

## ClubTennis>>mostrarPuntajesJugadoresEnFecha: aDate

```
| partidosDeLaFecha result |
result := WriteStream on: String new.
result
    nextPutAll: 'Puntajes para los partidos de la fecha', aDate asString;    cr.
partidosDeLaFecha := coleccionPartidos select: [ :p | p fecha = aDate ].
partidosDeLaFecha
    do: [ :partido | | j1 j2 totalGames |
        result
            nextPutAll: 'Partido: '; cr.
j1 := partido jugador1.
totalGames := 0.
result
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j1 getNombreJugador;
    nextPutAll: ': '.
(partido puntosDelJugador: j1)
    do: [ :gamesDelSet |
        result
            nextPutAll: gamesDelSet asString, ';'.
            totalGames := totalGames + gamesDelSet ].
result nextPutAll: ' Puntos del partido: '.
j1 zona = 'A'
    ifTrue: [ result nextPutAll: (totalGames * 2) asString ].
j1 zona = 'B'
    ifTrue: [ result nextPutAll: totalGames asString ].
j1 zona = 'C'
    ifTrue: [ partido ganador = j1
        ifTrue: [ result nextPutAll: totalGames asString ]
        ifFalse: [ result nextPutAll: 0 asString ] ].
j2 := partido jugador2.
totalGames := 0.
result
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j2 getNombreJugador;
    cr.
(partido puntosDelJugador: j2)
    do: [ :gamesDelSet |
        result
            nextPutAll: gamesDelSet asString;
            nextPut: ';'.
            totalGames := totalGames + gamesDelSet ].
result nextPutAll: ' Puntos del partido: '.
j2 zona = 'A'
    ifTrue: [ result nextPutAll: (totalGames * 2) asString ].
j2 zona = 'B'
    ifTrue: [ result nextPutAll: totalGames asString ].
j2 zona = 'C'
    ifTrue: [ partido ganador = j2
        ifTrue: [ result nextPutAll: totalGames asString ]
        ifFalse: [ result nextPutAll: 0 asString ] ] ].
```

^ result contents

## 1) Extract Method en #mostrarPuntajesJugadoresEnFecha:

### ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate

```
| partidosDeLaFecha result |
result := WriteStream on: String new.
result
    nextPutAll: 'Puntajes para los partidos de la fecha', aDate asString;    cr.
partidosDeLaFecha := coleccionPartidos select: [ :p | p fecha = aDate ].
partidosDeLaFecha
    do: [ :partido | self mostrarPartido: partido en: result].
^ result contents
```

### ClubTenis>>mostrarPartido: partido en: result

```
| j1 j2 totalGames |
result nextPutAll: 'Partido: '; cr.
j1 := partido jugador1.
totalGames := 0.
result
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j1 getNombreJugador;
    nextPutAll: ': '.
(partido puntosDelJugador: j1)
    do: [ :gamesDelSet |
        result nextPutAll: gamesDelSet asString, ';'.
        totalGames := totalGames + gamesDelSet ].
result nextPutAll: ' Puntos del partido: '.
j1 zona = 'A'
    ifTrue: [ result nextPutAll: (totalGames * 2) asString ].
j1 zona = 'B'
    ifTrue: [ result nextPutAll: totalGames asString ].
j1 zona = 'C'
    ifTrue: [ partido ganador = j1
        ifTrue: [ result nextPutAll: totalGames asString ]
        ifFalse: [ result nextPutAll: 0 asString ] ].
j2 := partido jugador2.
totalGames := 0.
result
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j2 getNombreJugador;
    cr.
(partido puntosDelJugador: j2)
    do: [ :gamesDelSet |
        result
            nextPutAll: gamesDelSet asString;
            nextPut: ';'.
        totalGames := totalGames + gamesDelSet ].
result nextPutAll: ' Puntos del partido: '.
j2 zona = 'A'
    ifTrue: [ result nextPutAll: (totalGames * 2) asString ].
j2 zona = 'B'
    ifTrue: [ result nextPutAll: totalGames asString ].
j2 zona = 'C'
    ifTrue: [ partido ganador = j2
        ifTrue: [ result nextPutAll: totalGames asString ]
        ifFalse: [ result nextPutAll: 0 asString ] ] ].
```

