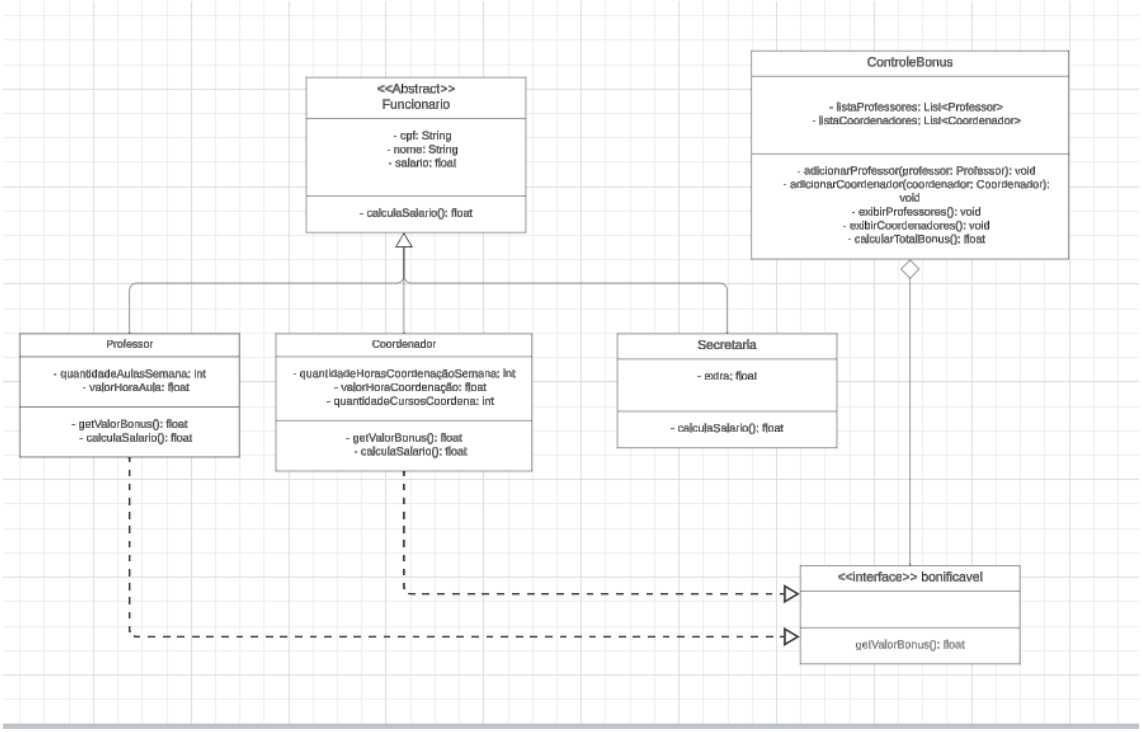
Nome: Diego Hessel Motta Verdi RA: 01231048S



1) Como fazer para que as classes que representam esses 3 funcionários sejam obrigadas a implementar o método calculaSalario() ? Seria melhor usar classe abstrata ou interface? Justifique.

R) Calcular Salário**:** Optamos por uma classe abstrata "Funcionario" com o método abstrato calculaSalario(). Isso garante que todas as classes que herdam de "Funcionario" (Secretaria, Professor e Coordenador) implementem esse método.

2) Se quisermos obrigar que a classe que representa o Professor e a classe que representa o Coordenador implementem o método: getValorBonus(), qual seria a melhor forma: classe abstrata ou interface? Justifique.

R) Obter Valor Bônus: Usamos uma interface "Bonificavel" com o método abstrato getValorBonus(). Essa escolha permite que apenas as classes que de fato recebem bônus (Professor e Coordenador) implementem esse método. A Secretaria, por não receber bônus, não precisa implementar a interface.

3) Considere também a classe ControleBonus, que terá como atributo um ou mais List para conter todos os objetos que recebem bônus. Pergunta: é necessário ter 2 List ou apenas um?

R) Duas Listas**:** Optamos por duas Listas separadas: listaProfessores e listaCoordenadores. Essa separação facilita a manipulação e o controle dos bônus específicos de cada tipo de funcionário.

4) O polimorfismo está presente nesse sistema? Justifique.

R) Sim, o polimorfismo está presente nesse sistema de diversas maneiras:

- A classe abstrata Funcionario define um tipo genérico.
- As classes Secretaria, Professor e Coordenador são subtipos de Funcionario.
- Uma variável do tipo Funcionario pode referenciar um objeto de qualquer um dos subtipos.
- Isso permite usar a mesma interface para diferentes tipos de funcionários, com comportamentos específicos