

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1



Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Integrante	LU	Correo electrónico
Guberman, Diego Andrés	469/17	diego98g@hotmail.com
Ramis Folberg, Ezequiel Leonel	881/21	ezequielramis.hello@gmail.com
Sabetay, Kevin Damian	476/16	kevin.sabetay96@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. Preámbulo	3
2. Módulo Juego	3
2.1. Interfaz	3
2.2. Implementación	4
2.3. Servicios usados	6
3. Módulo Servidor	7
3.1. Interfaz	7
3.2. Implementación	8
3.3. Servicios usados	8
4. Módulos auxiliares	9
4.1. Módulo Letra	9
4.2. Módulo Variante	9
4.2.1. Interfaz	9
4.2.2. Implementación	10
4.3. Módulo Ocurrencia	10
4.3.1. Interfaz	10
4.3.2. Implementación	10
4.4. Módulo Notificación	10
4.4.1. Interfaz	11
4.4.2. Implementación	11

1. Preámbulo

Antes de presentar los módulos, definimos las siguientes variables para las complejidades temporales:

- N — tamaño del tablero.
- K — cantidad de jugadores.
- $|\Sigma|$ — cantidad de letras en el alfabeto.
- F — cantidad de fichas por jugador.
- $L_{\text{máx}}$ — longitud de la palabra legítima más larga definida por la variante del juego de la que se trate.

2. Módulo Juego

2.1. Interfaz

se explica con: JUEGO

géneros: juego

usa: BOOL, NAT, COLA, LETRA, OCURRENCIA, VARIANTE

operaciones:

NUEVOJUEGO(in k : nat, in v : variante, in r : cola(letra)) $\rightarrow res$: juego

Pre $\equiv \{ \text{tamaño}(r) \geq \text{tamañoTablero}(v) * \text{tamañoTablero}(v) + k * \#fichas(v) \wedge k > 0 \}$

Post $\equiv \{ res =_{\text{obs}} \text{nuevoJuego}(k, v, r) \}$

Complejidad: $O(N^2 + |\Sigma|K + FK)$

Descripción: Dada una cantidad de jugadores, una variante de juego y un repositorio de fichas, se inicia un nuevo juego con el tablero vacío.

Aliasing: ??

JUGADAVÁLIDA?(in j : juego, in o : ocurrencia) $\rightarrow res$: bool

Pre $\equiv \{ \text{true} \}$

Post $\equiv \{ res =_{\text{obs}} \text{jugadaVálida?}(j, o) \}$

Complejidad: $O(L_{\text{máx}}^2)$

Descripción: Determina si una jugada es válida.

Aliasing: ??

UBICAR(in/out j : juego, in o : ocurrencia)

Pre $\equiv \{ \text{jugadaVálida}(j, o) \wedge j =_{\text{obs}} J_0 \}$

Post $\equiv \{ j =_{\text{obs}} \text{ubicar}(J_0, o) \}$

Complejidad: $O(m)$, donde m es el número de fichas que se ubican.

Descripción: Ubica un conjunto de fichas en el tablero.

Aliasing: ??

VARIANTE(in j : juego) $\rightarrow res$: variante

Pre $\equiv \{ \text{true} \}$

Post $\equiv \{ res =_{\text{obs}} \text{variante}(j) \}$

Complejidad: $O(1)$

Descripción: Obtiene información sobre la variante del juego.

Aliasing: ??

TURNNO(in j : juego) $\rightarrow res$: nat

Pre $\equiv \{ \text{true} \}$

Post $\equiv \{ res =_{\text{obs}} \text{turnno}(j) \}$

Complejidad: $O(1)$

Descripción: Obtiene al jugador del turno actual.

Aliasing: ??

PUNTAJE(**in** j : juego, **in** i : nat) $\rightarrow res$: nat

Pre $\equiv \{i < \#jugadores(j)\}$

Post $\equiv \{res =_{\text{obs}} \text{puntaje}(j, i)\}$

Complejidad: $O(1 + m \cdot L_{\text{máx}})$, donde m es la cantidad de fichas que ubicó el jugador desde la última vez que se invocó a esta operación.

Descripción: Obtiene el puntaje de un jugador.

