

Juego de palabras

Especificación

Versión 2 — 14/10/2022

Algoritmos y Estructuras de Datos 2

Segundo cuatrimestre de 2022

Cambios en la Versión 2 del 14/10/2022:

- Se agregó la notificación del turno del próximo jugador en el axioma `notificaciones(recibirMensaje(s, cid, o), cid')` del TAD `SERVIDOR`.
- Se agregaron condiciones en los axiomas de las operaciones `secuenciaFormaPalabraHorizontal?` y `secuenciaFormaPalabraVertical?`, para asegurar que las letras de la palabra se encuentren respectivamente en la misma fila/columna.
- Se ajustaron algunas comparaciones por desigualdad no estricta (" \leq ") que debían ser estrictas (" $<$ ") en varios puntos.
- En la signatura del observador configuración del TAD `SERVIDOR`, el tipo del repositorio figuraba como de tipo `secu(letra)` pero debía ser de tipo `cola(letra)`.
- Se hicieron algunos ajustes menores de forma.

1. Descripción informal

El objetivo del [TP 1](#) y el [TP 2](#) de la materia va a ser **diseñar** e **implementar en C++** un servidor para un juego de varios jugadores que se alternan ubicando fichas sobre un tablero para formar palabras¹. Resumimos algunas características del juego que vamos a modelar:

1.1. El juego de palabras

- Un **tablero** es una grilla cuadrada de $N \times N$ celdas. El tamaño exacto del tablero depende de la **variante** del juego de la que se trate. Todas las celdas del tablero son iguales; no modelaremos multiplicadores de letra ni de palabra.
- En el juego intervienen K **jugadores**, que juegan por turnos (en ronda).
- Cada jugador tiene siempre en su posesión una cantidad F de **fichas**. La cantidad de fichas puede diferir de acuerdo con la variante del juego.

¹Similar al popular juego de mesa "*Scrabble*".

- Cada ficha tiene asociada una letra del alfabeto y tiene asociado además un **valor**, que queda determinado por la letra en cuestión. El valor de cada letra puede diferir de acuerdo con la variante del juego².
- Inicialmente el tablero se encuentra vacío. A medida que el juego avanza, algunas celdas pasan a estar ocupadas por fichas. Una **ocurrencia de una palabra** en el tablero es una secuencia *maximal* de fichas consecutivas dispuestas en fila o columna en las que se puede leer esa palabra. Cuando decimos “maximal” nos referimos a que no debe existir otra secuencia de fichas consecutivas más larga que la contenga. Por ejemplo, en el siguiente tablero hay ocurrencias de las palabras CALLES, AHONDAN y PONIENTE, pero no hay ocurrencias las palabras CAL, ONDA ni ENTE.

				P			
	C			O			
	A	H	O	N	D	A	N
	L			I			
	L			E			
	E			N			
	S			T			
				E			

- En cada turno un jugador puede elegir un subconjunto (posiblemente vacío) de las fichas que tiene en su posesión y **ubicar** cada una de esas fichas en alguna celda del tablero que todavía no se encuentre ocupada. Para que la jugada sea **válida** deben reunirse las dos condiciones siguientes:

1. Debe existir una ocurrencia de una palabra en el tablero que contenga a todas las fichas ubicadas.

Por ejemplo, un jugador puede ubicar las fichas **M**, **S**, **T**, **E** porque, a pesar de que la **M** está separada de la **S**, forman parte de una única palabra contigua, “**ASOMASTE**”:

T	O	M	A	R	E		
A	S	O	M	A	S	T	E
J	I	R	A	F	A		
A	T	A		A			
B	O	D	E	G	A		
A	S	A		A			

2. Todas las palabras que se forman en el tablero deben ser **legítimas**. El conjunto de palabras legítimas puede diferir de acuerdo con la variante del juego.
- Cada jugador tiene un **puntaje**. Cuando un jugador ubica fichas, gana puntos por cada una de las palabras que se hayan formado en el tablero, es decir, palabras que tengan ocurrencias en el tablero e incluyan al menos una de las fichas ubicadas por el jugador. En el ejemplo de arriba, se deben sumar los puntos de la palabra “**ASOMASTE**” (leída horizontalmente), así como de las palabras “**AMA**”, “**ESA**”, “**T**” y “**E**” (leídas verticalmente).

²Por ejemplo, en algunas variantes en español las fichas con la letra Q siempre tienen valor 5, en tanto que, en algunas variantes en inglés las fichas con la letra Q siempre tienen valor 10.