## 2) Move Method #mostrarPartido:en: a la clase Partido

### ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate

```
| partidosDeLaFecha result |
result := WriteStream on: String new.
result
    nextPutAll: 'Puntajes para los partidos de la fecha', aDate asString;    cr.
partidosDeLaFecha := coleccionPartidos select: [ :p | p fecha = aDate ].
partidosDeLaFecha
    do: [ :partido | partido mostrarEn: result].
^ result contents
```

### Partido>>mostrarEn: result

```
| j1 j2 totalGames |
result nextPutAll: 'Partido: '; cr.
j1 := self jugador1.
totalGames := 0.
result
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j1 getNombreJugador;
    nextPutAll: ': '.
(self puntosDelJugador: j1)
    do: [ :gamesDelSet |
        result nextPutAll: gamesDelSet asString, ';'.
        totalGames := totalGames + gamesDelSet ].
result nextPutAll: ' Puntos del partido: '.
j1 zona = 'A'
    ifTrue: [ result nextPutAll: (totalGames * 2) asString ].
j1 zona = 'B'
    ifTrue: [ result nextPutAll: totalGames asString ].
j1 zona = 'C'
    ifTrue: [ self ganador = j1
        ifTrue: [ result nextPutAll: totalGames asString ]
        ifFalse: [ result nextPutAll: 0 asString ] ].
j2 := self jugador2.
totalGames := 0.
result
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j2 getNombreJugador;
    cr.
(self puntosDelJugador: j2)
    do: [ :gamesDelSet |
        result
            nextPutAll: gamesDelSet asString;
            nextPut: ';'.
        totalGames := totalGames + gamesDelSet ].
result nextPutAll: ' Puntos del partido: '.
j2 zona = 'A'
    ifTrue: [ result nextPutAll: (totalGames * 2) asString ].
j2 zona = 'B'
    ifTrue: [ result nextPutAll: totalGames asString ].
j2 zona = 'C'
    ifTrue: [ self ganador = j2
        ifTrue: [ result nextPutAll: totalGames asString ]
        ifFalse: [ result nextPutAll: 0 asString ] ] ].
```

### 3) Rename Variable (parámetro result -> unStream)

ClubTennis>>mostrarPuntajesJugadoresEnFecha: aDate  
...idem...

Partido>>mostrarEn: unStream

```
| j1 j2 totalGames |
unStream nextPutAll: 'Partido: '; cr.
j1 := self jugador1.
totalGames := 0.
unStream
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j1 getNombreJugador;
    nextPutAll: ': '.
(self puntosDelJugador: j1)
    do: [ :gamesDelSet |
        unStream nextPutAll: gamesDelSet asString, ';'.
        totalGames := totalGames + gamesDelSet ].
unStream nextPutAll: ' Puntos del partido: '.
j1 zona = 'A'
    ifTrue: [ unStream nextPutAll: (totalGames * 2) asString ].
j1 zona = 'B'
    ifTrue: [ unStream nextPutAll: totalGames asString ].
j1 zona = 'C'
    ifTrue: [ self ganador = j1
        ifTrue: [ unStream nextPutAll: totalGames asString ]
        ifFalse: [ unStream nextPutAll: 0 asString ] ].
j2 := self jugador2.
totalGames := 0.
unStream
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: j2 getNombreJugador;
    cr.
(self puntosDelJugador: j2)
    do: [ :gamesDelSet |
        unStream
            nextPutAll: gamesDelSet asString;
            nextPut: ';'.
        totalGames := totalGames + gamesDelSet ].
unStream nextPutAll: ' Puntos del partido: '.
j2 zona = 'A'
    ifTrue: [ unStream nextPutAll: (totalGames * 2) asString ].
j2 zona = 'B'
    ifTrue: [ unStream nextPutAll: totalGames asString ].
j2 zona = 'C'
    ifTrue: [ self ganador = j2
        ifTrue: [ unStream nextPutAll: totalGames asString ]
        ifFalse: [ unStream nextPutAll: 0 asString ] ] ].
```

#### 4) Extract Method en #mostrarEn:

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**  
*...idem...*

**Partido>>mostrarEn: unStream**  
unStream nextPutAll: 'Partido: '; cr.  
**self mostrarPuntosDe: self jugador1 en: unStream.**  
**self mostrarPuntosDe: self jugador2 en: unStream.**

