

 UEPB	UNIVERSIDADE ESTADUAL DA PARAÍBA CENTRO DE CIÊNCIAS E TECNOLOGIA CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO		
ATIVIDADE DE ESTUDO VI		ANO 2021.1	
CURSO	Ciência da Computação		
DISCIPLINA	Linguagem de Programação II		
PROFESSOR		TITULAÇÃO	
Jucelio Soares dos Santos		Mestrado	
NOME	Lucas de Lucena Siqueira		
MATRÍCULA	201080354	CONCEITO	
DATA			

DESCRIÇÃO DA ATIVIDADE

1. Conforme mostrado nesta aula implemente:
 - a. A função cadastro para o jogo da velha.

```

➤ public static void cadastro() {
    System.out.println("Digite o nome do primeiro jogador: ");
    String nome_jogador1 = sc.next();
    jogador1 = buscarJogador(nome_jogador1);
    System.out.println("Digite o nome do segundo jogador: ");
    String nome_jogador2 = sc.next();
    jogador2 = buscarJogador(nome_jogador2);
}

```

- b. A funcionalidade de contagem de tempo no jogo da velha;

```

public static void setTempo() {
    inicio = System.currentTimeMillis();
}

public static void finalTempo() {
    System.out.println("Tempo de jogo: " + ((System.currentTimeMillis()-inicio)/1000) + " segundos.");
}

```

- c. As funcionalidades de ranking para o jogo da velha;

```

public static Jogador buscarJogador(String nome) {
    Jogador jogador = null;
    for(int i=0; (i<quantidadeJogadores) && (jogador == null); i++) {
        if(jogadores[i] != null && jogadores[i].name.equalsIgnoreCase(nome)) {
            jogador = jogadores[i];
        }
    }
    if(jogador == null) {
        jogador = new Jogador();
        jogador.name = nome;
        if(quantidadeJogadores<50) {
            jogadores[quantidadeJogadores] = jogador;
            quantidadeJogadores = quantidadeJogadores + 1;
        }
    }
    return jogador;
}
}

```

d. A funcionalidades de salvar ranking em arquivo para o jogo da velha.

```

public static void salvarJogadores() {
    try {
        ObjectOutputStream saida = new ObjectOutputStream(new FileOutputStream(arquivo));
        saida.writeObject(jogadores);
    } catch(Exception e) {
        throw new RuntimeException(e);
    }
}

public static void lerJogadores() {
    try {
        ObjectInputStream saida = new ObjectInputStream(new FileInputStream(arquivo));
        jogadores = (Jogador[]) saida.readObject();
        while (jogadores[quantidadeJogadores] != null && quantidadeJogadores < 50) {
            quantidadeJogadores = quantidadeJogadores + 1;
        }
    } catch (FileNotFoundException e) {
    }
    catch (Exception e) {
        throw new RuntimeException(e);
    }
}
}

```

2. Descreva os principais recursos que você utilizou para implementar as seguintes funcionalidades:
 - a. Cadastro de jogadores.

É uma função simples onde lemos os nomes dos jogadores e adicionamos em uma variável e depois fazemos a busca desse jogador e verificamos se já existe ou não no array. Se não existir, adicionamos no array.

b. Contagem de tempo da partida.

Foi criada uma função para pegar o tempo de início e outra para pegar o tempo novamente subtraindo o início.

c. Registro de vitórias e derrotas dos jogadores.

Foi criada a classe Jogador com nome, total de vitórias e total de derrotas e na classe principal foi criado um array para guardar os objetos do tipo Jogador onde foi salvo O nome e o total de vitória e derrota de cada jogador em cada posição do array.

d. Salvar e ler os registros em arquivo.

Foi criado um arquivo e então duas funções: uma chamada salvarJogadores onde vai escrever no arquivo o histórico da partida e a lerJogadores que vai verificar no arquivo que foi criado todo o histórico salvo.