El puntaje que otorga una palabra se computa como la suma de los puntajes de todas las letras que la componen.

- Después de que un jugador ubica fichas en el tablero y finaliza su turno, repone las fichas que le falten para volver a tener F fichas en su posesión. Las fichas se reponen tomándolas de un **repositorio de fichas**. Un repositorio de fichas es una cola de letras que sigue alguna distribución. El orden de las letras en la bolsa respeta un orden fijo que se determina al inicio del juego. Suponemos que la bolsa se inicializa con al menos $N^2 + K \cdot F$ fichas, lo que alcanza para cubrir totalmente el tablero y las fichas de cada jugador.
- No modelaremos condiciones de finalización del juego. Es decir, a efectos de este TP, el juego puede continuar indefinidamente.

1.2. El servidor

- Un **servidor** se inicializa con una configuración que incluye la variante del juego y la cantidad de jugadores que intervienen.
- Un **cliente** puede conectarse a un servidor. El servidor identifica a cada cliente con un identificador numérico único.
- Cuando se conectaron tantos clientes como los jugadores que intervienen, comienza el juego.
- En cualquier momento, un cliente puede enviar un **mensaje** al servidor. El mensaje del cliente indica su jugada, que incluye las letras que juega con sus respectivas posiciones. El mensaje que envía el cliente al servidor puede representar una jugada válida, pero también puede tratarse de una jugada inválida. Es responsabilidad del servidor verificar si se trata de una jugada válida.
- El servidor comunica respuestas y, más en general, **notificaciones**, a los clientes. El servidor no envía las notificaciones directamente a los clientes, sino que los almacena en una cola de notificaciones destinada específicamente a ese cliente. En cualquier momento un cliente puede consultar los mensajes de su cola de notificaciones, lo que tiene el efecto de vaciar dicha cola. Las posibles notificaciones que el servidor puede comunicar son:

IDCLIENTE (<i>cid</i>)	Evento inicial que comunica al cliente su propio identificador.
EMPEZAR (<i>n</i>)	Evento que indica el inicio del juego sobre un tablero de $n \times n$. Se envía por única vez al inicio del juego.
TURNODE (<i>cid</i>)	Evento que indica que le toca jugar al cliente identificado con el identificador <i>cid</i> .
UBICAR (<i>cid, jugada</i>)	Evento que indica que el cliente identificado por <i>cid</i> jugó ubicando fichas sobre el tablero según lo indicado en <i>jugada</i> .
REPONER (<i>letras</i>)	Envía al jugador letras para reponer su “mano”.
SUMAPUNTOS (<i>cid, n</i>)	Indica que el cliente identificado por <i>cid</i> ganó <i>n</i> puntos.
MAL	Evento que sirve como respuesta al cliente para indicarle que el mensaje enviado es inválido, ya sea porque el juego aún no comenzó, porque no es el turno del jugador o porque la jugada es inválida (el jugador no tiene suficientes letras para hacer esa jugada, porque las posiciones se encuentran fuera de rango, etc.).

2. Especificación formal

Suponemos dado un tipo abstracto de datos LETRA. Asumimos que el tipo LETRA tiene finitos valores, es decir, las letras están en correspondencia biunívoca con un conjunto de la forma $\{0, \dots, A-1\}$. Más precisamente, suponemos que existen funciones:

$$\begin{aligned} \text{ord} & : \text{letra} \longrightarrow \text{nat} \\ \text{ord}^{-1} & : \text{nat } n \longrightarrow \text{letra} \end{aligned} \quad \{n < A\}$$

tales que:

$$\begin{aligned} \text{ord}^{-1}(\text{ord}(\ell)) &\equiv \ell \quad \text{para todo } \ell : \text{letra}, \\ \text{ord}(\text{ord}^{-1}(n)) &\equiv n \quad \text{para todo } n : \text{nat tal que } n < A. \end{aligned}$$

2.1. Especificación formal del juego

2.1.1. Operaciones sobre colas

Extendemos el TAD COLA con las siguientes funciones (en gris las operaciones auxiliares):

$$\begin{aligned} \text{próximoN} & : \text{cola}(\alpha) \ q \times \text{nat } n \longrightarrow \text{multiconj}(\alpha) \quad \{n \leq \text{tamaño}(q)\} \\ \text{próximoN}(q, n) &\equiv \text{if } n = 0 \text{ then } \emptyset \text{ else } \text{Ag}(\text{próximo}(q), \text{próximoN}(\text{desencolar}(q), n - 1)) \text{ fi} \\ \text{desencolarN} & : \text{cola}(\alpha) \ q \times \text{nat } n \longrightarrow \text{cola}(\alpha) \quad \{n \leq \text{tamaño}(q)\} \\ \text{desencolarN}(q, n) &\equiv \text{if } n = 0 \text{ then } q \text{ else } \text{desencolar}(\text{desencolarN}(q, n - 1)) \text{ fi} \end{aligned}$$