**Partido>>mostrarPuntosDe: unJugador en: unStream**  
| totalGames |  
totalGames := 0.  
unStream  
    nextPutAll: 'Puntaje del jugador: ';  
    nextPutAll: **unJugador** getNombreJugador;  
    nextPutAll: ': '.  
(self puntosDelJugador: **unJugador**)  
    do: [ :gamesDelSet |  
        unStream nextPutAll: gamesDelSet asString, ';'.  
        totalGames := totalGames + gamesDelSet ].  
unStream nextPutAll: ' Puntos del partido: '.  
**unJugador** zona = 'A'  
    ifTrue: [ unStream nextPutAll: (totalGames \* 2) asString ].  
**unJugador** zona = 'B'  
    ifTrue: [ unStream nextPutAll: totalGames asString ].  
**unJugador** zona = 'C'  
    ifTrue: [ self ganador = **unJugador**  
        ifTrue: [ unStream nextPutAll: totalGames asString ]  
        ifFalse: [ unStream nextPutAll: 0 asString ] ].

5) **Move Method Partido>>mostrarPuntosDe:en: a Jugador>>mostrarPuntosDePartido:en:**

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**  
*...idem...*

**Partido>>mostrarEn: unStream**  
unStream nextPutAll: 'Partido: '; cr.  
**jugador1 mostrarPuntosDePartido: self en: unStream.**  
**jugador2 mostrarPuntosDePartido: self en: unStream.**

**Jugador>>mostrarPuntosDePartido: unPartido en: unStream**  
| totalGames |  
totalGames := 0.  
unStream  
    nextPutAll: 'Puntaje del jugador: ';  
    nextPutAll: **self** getNombreJugador;  
    nextPutAll: ': '.  
**(unPartido puntosDelJugador: self)**  
    do: [ :gamesDelSet |  
        unStream nextPutAll: gamesDelSet asString, ';'.  
        totalGames := totalGames + gamesDelSet ].  
unStream nextPutAll: ' Puntos del partido: '.  
**self** zona = 'A'  
    ifTrue: [ unStream nextPutAll: (totalGames \* 2) asString ].  
**self** zona = 'B'  
    ifTrue: [ unStream nextPutAll: totalGames asString ].  
**self** zona = 'C'  
    ifTrue: [ **unPartido** ganador = **self**  
        ifTrue: [ unStream nextPutAll: totalGames asString ]  
        ifFalse: [ unStream nextPutAll: 0 asString ] ].

## 6) Replace Conditional with Polymorphism en Jugador>>mostrarPuntosDePartido:en:

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**  
...idem...

**Partido>>mostrarEn: unStream**  
...idem...

**Jugador>>mostrarPuntosDePartido: unPartido en: unStream**  
^self subclassResponsibility

**JugadorZonaA>>mostrarPuntosDePartido: unPartido en: unStream**  
| totalGames |  
totalGames := 0.  
unStream  
    nextPutAll: 'Puntaje del jugador: ';  
    nextPutAll: **self** getNombreJugador;  
    nextPutAll: ': '.  
(unPartido puntosDelJugador: self)  
    do: [ :gamesDelSet |  
        unStream nextPutAll: gamesDelSet asString, ';'.  
        totalGames := totalGames + gamesDelSet ].  
unStream nextPutAll: ' Puntos del partido: '.  
**unStream nextPutAll: (totalGames \* 2) asString**

...similar para JugadorZonaB y JugadorZonaC...

## 7) Extract Method y Pull Up Method

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**  
...idem...

**Partido>>mostrarEn: unStream**  
...idem...

**Jugador>>mostrarPuntosDePartido: unPartido en: unStream**  
^self subclassResponsibility

**Jugador>>mostrarNombreEn: unStream**  
unStream  
    nextPutAll: 'Puntaje del jugador: ';  
    nextPutAll: **self nombre**;           “Rename de #getNombreJugador”  
    nextPutAll: ': '.

**JugadorZonaA>>mostrarPuntosDePartido: unPartido en: unStream**  
| totalGames |  
totalGames := 0.  
**self mostrarNombreEn: unStream.**  
(unPartido puntosDelJugador: self)  
    do: [ :gamesDelSet |  
        unStream nextPutAll: gamesDelSet asString, ';'.  
        totalGames := totalGames + gamesDelSet ].  
unStream nextPutAll: ' Puntos del partido: '.  
unStream nextPutAll: (totalGames \* 2) asString

...similar para JugadorZonaB y JugadorZonaC...

