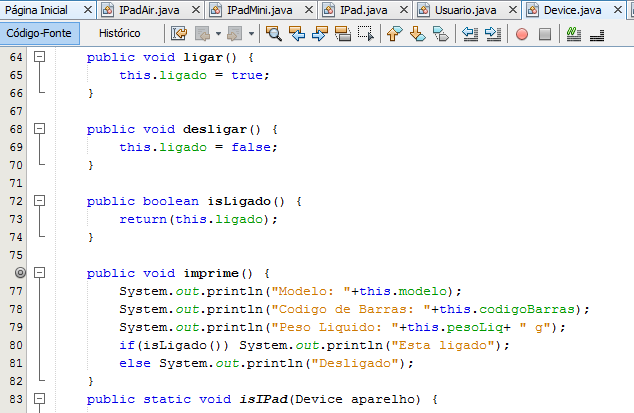
**Requisitos de implementação**

***Todos os atributos e funções membros devem estar relacionados a classe***

1. Pelo menos 4 atributos



1. Pelo menos 4 funções membros sem incluir get e set

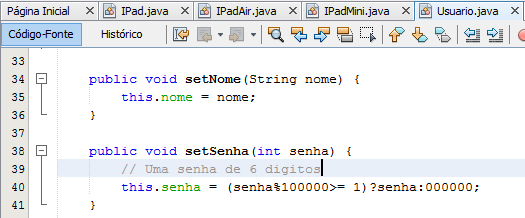
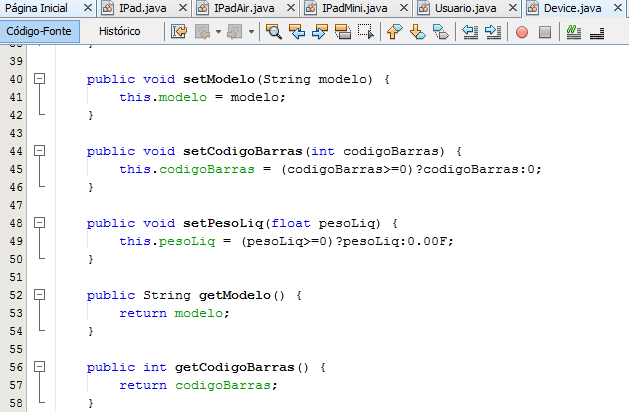


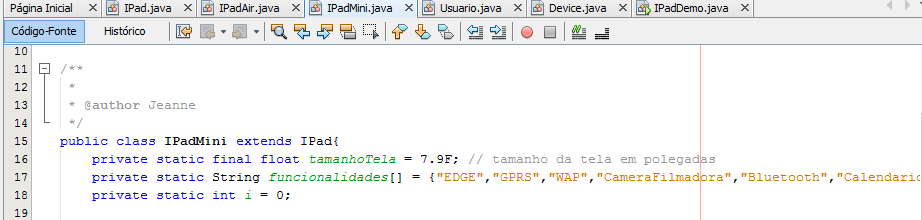
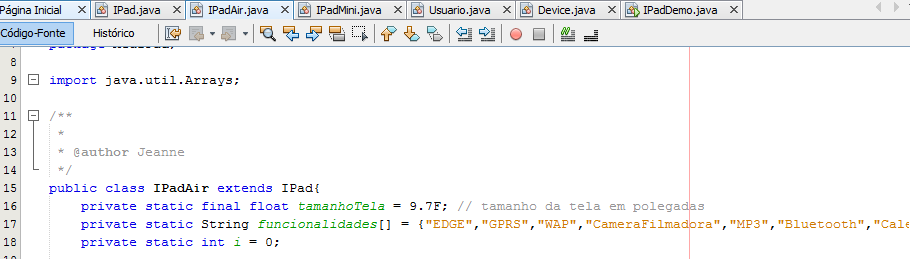
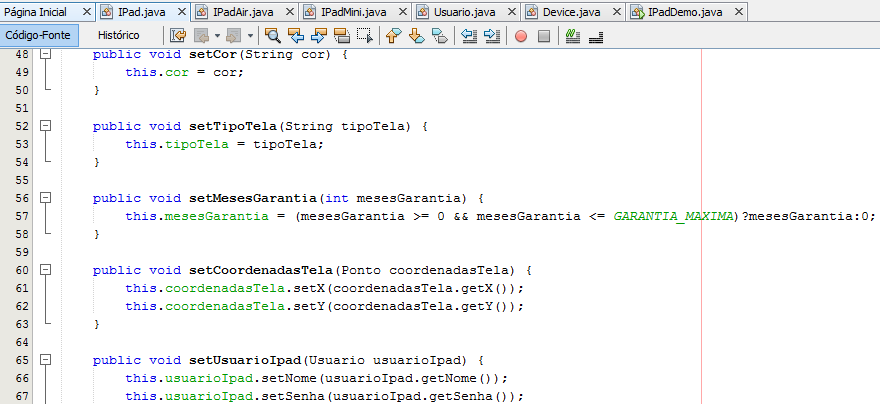
**Requisitos de implementação**

