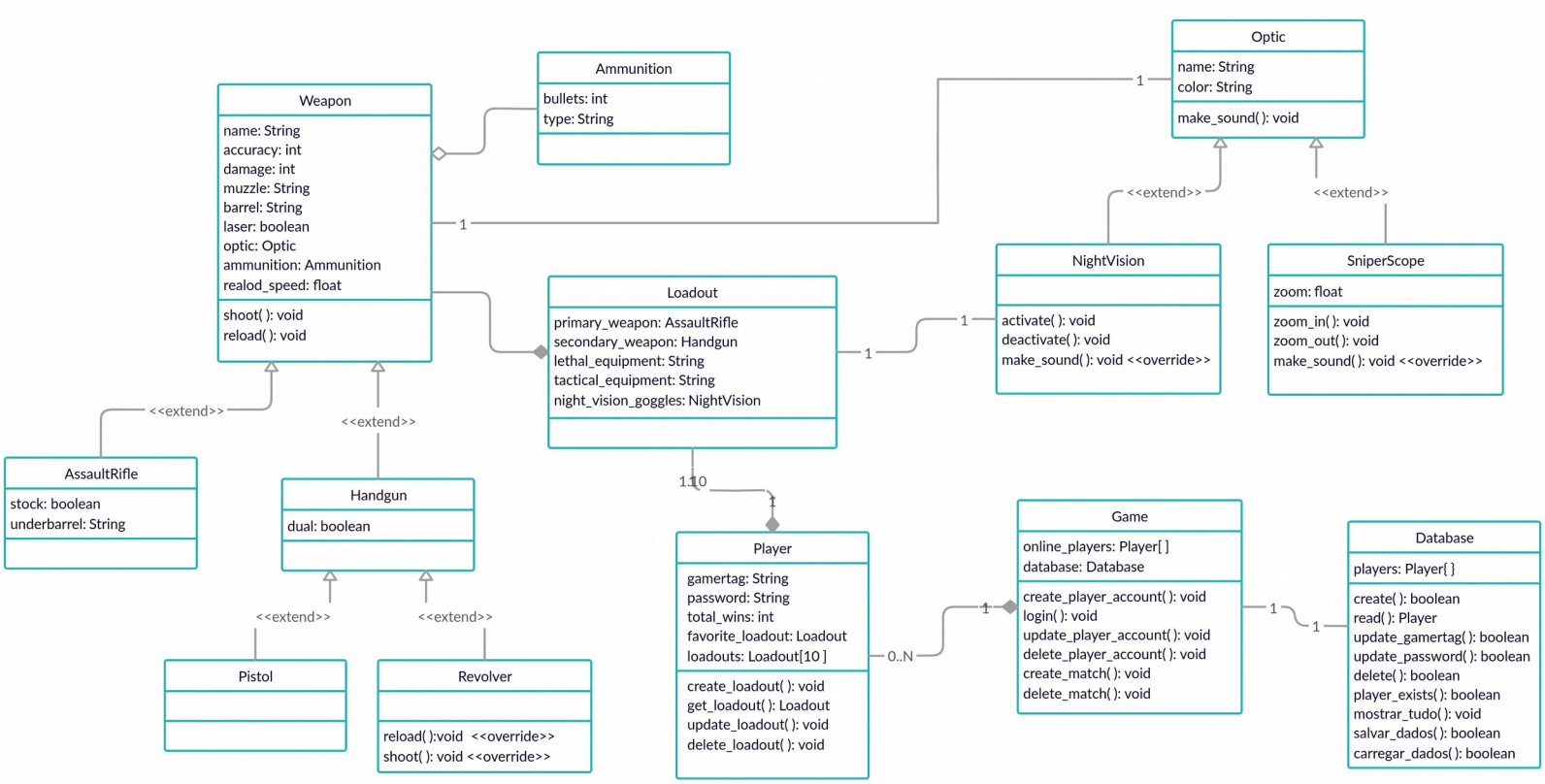
**PROJETO FINAL**

Você faz parte da equipe de desenvolvimento da Activision e irá trabalhar no desenvolvimento do Call of Duty: Modern Warfare (modo Warzone), um jogo First Person Shooter online.


John, o coordenador do projeto, decidiu que você será responsável por desenvolver o backend para a criação de funcionalidades para os jogadores. Além disso, você também deverá implementar o sistema de cadastro de jogadores. Para a sua sorte, ele disponibilizou o diagrama de classe do sistema.




## CRUD

- Você deverá implementar um sistema de cadastro para novos jogadores. Faça os 4 métodos de um CRUD (Create, Read, Update, Delete) dentro da classe ‘Database’. Os dados dos usuários serão salvos em um arquivo chamado “players.pkl”

**1) CREATE:**

- Na classe ‘Game’, implemente um método ‘create_player_account()’ que permita o usuário criar uma nova conta no jogo. Para criar uma nova conta, o jogador vai informar o nome (gamertag) e a senha de login. Para facilitar, você não precisa validar nenhum dos campos. A gamertag é um nome único, ou seja, dois jogadores não podem ter o mesmo nome. Portanto, se o usuário escolher uma gamertag indisponível, informe-o sobre o problema com a seguinte mensagem:

‘Um outro jogador já possui esse nome!’

