

```

public Equip[] processActa(Equip inputEqLocal, Equip inputEqVisitant, Acta inputActa) throws
CloneNotSupportedException{
    // CLONO equips nous amb el mateix nom
    Equip outEqLocal = (Equip)inputEqLocal.clone();
    Equip outEqVisitant = (Equip)inputEqVisitant.clone();
    // Agafo els valors dels equips d'entrada
    Map hashMapLocal = (HashMap)outEqLocal.getRow();
    Map hashMapVisitant = (HashMap)outEqVisitant.getRow();
    Integer[] eventsAcumulats = {0,0,0,0,0,0}; // Aquí aniré acumulant els events.
    Integer eventWhile = 0; // Em recorro tots els events del partit amb un while i creo index necessaris
    Integer typeWhile, localWhile, index;
    List<Event> listEvents = inputActa.getListEvents();
    while (eventWhile < inputActa.getNumEvents())
    { // Agafo el següent event i actualitzo l'index per la propera iteració
        Event thisEvent = listEvents.get(eventWhile);
        eventWhile += 1;
        // Això torna 0 si es Assaig, 1 si és Transpormació i 2 si és xut;
        typeWhile = Arrays.asList(thisEvent.getPossibleTypes()).indexOf(thisEvent.getType());
        // Això torna 0 si es Local, 1 si és Visitant;
        localWhile = Arrays.asList(thisEvent.getPossibleWho()).indexOf(thisEvent.getWho());
        index = thisEvent.getPossibleTypes().length*localWhile + typeWhile;
        eventsAcumulats[index] += 1; // Acumulo l'event
    }
    // Calculo els punts del partit
    Integer scoreLocal = eventsAcumulats[0]*pointsEvent[0] + eventsAcumulats[1]*pointsEvent[1] +
eventsAcumulats[2]*pointsEvent[2];
    Integer scoreVisitant = eventsAcumulats[3]*pointsEvent[0] + eventsAcumulats[4]*pointsEvent[1] +
eventsAcumulats[5]*pointsEvent[2];
    Integer signum = Long.signum((long)scoreLocal - scoreVisitant);
    switch (signum) // Faig coses en funció de Victòria - empat derrota
    {
        case -1 -> // Derrota Local
        {
            if (scoreVisitant - scoreLocal < 8) { // Bonus Point perdre per 7 o menys
                hashMapLocal.replace("BP", (Integer)hashMapLocal.get("BP") + 1); // Suma Bonus points
                hashMapLocal.replace("P", (Integer)hashMapLocal.get("P") + 1); // Suma Bonus points a points
                hashMapLocal.replace("D", (Integer)hashMapLocal.get("D") + 1); // Victoria Derrota
                hashMapVisitant.replace("V", (Integer)hashMapVisitant.get("V") + 1); // Victoria Derrota
                hashMapVisitant.replace("P", (Integer)hashMapVisitant.get("P") + pointsVictory ); // Suma punts partit
            }
            case 0 -> // Empat
            {
                hashMapLocal.replace("E", (Integer)hashMapLocal.get("E") + 1); // Victoria Derrota
                hashMapVisitant.replace("E", (Integer)hashMapVisitant.get("E") + 1); // Victoria Derrota
                hashMapLocal.replace("P", (Integer)hashMapLocal.get("P") + pointsDraw ); // Suma punts partit
                hashMapVisitant.replace("P", (Integer)hashMapVisitant.get("P") + pointsDraw ); // Suma punts partit
            }

            case 1 -> // Victòria Local
            {
                if (scoreLocal - scoreVisitant < 8) { // Bonus Point perdre per 7 o menys
                    hashMapVisitant.replace("BP", (Integer)hashMapVisitant.get("BP") + 1); // Suma Bonus points
                    hashMapVisitant.replace("P", (Integer)hashMapVisitant.get("P") + 1); // Suma Bonus points a points
                    hashMapLocal.replace("V", (Integer)hashMapLocal.get("V") + 1); // Victoria Derrota
                    hashMapVisitant.replace("D", (Integer)hashMapVisitant.get("D") + 1); // Victoria Derrota
                    hashMapLocal.replace("P", (Integer)hashMapLocal.get("P") + pointsVictory ); // Suma punts partit
                }
                default -> { } // No faig errors en aquest codi
            }
        }
        hashMapLocal.replace("PJ", (Integer)hashMapLocal.get("PJ") + 1); // Afegeixo 1 partit jugat PJ
        hashMapVisitant.replace("PJ", (Integer)hashMapVisitant.get("PJ") + 1);
        if (eventsAcumulats[0] > 3) // 4 o més assajos locals
        {
            hashMapLocal.replace("BP", (Integer)hashMapLocal.get("BP") + 1); // Suma Bonus points
            hashMapLocal.replace("P", (Integer)hashMapLocal.get("P") + 1);
        }
    }
}

```

```

        hashMapLocal.replace("DP", (Integer)hashMapLocal.get("DP") + scoreLocal - scoreVisitant ); // Actualitzo diferencia de punts
        hashMapLocal.replace("DA", (Integer)hashMapLocal.get("DA") + eventsAcumulats[0] - eventsAcumulats[3] ); // Actualitzo diferencia d'assajos
        if (eventsAcumulats[3] > 3) // 4 o més assajos Visitants
        {
            hashMapVisitant.replace("BP", (Integer)hashMapVisitant.get("BP") +1 ); // Suma Bonus points
            hashMapVisitant.replace("P", (Integer)hashMapVisitant.get("P") +1 );
        }
        hashMapVisitant.replace("DP", (Integer)hashMapVisitant.get("DP") - scoreLocal + scoreVisitant ); // Actualitzo diferencia de punts
        hashMapVisitant.replace("DA", (Integer)hashMapVisitant.get("DA") - eventsAcumulats[0] + eventsAcumulats[3] ); // Actualitzo diferencia d'assajos
        Integer[] scoreArray = new Integer[2]; // Incloc la puntuació a l'acta.
        scoreArray[0]= scoreLocal;
        scoreArray[1] = scoreVisitant;
        inputActa.setScore(scoreArray);
        outEqLocal.setROW(hashMapLocal); // Actualitzo els HashMaps
        outEqVisitant.setROW(hashMapVisitant);
        Equip[] arrayEquips = new Equip[2]; // Torno l'array d'equips actualitzats
        arrayEquips[0] = outEqLocal;
        arrayEquips[1] = outEqVisitant;
        return arrayEquips;
    }

```

9

10