2.1.2. Operaciones sobre conjuntos

Extendemos el TAD CONJUNTO con las siguientes funciones:

$$\begin{aligned} \text{partes} & : \text{conj}(\alpha) \longrightarrow \text{conj}(\text{conj}(\alpha)) \\ \text{partes}(c) &\equiv \text{if vacío?}(c) \text{ then } \{\emptyset\} \\ &\quad \text{else } \text{partes}(\text{sinUno}(c)) \cup \text{agregarATodos}(\text{dameUno}(c), \text{partes}(\text{sinUno}(c))) \\ &\quad \text{fi} \\ \text{agregarATodos} & : \alpha \times \text{conj}(\text{conj}(\alpha)) \longrightarrow \text{conj}(\text{conj}(\alpha)) \\ \text{agregarATodos}(x, c) &\equiv \text{if vacío?}(c) \text{ then } \emptyset \\ &\quad \text{else } \text{Ag}(\text{Ag}(x, \text{dameUno}(c)), \text{agregarATodos}(x, \text{sinUno}(c))) \\ &\quad \text{fi} \\ \text{permutaciones} & : \text{conj}(\alpha) \longrightarrow \text{conj}(\text{secu}(\alpha)) \\ \text{permutaciones}(c) &\equiv \text{if vacío?}(c) \text{ then } \{\langle \rangle\} \\ &\quad \text{else } \text{insertarEnTodas}(\text{dameUno}(c), \text{permutaciones}(\text{sinUno}(c))) \\ &\quad \text{fi} \\ \text{insertarEnTodas} & : \alpha \times \text{conj}(\text{secu}(\alpha)) \longrightarrow \text{conj}(\text{secu}(\alpha)) \end{aligned}$$

```

insertarEnTodas( $x, c$ )  $\equiv$  if vacío?( $c$ ) then
     $\emptyset$ 
else
    insertar( $\langle \rangle, x, \text{dameUno}(c)$ )  $\cup$  insertarEnTodas( $x, \text{sinUno}(c)$ )
fi

insertar : secu( $\alpha$ )  $\times$   $\alpha$   $\times$  secu( $\alpha$ )  $\longrightarrow$  conj(secu( $\alpha$ ))
insertar( $\text{prefijo}, x, \text{sufijo}$ )  $\equiv$  { $\text{prefijo} \ \& \ (x \bullet \text{sufijo})$ }  $\cup$ 
    if vacía?( $\text{sufijo}$ ) then
         $\emptyset$ 
    else
        insertar( $\text{prefijo} \circ \text{prim}(\text{sufijo}), x, \text{fin}(\text{sufijo})$ )
    fi