1. Cinco classes: uma superclasse e duas subclasses, e duas classes relacionadas ao projeto

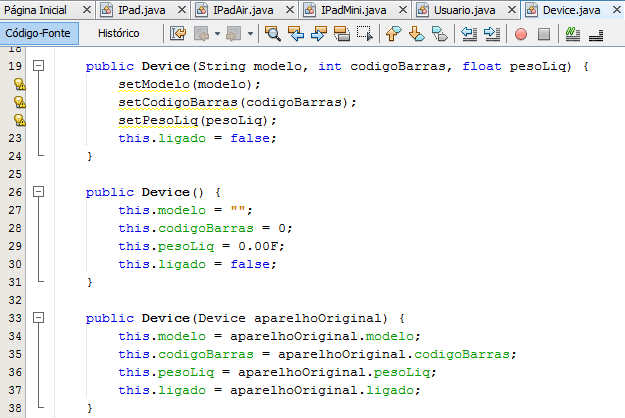


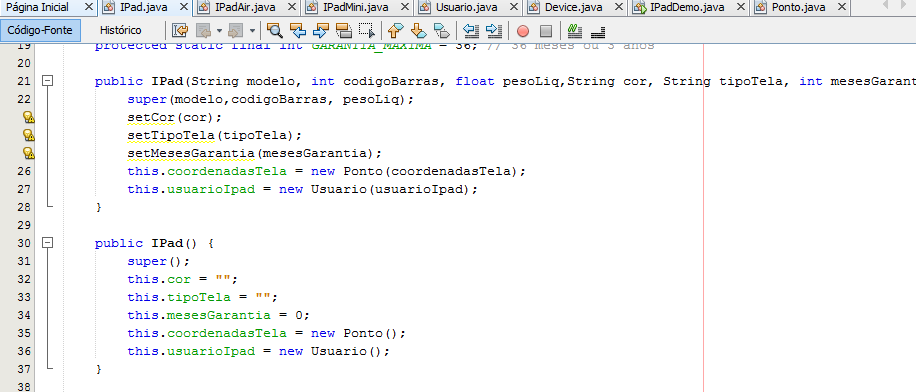
1. Sempre usar o super para o máximo de reaproveitamento de código
2. Diagrama de classes (obrigatório salvar também o png do diagrama no gitHub)
3. Todos os atributos devem ser inicializados. Fez validação de dados?

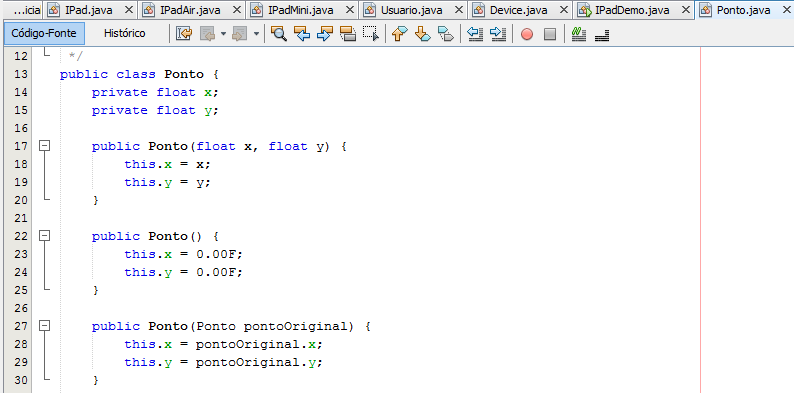




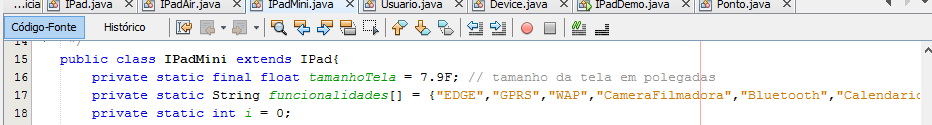
1. Dois construtores –ok Device.java,IPad.java,IPadAir.java,IPadMini.java

