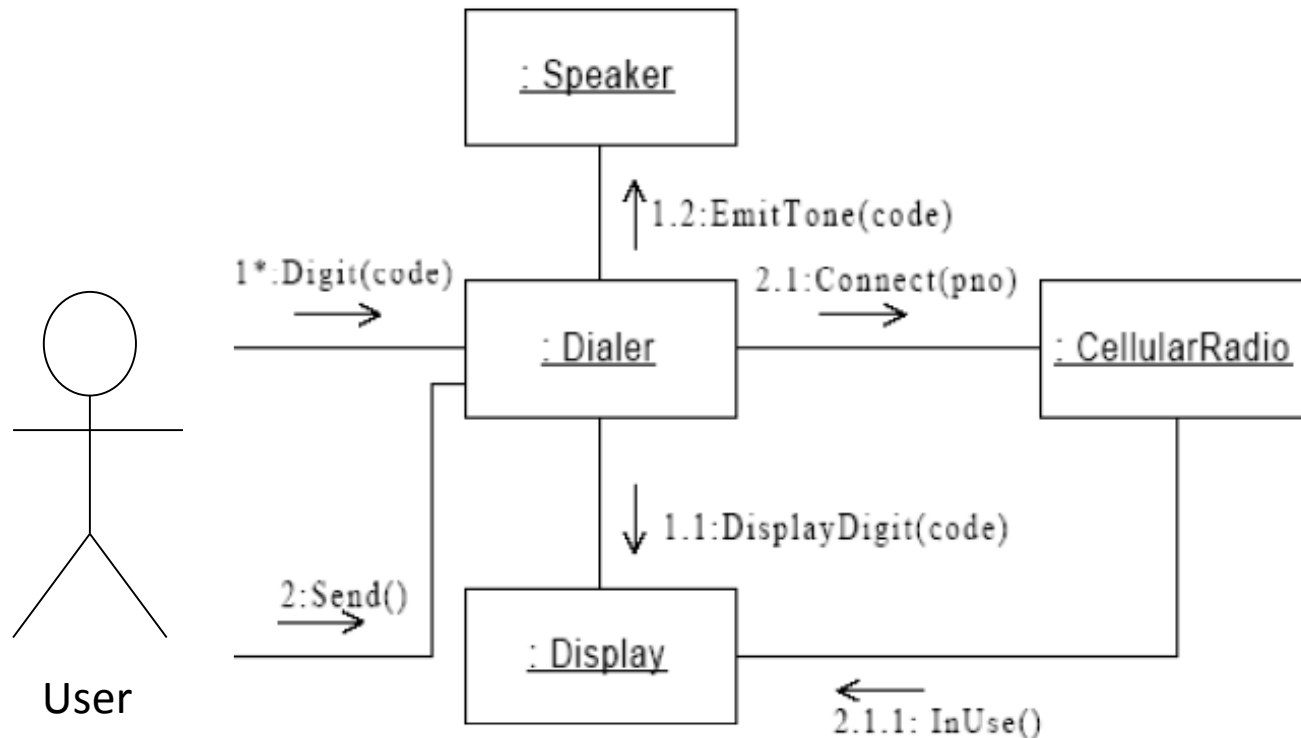


Identificar las clases relevantes y realizar el diagrama de colaboración de diseño para el siguiente caso de uso, que corresponde al inicio de realización de una llamada desde un móvil :

- El usuario pulsa los dígitos del número de teléfono
- Para cada dígito:
  - La **pantalla (Display)** se actualiza para añadir el dígito marcado
  - Se emite un **tono (Speaker)**
- El usuario pulsa el botón Enviar Se establece conexión con la **red (CellularRadio)** y una vez iniciado el proceso de conexión el indicador en uso se ilumina en pantalla

Ayuda: incluir una clase **Dialer** (controlador)