```

2.1.3. Ocurrencia

El TAD OCURRENCIA es un renombre de CONJUNTO(TUPLA(NAT, NAT, LETRA)). Por ejemplo, $\{\langle 2, 5, \text{H} \rangle, \langle 3, 5, 0 \rangle, \langle 4, 5, \text{L} \rangle, \langle 5, 5, \text{A} \rangle\}$ modela una ocurrencia de la palabra HOLA escrita verticalmente en el tablero desde la celda (2, 5) hasta la celda (5, 5). Definimos las siguientes funciones auxiliares para operar con ocurrencias:

```

ocurrenciaFormaPalabra? : ocurrencia  $\longrightarrow$  bool
ocurrenciaFormaPalabra?( $o$ )  $\equiv$  algunaSecuenciaFormaPalabra?(permutaciones( $o$ ))

algunaSecuenciaFormaPalabra? : conj(secu(tupla(nat,nat,letra)))  $\longrightarrow$  bool
algunaSecuenciaFormaPalabra?( $c$ )  $\equiv$   $\neg$ vacío?( $c$ )  $\wedge_L$ 
    (secuenciaFormaPalabra?(dameUno( $c$ ))  $\vee$ 
     algunaSecuenciaFormaPalabra?(sinUno( $c$ )))

secuenciaFormaPalabra? : secu(tupla(nat,nat,letra))  $\longrightarrow$  bool
secuenciaFormaPalabra?( $s$ )  $\equiv$  secuenciaFormaPalabraHorizontal?( $s$ )  $\vee$ 
    secuenciaFormaPalabraVertical?( $s$ )

secuenciaFormaPalabraHorizontal? : secu(tupla(nat,nat,letra))  $\longrightarrow$  bool
secuenciaFormaPalabraHorizontal?( $s$ )  $\equiv$  long( $s$ )  $\leq 1 \vee_L$ 
    ( $\pi_1(\text{prim}(s)) = \pi_1(\text{prim}(\text{fin}(s)))$ )  $\wedge$ 
    ( $\pi_2(\text{prim}(s)) + 1 = \pi_2(\text{prim}(\text{fin}(s)))$ )  $\wedge$ 
    secuenciaFormaPalabraHorizontal?(fin( $s$ )))

secuenciaFormaPalabraVertical? : secu(tupla(nat,nat,letra))  $\longrightarrow$  bool
secuenciaFormaPalabraVertical?( $s$ )  $\equiv$  long( $s$ )  $\leq 1 \vee_L$ 
    ( $\pi_1(\text{prim}(s)) + 1 = \pi_1(\text{prim}(\text{fin}(s)))$ )  $\wedge$ 
    ( $\pi_2(\text{prim}(s)) = \pi_2(\text{prim}(\text{fin}(s)))$ )  $\wedge$ 
    secuenciaFormaPalabraVertical?(fin( $s$ )))

palabraQueFormaLaOcurrencia : ocurrencia  $o$   $\longrightarrow$  secu(letra)
    {ocurrenciaFormaPalabra?( $o$ )}
palabraQueFormaLaOcurrencia( $o$ )  $\equiv$  palabraQueFormaAlgunaSecuencia(permutaciones( $o$ ))
palabraQueFormaAlgunaSecuencia : conj(secu(tupla(nat,nat,letra)))  $c$   $\longrightarrow$  secu(letra)
    {algunaSecuenciaFormaPalabra?( $c$ )}

```

$\text{palabraQueFormaAlgunaSecuencia}(c) \equiv \text{if } \text{secuenciaFormaPalabra?}(\text{dameUno}(c)) \text{ then}$
 $\text{palabraQueFormaLaSecuencia}(\text{dameUno}(c))$
 else
 $\text{palabraQueFormaAlgunaSecuencia}(\text{sinUno}(c))$
 fi

$\text{palabraQueFormaLaSecuencia} : \text{secu}(\text{tupla}(\text{nat}, \text{nat}, \text{letra})) \ s \longrightarrow \text{secu}(\text{letra})$
 $\{\text{secuenciaFormaPalabra?}(s)\}$

$\text{palabraQueFormaLaSecuencia}(s) \equiv \text{if } \text{vacío?}(s) \text{ then}$
 $\langle \rangle$
 else
 $\pi_3(\text{prim}(s)) \bullet \text{palabraQueFormaLaSecuencia}(\text{fin}(s))$
 fi

$\text{ocurrenciasQueFormanPalabrasMaximales} : \text{conj}(\text{ocurrencia}) \longrightarrow \text{conj}(\text{ocurrencia})$
 $\text{ocurrenciasQueFormanPalabrasMaximales}(c) \equiv \text{ocurrenciasMaximales}(\text{ocurrenciasQueFormanPalabras}(c), \text{ocurrenciasQueFormanPalabras}(c))$

$\text{ocurrenciasQueFormanPalabras} : \text{conj}(\text{ocurrencia}) \longrightarrow \text{conj}(\text{ocurrencia})$
 $\text{ocurrenciasQueFormanPalabras}(c) \equiv \text{if } \text{vacío?}(c) \text{ then}$
 \emptyset
 else
 if $\text{ocurrenciaFormaPalabra?}(\text{dameUno}(c)) \text{ then}$
 $\{\text{dameUno}(c)\}$
 else
 \emptyset
 fi $\cup \text{ocurrenciasQueFormanPalabras}(\text{sinUno}(c))$
 fi

$\text{ocurrenciasMaximales} : \text{conj}(\text{ocurrencia}) \times \text{conj}(\text{ocurrencia}) \longrightarrow \text{conj}(\text{ocurrencia})$
 $\text{ocurrenciasMaximales}(c, c_0) \equiv \text{if } \text{vacío?}(c) \text{ then}$
 \emptyset
 else
 if $\text{incluidaEnAlguna?}(\text{dameUno}(c), c_0) \text{ then}$
 \emptyset
 else
 $\{\text{dameUno}(c)\}$
 fi $\cup \text{ocurrenciasMaximales}(\text{sinUno}(c), c_0)$
 fi