Aliasing: ??

ENTABLERO?(**in** J : juego, **in** i : nat, **in** j : nat) $\rightarrow res$: bool

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{enTablero?}(\text{tablero}(J), i, j)\}$

Complejidad: $O(1)$

Descripción: Determina si una coordenada (i, j) está en el rango del tablero.

Aliasing: ??

HAYLETRA?(**in** J : juego, **in** i : nat, **in** j : nat) $\rightarrow res$: bool

Pre $\equiv \{\text{enTablero?}(\text{tablero}(J), i, j)\}$

Post $\equiv \{res =_{\text{obs}} \text{hayLetra?}(\text{tablero}(J), i, j)\}$

Complejidad: $O(1)$

Descripción: Determina si una celda del tablero dada una coordenada (i, j) está ocupada por una letra.

Aliasing: ??

LETRA(**in** J : juego, **in** i : nat, **in** j : nat) $\rightarrow res$: letra

Pre $\equiv \{\text{enTablero?}(\text{tablero}(J), i, j) \wedge_L \text{hayLetra?}(\text{tablero}(J), i, j)\}$

Post $\equiv \{res =_{\text{obs}} \text{letra}(\text{tablero}(J), i, j)\}$

Complejidad: $O(1)$

Descripción: Obtiene el contenido de una celda del tablero dada una coordenada (i, j) .

Aliasing: ??

#LETRA TIENE JUGADOR(**in** j : juego, **in** x : letra, **in** i : nat) $\rightarrow res$: nat

Pre $\equiv \{i < \#jugadores(j)\}$

Post $\equiv \{res =_{\text{obs}} \#(x, \text{fichas}(j, i))\}$

Complejidad: $O(1)$

Descripción: Dada una cierta letra x del alfabeto, conocer cuántas fichas tiene un jugador de dicha letra.

Aliasing: ??

2.2. Implementación

Representación

juego se representa con juego_estr

donde juego_estr es tupla(

```
    tablero: array_dimensionable(array_dimensionable(puntero(letra)))
  , jugadores: array_dimensionable(tupla(
      puntaje: nat
      , cantFichas: array_dimensionable(nat)
    ))
```

```

    , cantFichasTotal: nat
  ))
  , jugadorActual: nat
  , repositorio: cola(letra)
  , variante: variante
)

```

Invariante de Representación

Rep : juego_estr \rightarrow bool

Rep(e) \equiv ($\forall j : \text{jugador}^1$)($j \in {}^2e.\text{jugadores} \Rightarrow$ (
 $\text{long}(j.\text{cantFichas}) = \text{FICHASPORJUGADOR}(e.\text{variante}) \wedge$
 $\text{long}(j.\text{cantFichas}) = \text{DOM}() \wedge_L$
 $\sum_{f \in j.\text{cantFichas}} f = j.\text{cantFichasTotal}$
 $)) \wedge$
 $\text{long}(e.\text{tablero}) = \text{TAMAÑOTABLERO}(e.\text{variante}) \wedge$
 $e.\text{jugadorActual} \leq \text{long}(e.\text{jugadores}) \wedge$
 $(\forall \text{col} : \text{array_dimensionable}(\text{puntero}(\text{letra}))) (\text{col} \in e.\text{tablero} \Rightarrow \text{long}(\text{col}) = \text{long}(e.\text{tablero})) \wedge_L$
 $\sum_{j \in e.\text{jugadores}} j.\text{puntaje} \geq \sum_{i, j < \text{long}(e.\text{tablero})} \text{if } e.\text{tablero}[i][j] \neq \text{NULL}$
 $\text{then PUNTAJELETRA}(e.\text{variante}, *e.\text{tablero}[i][j])$
 $\text{else } 0$
 fi