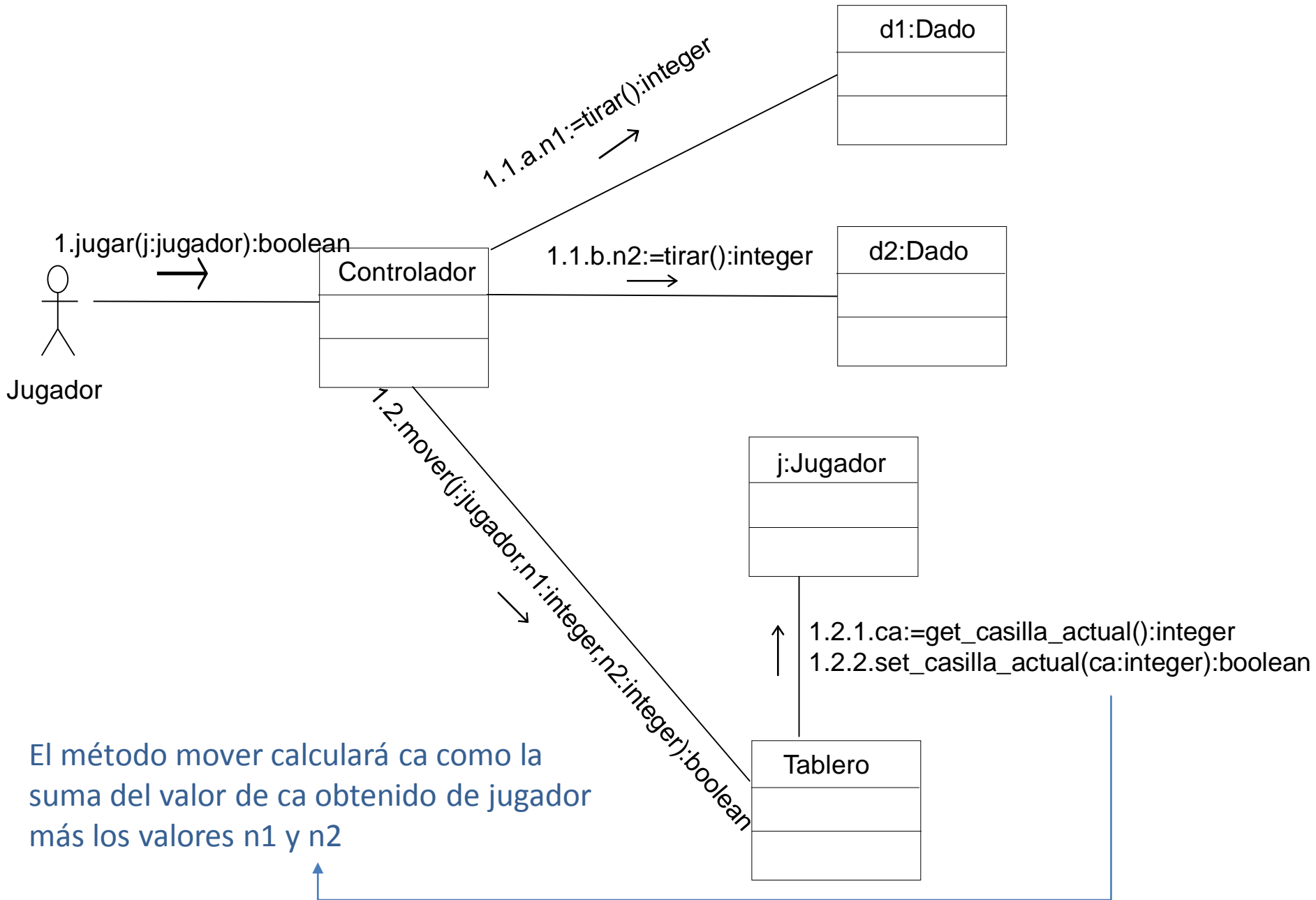


“y una vez iniciado el proceso de conexión el indicador en uso se ilumina” indica que una vez lanzado 2.1, sin necesidad de que termine, se debe ejecutar ‘In use’, por lo que debe ser 2.1.1 (llamado desde el método 2.1). Si se iluminará el indicador en uso después de terminada la conexión, sería 2.2

Realiza el diagrama de colaboración de diseño del caso de uso Realizar jugada, para el juego del parchís (con dos dados). Este caso de uso comienza cuando la interfaz (actor Jugador) envía el evento Jugar al Sistema.

Disponemos de la clase Dado, que tiene un método tirar():integer, la clase Tablero con el método mover(j:jugador, n1:integer, n2:integer):boolean y la clase Jugador con el atributo casilla\_actual: integer y los métodos get\_casilla\_actual():integer y set\_casilla\_actual(ca:integer):boolean.

El controlador dispondrá de un método jugar(j:jugador):boolean, que consistirá en tirar dos dados en paralelo, mirar en que casilla está el jugador y situarlo en la nueva casilla.

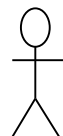


Una biblioteca tiene libros. Cada libro tiene varias copias. Cada copia tienen un atributo estado que indica si la copia está libre o prestada. Los lectores pueden pedir libros prestados. Los prestamos indicarán la copia prestada y la fecha inicial y final esperada del préstamo y la fecha real de fin del préstamo. Cuando un lector devuelve un préstamo después de su fecha final, por cada día de retraso se impone una multa de 2 días sin poder pedir ningún préstamo, por lo que cada lector tendrá asociada una fecha hasta la cual no puede pedir un nuevo préstamo (fecha\_multa).

Realizar el diagrama de colaboración del caso de uso Devolver préstamo.

Solución: La clase Préstamo es una clase de asociación entre las clases Copia y Lector, por lo que tanto Copia como Lector tiene visibilidad de Préstamo.

1.f\_multa:=devolverPrestamo(id\_l:int,id\_c:int):date



Bibliotecario

Controlador

1.1.id\_c:=buscar(id\_c:int):copia

1.2.set\_estado('libre')

1.3.id\_l:=buscar(id\_l:int):lector

1.4.f\_multa:=devolver(id\_c:copia):date

1.4.2.set\_f\_real\_fin()

Copia

id\_c:Copia

Lector

id\_l:Lector

1.4.3.f\_multa:=calMulta():date

1.4.1.id\_p:=buscar(id\_c:int):prestamo

id\_p:Prestamo

Prestamo

## Explicación necesaria de métodos:

El método `calMulta`, comparará la fecha del día con la fecha final esperada del préstamo, que conoce ya que ha instanciado el préstamo, y si es posterior, calculará los días de multa que serán los días de diferencia multiplicados por dos. Si el lector tiene null en `fecha_multa`, se calculará la fecha de multa sumando el número de días de multa a la fecha del día. Si tiene ya una fecha de multa se calculará la nueva fecha sumando el número de días de multa a la fecha de multa actual.