$\text{incluidaEnAlguna?} : \text{ocurrencia} \times \text{conj}(\text{ocurrencia}) \longrightarrow \text{bool}$
 $\text{incluidaEnAlguna?}(o, c) \equiv \neg \text{vacío?}(c) \wedge_L (o \subset \text{dameUno}(c) \vee \text{incluidaEnAlguna?}(o, \text{sinUno}(c)))$

$\text{letrasDeLaOcurrencia} : \text{ocurrencia} \longrightarrow \text{multiconj}(\text{letra})$
 $\text{letrasDeLaOcurrencia}(o) \equiv \text{if } \text{vacío?}(o) \text{ then}$
 \emptyset
 else
 $\text{Ag}(\pi_3(\text{dameUno}(o)), \text{letrasDeLaOcurrencia}(\text{sinUno}(o)))$
 fi

TAD TABLERO

7

```

contenidoDelTablero( $t, i, j$ )  $\equiv$  if hayLetra?( $t, i, j$ ) then
    { $\langle i, j, \text{letra}(t, i, j) \rangle$ }
else
     $\emptyset$ 
fi  $\cup$ 
if  $j + 1 < \text{tamaño}(t)$  then
    contenidoDelTablero( $t, i, j + 1$ )
else
    if  $i + 1 < \text{tamaño}(t)$  then
        contenidoDelTablero( $t, i + 1, 0$ )
    else
         $\emptyset$ 
    fi
fi

ponerLetras( $t, o$ )  $\equiv$  if vacío?( $o$ ) then
     $t$ 
else
    ponerLetra(ponerLetras( $t, \text{sinUno}(o)$ ),
         $\pi_1(\text{dameUno}(o)), \pi_2(\text{dameUno}(o)), \pi_3(\text{dameUno}(o))$ )
fi

celdasLibres?( $t, o$ )  $\equiv$  vacío?( $o$ )  $\vee_L$ 
    (libre?( $t, \pi_1(\text{dameUno}(o)), \pi_2(\text{dameUno}(o))$ )  $\wedge$ 
    celdasLibres?( $t, \text{sinUno}(o)$ ))

```

Fin TAD

2.1.5. Variante

TAD VARIANTE

géneros variante

usa BOOL, NAT, SECUENCIA, CONJUNTO, DICCIONARIO

exporta *observadores, generadores*

observadores básicos

tamañoTablero	: variante	\longrightarrow nat
#fichas	: variante	\longrightarrow nat
puntajeLetra	: variante \times letra	\longrightarrow nat
palabraLegítima?	: variante \times secu(letra)	\longrightarrow bool

generadores

nuevaVariante	: nat $n \times$ nat $f \times$ dicc(letra \times nat) \times conj(secu(letra))	\longrightarrow variante
		$\{n > 0 \wedge f > 0\}$

axiomas

($\forall v$: variante, $\forall n, f$: nat, $\forall d$: dicc(letra, nat), $\forall c$: conj(secu(letra)), $\forall l$: letra, $\forall p$: secu(letra))

tamañoTablero(nuevaVariante(n, f, d, c))	$\equiv n$
--	------------

#fichas(nuevaVariante(n, f, d, c))	$\equiv f$
--	------------

$$\text{palabraLegítima?}(\text{nuevaVariante}(n, f, d, c), p) \equiv p \in c$$

TAD JUEGO

géneros	juego
usa	BOOL, NAT, TUPLA, CONJUNTO, MULTICONJUNTO, COLA, LETRA, TABLERO, VARIANTE

exporta *observadores, generadores, otras operaciones*

observadores básicos

variante	: juego	→ variante	
#jugadores	: juego	→ nat	
repositorio	: juego	→ cola(letra)	
tablero	: juego	→ tab	
turno	: juego	→ nat	
fichas	: juego $j \times \text{nat } i$	→ multiconj(letra)	
			$\{i < \#jugadores(j)\}$
puntaje	: juego $j \times \text{nat } i$	→ nat	$\{i < \#jugadores(j)\}$

generadores

$$\begin{array}{ll} \text{nuevoJuego} & : \text{nat } k \times \text{variante } v \times \text{cola}(\text{letra}) \, r \longrightarrow \text{juego} \\ & \left\{ \begin{array}{l} \text{tamaño}(r) \geq \text{tamañoTablero}(v) * \text{tamañoTablero}(v) + k * \# \text{fichas}(v) \wedge \\ k > 0 \end{array} \right\} \\ \text{ubicar} & : \text{juego } j \times \text{ocurrencia } o \longrightarrow \text{juego} \\ & \{ \text{jugadaVálida?}(j, o) \} \end{array}$$