Función de Abstracción

Abs : juego_estr $e \rightarrow$ juego

{Rep(e)}

Abs(e) =_{obs} J: juego | $e.\text{variante} =_{\text{obs}} \text{variante}(J) \wedge$
 $e.\text{repositorio} =_{\text{obs}} \text{repositorio}(J) \wedge$
 $e.\text{jugadorActual} =_{\text{obs}} \text{turno}(J) \wedge$
 $(\text{long}(e.\text{jugadores}) =_{\text{obs}} \# \text{jugadores}(J) \wedge_L$
 $(\forall i : \text{nat})(i < \text{long}(e.\text{jugadores}) \Rightarrow_L ($
 $\text{PUNTAJE}(e.\text{jugadores}[i]) =_{\text{obs}} \text{puntaje}(J, i)) \wedge$
 $(\forall l : \text{letra})(e.\text{cantFichas}[\text{ORD}(l)] = \#(l, \text{fichas}(J, i)))$
 $)) \wedge$
 $\text{long}(e.\text{tablero}) =_{\text{obs}} \text{tamaño}(\text{tablero}(J)) \wedge_L$
 $(\forall i, j : \text{nat})(\text{enTablero?}(\text{tablero}(J), i, j) \Rightarrow_L ($
 $\text{hayLetra?}(\text{tablero}(J), i, j) \iff e.\text{tablero}[i][j] \neq \text{NULL} \wedge$
 $\text{hayLetra?}(\text{tablero}(J), i, j) \Rightarrow_L \text{letra}(\text{tablero}(J), i, j) =_{\text{obs}} *e.\text{tablero}[i][j]$
 $))$

¹Es un renombre de $\text{tupla}(\text{puntaje} : \text{nat}, \text{cantFichas} : \text{array_dimensionable}(\text{letra}), \text{cantFichasTotal} : \text{nat})$ para simplificar.

²Asumimos que existe la operación pertenece \in del tipo $\text{array_dimensionable}$.

Algoritmos

HACERGUIA(in A : guia, in $parámetroInútil$: Nat) \rightarrow bool

```

1:  $i \leftarrow 0$   $\triangleright$  esto es  $\Theta(1)$ 
2:  $n \leftarrow \text{guia.cantEjercicios}()$   $\triangleright \mathcal{O}(1)$ 
3:  $consultas \leftarrow \text{DICC} \text{VACIO}$ 
4: PREPARARMATE()  $\triangleright \Omega(n^n)$ 
5: while  $i < n$  do
6:   PENSAREJERCICIO(1)
7:   if TENGOCONSULTAS( $i$ ) then
8:     ESCRIBIRCONSULTASEJERCICIO( $i, consultas$ )
9:   else
10:    COMERBIZOCHITO()
11:    COMERBIZOCHITO()
12: for miVariable do
13:   hacer algo
14: return VACIO?(consultas)

```

2.3. Servicios usados

3. Módulo Servidor

3.1. Interfaz

se explica con: SERVIDOR

géneros: servidor

usa: NAT, JUEGO, OCURRENCIA, VARIANTE

operaciones:

NUEVOSEVIDOR(**in** $k : \text{nat}$, **in** $v : \text{variante}$) $\rightarrow res : \text{servidor}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{\exists r : \text{cola}(\text{letra}) \mid res =_{\text{obs}} \text{nuevoServidor}(k, v, r)\}$

Complejidad: $O(N^2 + |\Sigma|K + FK)$

Descripción: Dada una cantidad de jugadores y una variante de juego, se inicia un nuevo servidor y una nueva partida de juego.

Aliasing: ??

CONECTAR(**in/out** $s : \text{servidor}$)

Pre $\equiv \{\neg \text{empezó?}(s) \wedge s =_{\text{obs}} S_0\}$

Post $\equiv \{s =_{\text{obs}} \text{conectarCliente}(S_0)\}$

Complejidad: $O(1)$

Descripción: Conecta un cliente a un servidor.

Aliasing: ??

CONSULTAR(**in/out** $s : \text{servidor}$, **in** $cid : \text{nat}$)

Pre $\equiv \{cid \leq \#conectados(s) \wedge s =_{\text{obs}} S_0\}$

Post $\equiv \{s =_{\text{obs}} \text{consultar}(S_0, cid)\}$

Complejidad: $O(n)$, donde n es la cantidad de mensajes en la cola de dicho cliente.

Descripción: Consulta la cola de notificaciones de un cliente (lo cual vacía dicha cola).

Aliasing: ??