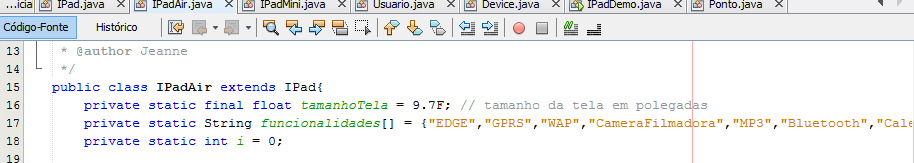




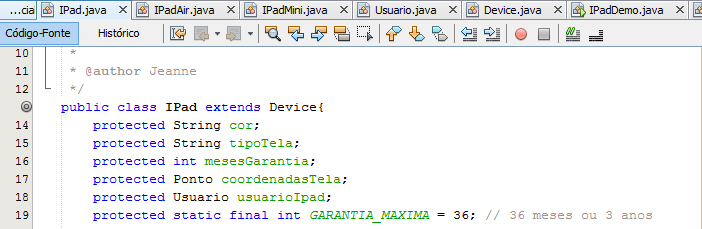


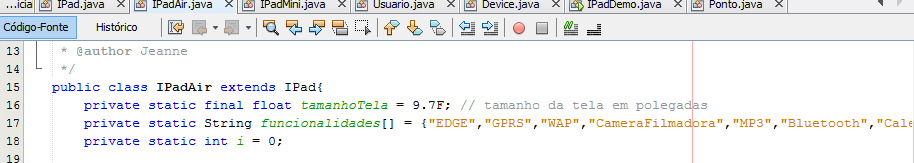
1. Um atributo static. – ok ,IPadAir.java,IPadMini.java

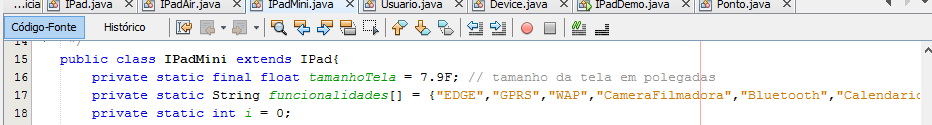




1. Um atributo const static –ok IPad.java,IPadAir.java,IPadMini.java

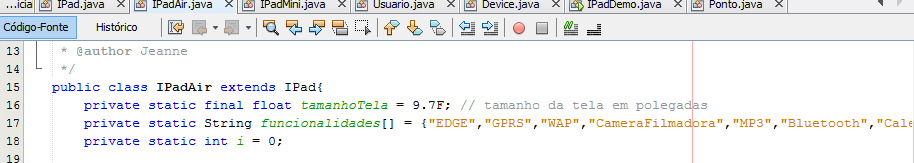


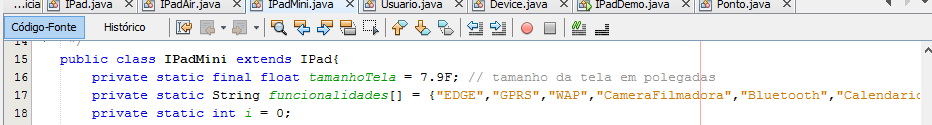




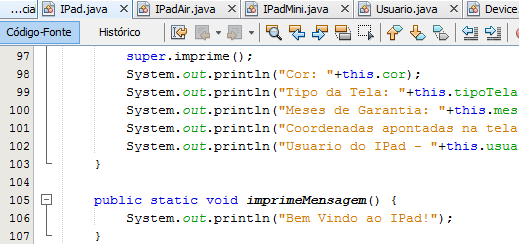
*Correta modelagem dos statics?*

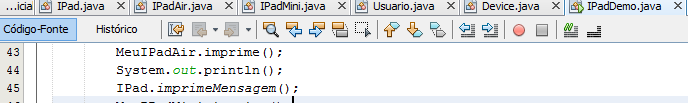
1. Um array – ok IPadAir.java,IPadMini.java