otras operaciones

jugadaVálida?	: juego \times ocurrencia	\rightarrow bool
puntajeObtenido	: juego $j \times$ ocurrencia o	\rightarrow nat
		$\left\{ \begin{array}{l} (\forall i', j': \text{nat})(\forall l: \text{letra}) ((i', j', l) \in o \Rightarrow \\ \quad (\text{ocupada?}(\text{tablero}(j), i', j') \\ \quad \wedge_L \text{letra}(\text{tablero}(j), i', j') = l)) \end{array} \right\}$
palabrasUbicadas	: conj(ocurrencia) \times ocurrencia	\rightarrow conj(ocurrencia)
esPalabraUbicada?	: ocurrencia \times ocurrencia	\rightarrow bool
puntajePalabras	: juego \times tab $t \times$ conj(ocurrencia) os	\rightarrow nat
		$\{os \subseteq \text{ocurrenciasDePalabras}(t)\}$
puntajePalabra	: juego \times tab \times ocurrencia	\rightarrow nat
		$\{(\forall i, j: \text{nat})(\forall l: \text{letra}) ((i, j, l) \in o \Rightarrow (\text{ocupada?}(t, i, j) \wedge_L \text{letra}(t, i, j) = l))\}$
algunaIncluyeJugada?	: conj(ocurrencia) \times ocurrencia	\rightarrow bool

todasLegítimas? : juego \times conj(ocurrencia) \longrightarrow bool

axiomas

$\forall j$: juego, $\forall v$: variante, $\forall k, i, i', j'$: nat, $\forall r$: cola(letra), $\forall os$: conj(ocurrencia), $\forall o, o'$: ocurrencia,
 $\forall t$: tab

variante(nuevoJuego(k, v, r)) $\equiv v$
 variante(ubicar(j, o)) \equiv variante(j)
 #jugadores(nuevoJuego(k, v, r)) $\equiv k$
 #jugadores(ubicar(j, o)) \equiv #jugadores(j)
 repositorio(nuevoJuego(k, v, r)) \equiv desencolarN($r, k * \#fichas(v)$)
 repositorio(ubicar(j, o)) \equiv desencolarN(repositorio(j), $\#(o)$)
 tablero(nuevoJuego(k, v, r)) \equiv nuevoTablero(tamañoTablero(v))
 tablero(ubicar(j, o)) \equiv ponerLetras(tablero(j), o)
 turno(nuevoJuego(k, v, r)) $\equiv 0$
 turno(ubicar(j, o)) \equiv (turno(j) + 1) **mód** #jugadores(j)
 fichas(nuevoJuego(k, v, r), i) \equiv próximosN(desencolarN($r, i * \#fichas(v)$), $\#fichas(v)$)
 fichas(ubicar(j, o), i) \equiv **if** $i = \text{turno}(j)$ **then**
 ($fichas(j, i) - \text{letrasDeLaOcurrencia}(o)$) \cup
 próximosN(repositorio(j), $\#(o)$)
 else
 $fichas(j, i)$
 fi
 puntaje(nuevoJuego(k, v, r), i) $\equiv 0$
 puntaje(ubicar(j, o), i) \equiv **if** $i = \text{turno}(j)$ **then**
 puntajeObtenido(j, o)
 else
 0
 fi + puntaje(j, i)
 puntajeObtenido(j, o) \equiv puntajePalabras(j, t' ,
 palabrasUbicadas(ocurrenciasDePalabras(t'), o))
 donde $t' \equiv$ ponerLetras(tablero(j), o)
 palabrasUbicadas(os, o) \equiv **if** vacío?(os) **then**
 \emptyset
 else
 if esPalabraUbicada?(dameUno(os), o) **then**
 {dameUno(os)}
 else
 \emptyset
 fi \cup palabrasUbicadas(sinUno(os), o)
 fi
 esPalabraUbicada?(o, o') $\equiv \neg$ vacío?($o \cap o'$)

```

puntajePalabras( $j, t, os$ )  $\equiv$  if vacío?( $os$ ) then
    0
else
    puntajePalabra( $j, t, dameUno(os)$ ) +
    puntajePalabras( $j, t, sinUno(os)$ )
fi
puntajePalabra( $j, t, o$ )  $\equiv$  if vacía?( $o$ ) then
    0
else
    puntajeLetra(variante( $j$ ),  $\pi_3(dameUno(o))$ ) +
    puntajePalabra( $j, t, sinUno(o)$ )
fi

jugadaVálida?( $j, o$ )  $\equiv$  vacío?( $o$ )  $\vee$ 
    (celdasLibres?(tablero( $j$ ),  $o$ )  $\wedge_L$ 
    (algunaIncluyeJugada?(ocurrenciasDePalabras( $t'$ ),  $o$ )  $\wedge$ 
    todasLegítimas?( $j$ , ocurrenciasDePalabras( $t'$ ))))
donde  $t' \equiv ponerLetras(tablero(j), o)$ 

algunaIncluyeJugada?( $os, o$ )  $\equiv$   $\neg$  vacío?( $os$ )  $\wedge_L$ 
    ( $o \subseteq dameUno(os)$ )  $\vee$ 
    algunaIncluyeJugada?(sinUno( $os$ ),  $o$ )

todasLegítimas?( $j, os$ )  $\equiv$  vacío?( $os$ )  $\vee_L$ 
    (palabraLegítima?(
        variante( $j$ ),
        palabraQueFormaLaOcurrencia(dameUno( $os$ )))  $\wedge$ 
    todasLegítimas?( $j, sinUno(os)$ ) )