## 8) Replace Temp with Query

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**  
*...idem...*

**Partido>>mostrarEn: unStream**  
*...idem...*

**Jugador>>mostrarPuntosDePartido: unPartido en: unStream**  
*^self subclassResponsibility*

**Jugador>>mostrarNombreEn: unStream**  
*...idem...*

**Jugador>>totalGamesEn: unPartido**  
| totalGames |  
totalGames := 0.  
(unPartido puntosDelJugador: self)  
do: [ :gamesDelSet | totalGames := totalGames + gamesDelSet]

**JugadorZonaA>>mostrarPuntosDePartido: unPartido en: unStream**  
self mostrarNombreEn: unStream.  
(unPartido puntosDelJugador: self)  
do: [ :gamesDelSet |  
unStream nextPutAll: gamesDelSet asString, ';'].  
unStream nextPutAll: ' Puntos del partido: '.  
unStream nextPutAll: ((self totalGamesEn: unPartido) \* 2) asString

*...similar para JugadorZonaB y JugadorZonaC...*

## 9) Extract Method y Pull Up Method

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**  
*...idem...*

**Partido>>mostrarEn: unStream**  
*...idem...*

**Jugador>>mostrarPuntosDePartido: unPartido en: unStream**  
*...idem...*

**Jugador>>mostrarNombreEn: unStream**  
*...idem...*

**Jugador>>totalGamesEn: unPartido**  
*...idem...*

**Jugador>>mostrarSetsEnPartido: unPartido en: unStream**  
(unPartido puntosDelJugador: self)  
do: [ :gamesDelSet |  
unStream nextPutAll: gamesDelSet asString, ';'].

**JugadorZonaA>>mostrarPuntosDePartido: unPartido en: unStream**  
self mostrarNombreEn: unStream.  
**self mostrarSetsEnPartido: unPartido en: unStream.**  
unStream nextPutAll: ' Puntos del partido: '.  
unStream nextPutAll: ((self totalGamesEn: unPartido) \* 2) asString



6-B) Alternativamente a bajar todo el código de #mostrarPuntosDePartido:en: de Jugador a las subclases, puedo aplicar Extract Method en Jugador y luego bajar a las subclases solo el método que difiere.

**ClubTenis>>mostrarPuntajesJugadoresEnFecha: aDate**

```
| result |
result := WriteStream on: String new.
result nextPutAll: 'Puntajes para los partidos de la fecha' , aDate asString;    cr.
(coleccionPartidos select: [ :p | p fecha = aDate ])
    do: [ :partido | partido mostrarEn: result].
^result contents
```

**Partido>>mostrarEn: unStream**

```
unStream nextPutAll: 'Partido: '; cr.
self jugador1 mostrarPuntosDePartido: self en: unStream.
self jugador2 mostrarPuntosDePartido: self en: unStream.
```

**Jugador>>mostrarPuntosDePartido: unPartido en: unStream**

```
self mostrarNombreEn: unStream.
self mostrarSetsEnPartido: unPartido en: unStream.
self mostrarPuntosTotalesEn: unPartido en: unStream
```

**Jugador>>mostrarNombreEn: unStream**

```
unStream
    nextPutAll: 'Puntaje del jugador: ';
    nextPutAll: self nombre;
    nextPutAll: ': '.
```

**Jugador>>mostrarSetsEnPartido: unPartido en: unStream**

```
(unPartido puntosDelJugador: self)
    do: [ :gamesDelSet |
        unStream nextPutAll: gamesDelSet asString, ';' ].
```

**Jugador>>mostrarPuntosTotales: unPartido en: unStream**

```
unStream nextPutAll: ' Puntos del partido: '.
unStream nextPutAll: (self calcularPuntosEn: unPartido) asString.
```

**Jugador>> calcularPuntosEn: unPartido**

```
^self subclassResponsibility
```

**JugadorZonaA>> calcularPuntosEn: unPartido**

```
^(self totalGamesEn: unPartido) * 2
```

**JugadorZonaB>> calcularPuntosEn: unPartido**

```
^(self totalGamesEn: unPartido)
```

**JugadorZonaC>> calcularPuntosEn: unPartido**

```
^unPartido ganador = self
    ifTrue: [self totalGamesEn: unPartido]
    ifFalse: [0]
```