

```
// Función recursiva para generar el calendario de enfrentamientos
```

```
fun generarCalendario(n: Int): Array<Array<Int?>> {  
    // Base: Si solo hay 2 jugadores, se enfrentan entre sí  
    if (n == 2) {  
        return arrayOf(  
            arrayOf(1, 2),  
            arrayOf(2, 1)  
        )  
    }  
}
```

```
// Recursividad: Dividimos en dos mitades
```

```
val mitad = n / 2
```

```
val mitadSuperior = generarCalendario(mitad)
```

```
val mitadInferior = generarCalendario(mitad)
```

```
// Creamos la tabla de n x (n-1) días
```

```
val tabla = Array(n) { arrayOfNulls<Int>(n - 1) }
```

```
// Copiamos los enfrentamientos dentro de cada mitad
```

```
for (i in 0 until mitad) {  
    for (j in 0 until mitad - 1) {  
        tabla[i][j] = mitadSuperior[i][j] // Mitad superior  
        tabla[i + mitad][j] = mitadInferior[i][j] // Mitad inferior  
    }  
}
```

```
// Rellenamos los enfrentamientos entre mitades (rotando los jugadores)
```

```
for (i in 0 until mitad) {  
    for (j in mitad - 1 until n - 1) {
```

```

        tabla[i][j] = i + mitad + 1 // Jugadores de la mitad superior enfrentan a los de la inferior
        tabla[i + mitad][j] = i + 1 // Jugadores de la mitad inferior enfrentan a los de la superior
    }
}

return tabla
}

```

// Función para imprimir la tabla de enfrentamientos, mostrando día por día

```

fun imprimirCalendario(calendario: Array<Array<Int?>>){
    for (dia in calendario[0].indices) {
        println("Día ${dia + 1}:")
        for (jugador in calendario.indices step 2) {
            val oponente = calendario[jugador][dia]
            println("Jugador ${jugador + 1} vs Jugador ${oponente}")
        }
        println() // Separar los días
    }
}

```

```

fun main() {
    val n = 8 // Cantidad de jugadores, debe ser potencia de 2
    if (n > 0 && (n and (n - 1)) == 0) { // Comprobación de que n es potencia de 2
        val calendario = generarCalendario(n)
        imprimirCalendario(calendario)
    } else {
        println("El número de jugadores debe ser una potencia de 2.")
    }
}

```