RECIBIR(**in/out** $s : \text{servidor}$, **in** $cid : \text{nat}$, **in** $o : \text{ocurrencia}$)

Pre $\equiv \{cid \leq \#conectados(s) \wedge s =_{\text{obs}} S_0\}$

Post $\equiv \{s =_{\text{obs}} \text{recibirMensaje}(S_0, cid, o)\}$

Complejidad: ??

Descripción: Recibe un mensaje de un cliente.

Aliasing: ??

CLIENTES ESPERADOS(**in** $s : \text{servidor}$) $\rightarrow res : \text{nat}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \#esperados(s)\}$

Complejidad: $O(1)$

Descripción: Obtiene el número de clientes esperados.

Aliasing: ??

CLIENTES CONECTADOS(**in** $s : \text{servidor}$) $\rightarrow res : \text{nat}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \#conectados(s)\}$

Complejidad: $O(1)$

Descripción: Obtiene el número de clientes conectados.

Aliasing: ??

PARTIDA(**in** s : **servidor**) $\rightarrow res$: **juego**
Pre $\equiv \{\text{true}\}$
Post $\equiv \{res =_{\text{obs}} \text{juego}(s)\}$
Complejidad: $O(1)$
Descripción: Obtiene el juego que se está jugando en el servidor.
Aliasing: ??

3.2. Implementación

Representación

servidor se representa con **servidor_estr**

donde **servidor_estr** es **tupla**(
 juego: **juego**
 , **jugadoresConectados**: **nat**
 , **jugadoresEsperados**: **nat**
 , **notificaciones**: **array_dimensionable**(cola(notif))
)

Invariante de Representación

Función de Abstracción

Algoritmos

3.3. Servicios usados

4. Módulos auxiliares

4.1. Módulo Letra

Se asume una implementación acorde¹ al módulo de género **letra** con las siguientes operaciones en la interfaz (todas con orden de complejidad $O(1)$):

- **DOM** : $\rightarrow \text{nat}$ — Tamaño del dominio del tipo **letra**. Corresponde con la variable A de su especificación.
- **ORD** : **letra** $\rightarrow \text{nat}$ — Dada una letra, devuelve su correspondiente índice.
- **ORD**⁻¹ : $\text{nat } n \rightarrow \text{letra } \{n < A\}$ — Dado un índice, devuelve su correspondiente letra.

4.2. Módulo Variante

4.2.1. Interfaz

se explica con: **VARIANTE**

géneros: **variante**

usa: ??

operaciones:

NUEVAVARIANTE(
 in $n : \text{nat}$,
 in $f : \text{nat}$,
 in $\text{puntajes} : \text{dicc}^2(\text{letra}, \text{nat})$,
 in $\text{legítimas} : \text{conj}^3(\text{secu}(\text{letra}))$
) $\rightarrow \text{res} : \text{variante}$
Pre $\equiv \{n > 0 \wedge f > 0\}$
Post $\equiv \{\text{res} =_{\text{obs}} \text{nuevaVariante}(n, f, \text{puntajes}, \text{legítimas})\}$
Complejidad: $O(\#\text{legítimas} \cdot L_{\text{máx}})$
Descripción: Genera una variante de juego.
Aliasing: ??

TAMAÑO TABLERO(**in** $v : \text{variante}$) $\rightarrow \text{res} : \text{nat}$
Pre $\equiv \{\text{true}\}$
Post $\equiv \{\text{res} =_{\text{obs}} \text{tamañoTablero}(v)\}$
Complejidad: $O(1)$
Descripción: Devuelve el tamaño del tablero.
Aliasing: ??

FICHAS POR JUGADOR(**in** $v : \text{variante}$) $\rightarrow \text{res} : \text{nat}$
Pre $\equiv \{\text{true}\}$
Post $\equiv \{\text{res} =_{\text{obs}} \#\text{fichas}(v)\}$
Complejidad: $O(1)$
Descripción: Devuelve la cantidad de fichas que debe de tener cada jugador.
Aliasing: ??