Dica: para saber se a gamertag está indisponível, pesquisar se o jogador já existe no banco de dados usando o metodo “player_exists( )”

Por fim, caso a gamertag esteja disponível para escolha, chame o método “create( )”, da classe ‘Database’, e insira o jogador no sistema, salvando as informações no arquivo “players.pkl”

**2) READ:**

- Implemente um método ‘read( )’ na classe ‘Database’ que permita recuperar o dados de jogadores cadastrados no sistema. Este método será usado quando o usuário fizer login no sistema, informando a gamertag e a senha.

Exemplo:

player = read(gamertag, senha)

**3) UPDATE:**

- Implemente um método ‘update_gamertag( )’ e um métdo ‘update_password( )’ para permitir que o usuário altere/atualize seus dados. Ele pode alterar a senha e a gamertag (considerando que outro jogador já não tenha o mesmo nome). Salve as alterações no arquivo players.pkl

Exemplo:

update_gamertag(gamertag, senha, nova_gamertag)
update_password(gamertag, senha_atual, nova_senha)

**4) DELETE:**

- Implemente um método ‘delete( )’ na classe Database para excluir a conta de um jogador. O registro do jogador no arquivo ‘players.pkl’ deverá ser apagado.

Exemplo:

delete(gamertag, senha)

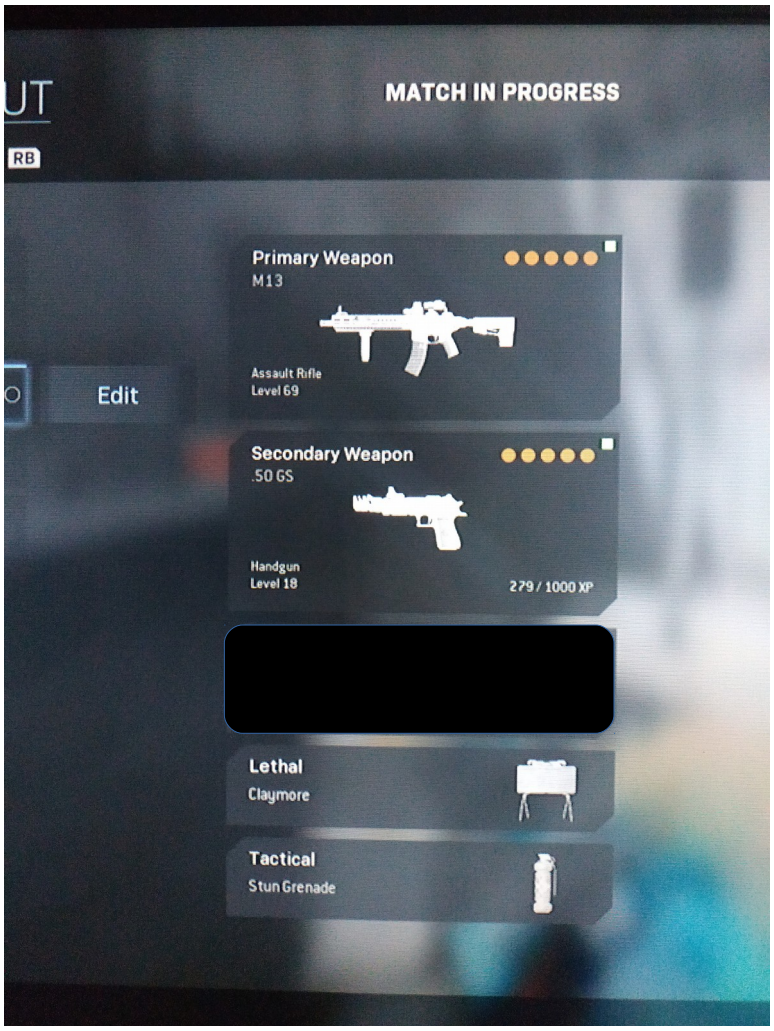
**Teste as 4 funcionalidades do CRUD com valores de exemplo ***

---

## LOADOUT

- Continuando a construção do modelo dos dados do jogador, além da gamertag e da senha, todo jogador terá a opção de criar loadouts customizados para jogar. Utilize os conceitos de orientação a objetos (herança, polimorfismo, abstração ...) para construir toda a hierarquia de classes (armas, equipamentos, acessórios, etc)

Um loadout possui as seguintes características:




> Arma Primária
> Arma Secundária
> Equipamento Letal
> Equipamento Tático

A arma primaria é, obrigatoriamente, uma Assault Rifle
A arma secundária é, obrigatoriamente, uma Handgun (Pistola ou Revólver)
Os equipamentos táticos e letais podem ser qualquer tipo de equipamento (uma String apenas com o nome do equipamento)

---