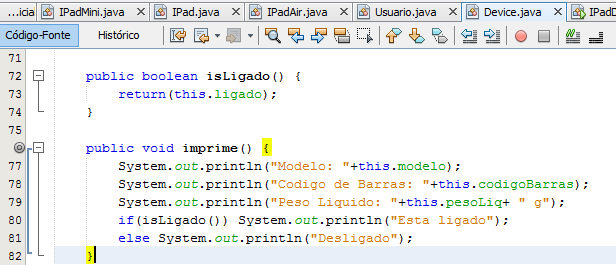


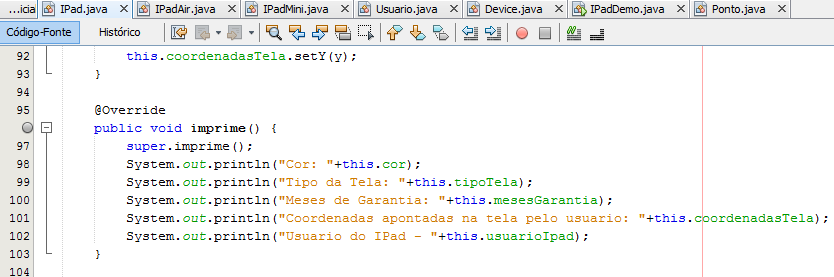
1. Método static – deve ser chamado no main –ok

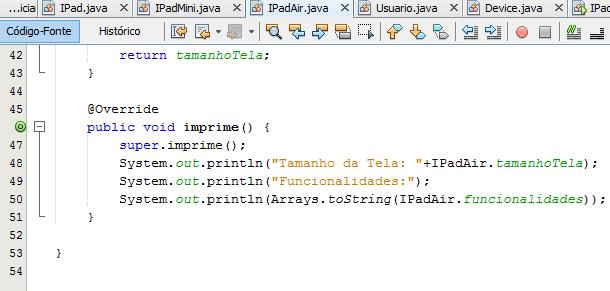


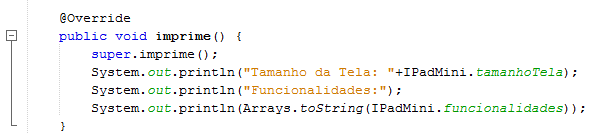


1. Sobrescrita de método: chamar dentro do método da classe. Usar o @override –ok IPad.java, IPadAir.java,IPadMini.java

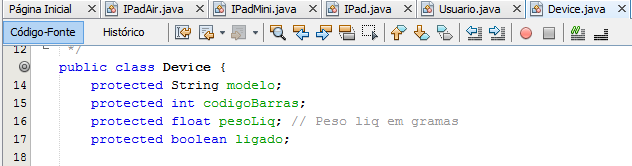


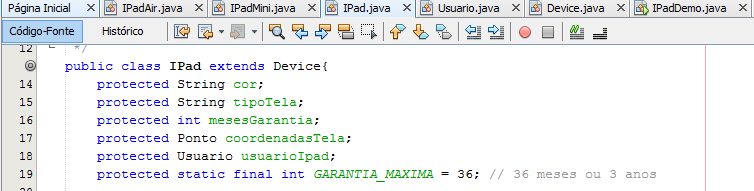




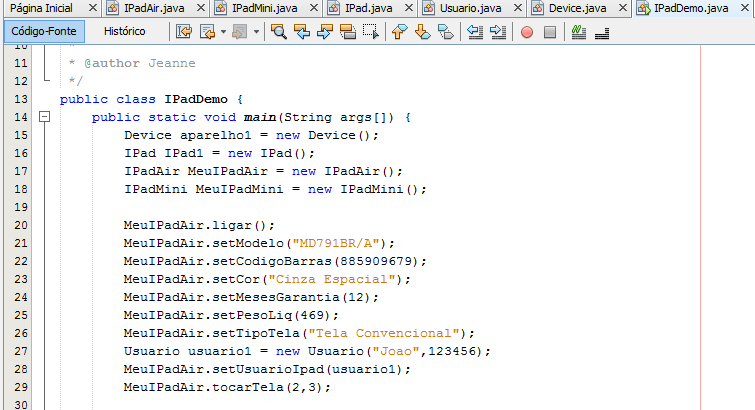


1. Usar Protected acessando diretamente os atributos na classe derivada –ok Device.java,IPad.java





1. Usar suas classes no main –ok



Opcionais que garantem pontos extras:

Trabalhar com ENUM

Trabalhar com pacotes