```

Fin TAD

2.2. Especificación formal del servidor

TAD IDCLIENTE es NAT

TAD TIPONOTIFICACIÓN es ENUM(**IdCliente**, **Empezar**, **TurnoDe**, **Ubicar**, **Reponer**, **SumaPuntos**, **Mal**)

TAD NOTIFICACIÓN

géneros notif

usa BOOL, NAT, MULTICONJUNTO, LETRA, OCURRENCIA, IDCLIENTE

exporta *observadores, generadores, otras operaciones*

observadores básicos

datos : notif \longrightarrow tupla(tipoNotif, idCliente, nat, multiconj(letra), ocurrencia)

generadores

IDCLIENTE : idCliente \longrightarrow notif

EMPEZAR : nat \longrightarrow notif

TURNODE : idCliente \longrightarrow notif
UBICAR : idCliente \times ocurrencia \longrightarrow notif
REPONER : multiconj(letra) \longrightarrow notif
SUMAPUNTOS : idCliente \times nat \longrightarrow notif
MAL : \longrightarrow notif

axiomas

datos(**IDCLIENTE**(cid)) $\equiv \langle \text{IdCliente}, cid, 0, \emptyset, \emptyset \rangle$
 datos(**EMPEZAR**(n)) $\equiv \langle \text{Empezar}, 0, n, \emptyset, \emptyset \rangle$
 datos(**TURNODE**(cid)) $\equiv \langle \text{TurnoDe}, cid, 0, \emptyset, \emptyset \rangle$
 datos(**UBICAR**(cid, o,)) $\equiv \langle \text{Ubicar}, cid, 0, \emptyset, o \rangle$
 datos(**REPONER**(f)) $\equiv \langle \text{Reponer}, 0, 0, f, \emptyset \rangle$
 datos(**SUMAPUNTOS**(cid, n)) $\equiv \langle \text{SumaPuntos}, cid, n, \emptyset, \emptyset \rangle$
 datos(**MAL**) $\equiv \langle \text{Mal}, 0, 0, \emptyset, \emptyset \rangle$

Fin TAD

TAD SERVIDOR

géneros servidor

usa NAT, TUPLA, SECUENCIA, LETRA, OCURRENCIA, VARIANTE, JUEGO, IDCLIENTE

exporta observadores, generadores, otras operaciones

observadores básicos

#esperados : servidor \longrightarrow nat
 #conectados : servidor \longrightarrow nat
 configuración : servidor $s \longrightarrow$ tupla(variante, cola(letra))
 juego : servidor $s \longrightarrow$ juego
 notificaciones : servidor $s \times$ idCliente $cid \longrightarrow$ secu(notif)

$\{\neg \text{empezó?}(s)\}$
 $\{\text{empezó?}(s)\}$
 $\{cid < \#conectados(s)\}$

generadores

nuevoServidor : nat $k \times$ variante $v \times$ cola(letra) $r \longrightarrow$ servidor
 $\{\text{long}(r) \geq \text{tamañoTablero}(v) * \text{tamañoTablero}(v) + k * \#fichas(v)\}$
 conectarCliente : servidor \longrightarrow servidor
 $\{\neg \text{empezó?}(s)\}$
 consultar : servidor $s \times$ idCliente $cid \longrightarrow$ servidor
 $\{cid < \#conectados(s)\}$
 recibirMensaje : servidor $s \times$ idCliente $cid \times$ ocurrencia \longrightarrow servidor
 $\{cid < \#conectados(s)\}$