PUNTAJE LETRA(**in** $v : \text{variante}$, **in** $l : \text{letra}$) $\rightarrow \text{res} : \text{nat}$
Pre $\equiv \{\text{true}\}$
Post $\equiv \{\text{res} =_{\text{obs}} \text{puntajeLetra}(v, l)\}$
Complejidad: $O(1)$

¹Una buena opción es usar un **Enumerado**.

²No se refiere al género del módulo Diccionario Lineal debido a que se necesita que las operaciones de búsqueda sean $O(1)$.

³No se refiere al género del módulo Conjunto Lineal debido a que se necesita que las operaciones de búsqueda sean $O(L_{\text{máx}})$.

Descripción: Devuelve el puntaje de una letra.

Aliasing: ??

$\text{PALABRALEGÍTIMA?}(\text{in } v : \text{variante}, \text{in } l : \text{secu}(\text{letra})) \rightarrow res : \text{bool}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{palabraLegítima}(v, l)\}$

Complejidad: $O(L_{\text{máx}})$

Descripción: Determina si una palabra es legítima dentro de la variante de juego.

Aliasing: ??

$\text{LONGPALABRAMÁSLARGA}(\text{in } v : \text{variante}) \rightarrow res : \text{nat}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{$

$(\exists p : \text{secu}(\text{letra}))(res =_{\text{obs}} \text{long}(p) \wedge \text{palabraLegítima?}(v, p) \wedge$
 $(\forall p_2 : \text{secu}(\text{letra}))(\text{palabraLegítima?}(v, p_2) \Rightarrow res \geq \text{long}(p_2)))$

$\}$

Complejidad: $O(1)$

Descripción: Obtiene la longitud de la palabra legítima más larga de la variante.

Aliasing: ??

4.2.2. Implementación

4.3. Módulo Ocurrencia

4.3.1. Interfaz

se explica con: OCURRENCIA

géneros: ocurrencia

usa: ??

operaciones:

$\text{FORMAPALABRA?}(\text{in } o : \text{ocurrencia}) \rightarrow res : \text{bool}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{ocurrenciaFormaPalabra?}(o)\}$

Complejidad: $O(L_{\text{máx}}^2)$

Descripción: Determina si una ocurrencia forma una palabra.

Aliasing: ??

$\text{PALABRAQUEFORMA}(\text{in } o : \text{ocurrencia}) \rightarrow res : \text{secu}(\text{letra})$

Pre $\equiv \{\text{ocurrenciaFormaPalabra?}(o)\}$

Post $\equiv \{res =_{\text{obs}} \text{palabraQueFormaLaOcurrencia}(o)\}$

Complejidad: $O(L_{\text{máx}})$

Descripción: Obtiene la palabra formada por una ocurrencia.

Aliasing: ??

4.3.2. Implementación

4.4. Módulo Notificación

Se asume una implementación acorde⁴ al módulo TIPONOTIFICACIÓN de género tipoNotif para usarse en este módulo.

⁴Una buena opción es usar un **Enumerado**.

4.4.1. Interfaz

se explica con: NOTIFICACIÓN

géneros: notif

usa: ??

operaciones:

IDCLIENTE(in $cid : \text{nat}$) $\rightarrow res : \text{notif}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{IDCLIENTE}(cid)\}$

Complejidad: $O(1)$

Descripción: Genera una notificación de tipo **IDCLIENTE**.

Aliasing: ??

EMPEZAR(in $n : \text{nat}$) $\rightarrow res : \text{notif}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{EMPEZAR}(n)\}$

Complejidad: $O(1)$

Descripción: Genera una notificación de tipo **EMPEZAR**.

Aliasing: ??

TURNODE(in $cid : \text{nat}$) $\rightarrow res : \text{notif}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{TURNODE}(cid)\}$

Complejidad: $O(1)$

Descripción: Genera una notificación de tipo **TURNODE**.

Aliasing: ??

UBICAR(in $cid : \text{nat}$, in $o : \text{ocurrencia}$) $\rightarrow res : \text{notif}$

Pre $\equiv \{\text{true}\}$

Post $\equiv \{res =_{\text{obs}} \text{UBICAR}(cid, o)\}$

Complejidad: $O(1)$

Descripción: Genera una notificación de tipo **UBICAR**.

Aliasing: ??

4.4.2. Implementación