otras operaciones

empezó? : servidor \longrightarrow bool

axiomas

$(\forall s: \text{servidor}, \forall k: \text{nat}, \forall v: \text{variante}, \forall cid, cid': \text{idCliente}, \forall o: \text{ocurrencia}, \forall r: \text{cola}(\text{letra}))$
 #esperados(nuevoServidor(k, v, r)) $\equiv k$
 #esperados(conectarCliente(s)) $\equiv \#esperados(s)$
 #esperados(consultar(s, cid)) $\equiv \#esperados(s)$

$$\begin{aligned}
\#esperados(\text{recibirMensaje}(s, cid, o)) &\equiv \#esperados(s) \\
\#conectados(\text{nuevoServidor}(k, v, r)) &\equiv 0 \\
\#conectados(\text{conectarCliente}(s)) &\equiv 1 + \#conectados(s) \\
\#conectados(\text{consultar}(s, cid)) &\equiv \#conectados(s) \\
\#conectados(\text{recibirMensaje}(s, cid, o)) &\equiv \#conectados(s) \\
\text{configuración}(\text{nuevoServidor}(k, v, r)) &\equiv \langle v, r \rangle \\
\text{configuración}(\text{conectarCliente}(s)) &\equiv \text{configuración}(s) \\
\text{configuración}(\text{consultar}(s, cid)) &\equiv \text{configuración}(s) \\
\text{configuración}(\text{recibirMensaje}(s, cid, o)) &\equiv \text{configuración}(s) \\
\text{juego}(\text{conectarCliente}(s)) &\equiv \text{nuevoJuego}(\#esperados(s), \\
&\quad \pi_1(\text{configuración}(s)), \\
&\quad \pi_2(\text{configuración}(s))) \\
\text{juego}(\text{consultar}(s)) &\equiv \text{juego}(s) \\
\text{juego}(\text{recibirMensaje}(s, cid, o)) &\equiv \text{if } \text{turno}(\text{juego}(s)) = cid \wedge \text{jugadaVálida?}(\text{juego}(s), o) \\
&\quad \text{then} \\
&\quad \text{ubicar}(\text{juego}(s), o) \\
&\quad \text{else} \\
&\quad \text{juego}(s) \\
&\quad \text{fi} \\
\text{notificaciones}(\text{conectarCliente}(s), cid) &\equiv \text{if } cid = \#conectados(s) + 1 \text{ then} \\
&\quad \langle \rangle \circ \text{IDCLIENTE}(cid) \\
&\quad \text{else} \\
&\quad \text{notificaciones}(s, cid) \\
&\quad \text{fi} \\
&\quad \& \\
&\quad \text{if } \#conectados(s) + 1 = \#esperados(s) \text{ then} \\
&\quad \quad \langle \rangle \circ \text{EMPEZAR}(\text{tamañoTablero}(\text{variante}(s))) \\
&\quad \quad \circ \text{TURNODE}(0) \\
&\quad \text{else} \\
&\quad \quad \langle \rangle \\
&\quad \text{fi} \\
\text{notificaciones}(\text{consultar}(s, cid), cid') &\equiv \text{if } cid = cid' \text{ then} \\
&\quad \langle \rangle \\
&\quad \text{else} \\
&\quad \text{notificaciones}(s, cid') \\
&\quad \text{fi} \\
\text{notificaciones}(\text{recibirMensaje}(s, cid, o), cid') &\equiv
\end{aligned}$$

```

    notificaciones( $s$ ,  $cid'$ ) &
    if jugadaVálida?(juego( $s$ ),  $o$ ) then
        ⟨⟩
        ◦ UBICAR( $cid$ ,  $m$ )
        ◦ SUMAPUNTOS( $cid$ , puntaje(ubicar( $j$ ,  $o$ ),  $cid$ ) - puntaje( $j$ ,  $cid$ ))
        &
        if  $cid = cid'$  then
            ⟨⟩ ◦ REPONER(fichas(ubicar( $j$ ,  $o$ ),  $cid$ ) - fichas( $j$ ,  $cid$ ))
        else
            ⟨⟩
        fi
        &
        TURNODE(turno(ubicar( $j$ ,  $o$ )))
    else
        if  $cid = cid'$  then ⟨⟩ ◦ MAL else ⟨⟩ fi
    fi

    empezó?( $s$ ) ≡ #conectados( $s$ ) = #esperados( $s$ )
Fin TAD

```