



ORGANIZACIÓN DE COMPUTADORAS  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Segundo Cuatrimestre de 2019



<b>Comisión N°: 16.</b>	<b>Ayudante que corrigió: Francisco Gómez Volonterio.</b>	<b>Resultado: Desaprobado.</b> <b>Nota sugerida: D.</b>
-------------------------	---	--

**Entrega Parcial N° 1 (TDA Lista): Aprobado.**

Ítem	Resultado	Observaciones
Headers: Respeta encabezado definido. Compilación :: sin warnings.	OK	-
Tester :: casos de prueba significativos.	ERROR	- No incorpora todos los casos de prueba.
Tester :: resultado casos de prueba.	OK	-
Tester :: resultado casos de prueba indicados por la cátedra.	ERROR	- No incorpora casos de prueba en error.
Código fuente :: manejo de memoria dinámica en l_insertar.	OK	-
Código fuente :: manejo de memoria dinámica en l_eliminar.	ERROR	- Utiliza de forma ineficiente a la función l_fin(). - No invalida los punteros de las estructuras eliminadas, cuando esto sería adecuado.
Código fuente :: manejo de memoria dinámica en l_destruir.	ERROR	- Utiliza de forma ineficiente a la función l_fin(). - No invalida los punteros de las estructuras eliminadas, cuando esto sería adecuado.



ORGANIZACIÓN DE COMPUTADORAS  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Segundo Cuatrimestre de 2019



**Entrega Parcial N° 2 (TDA Árbol): Desaprobado.**

Ítem	Resultado	Observaciones
Headers: Respeta encabezado definido. Compilación :: sin warnings.	ERROR	- La compilación indica diversos warnings.
Tester :: casos de prueba significativos.	ERROR	- No se testean inserciones en posiciones nh != NULL.
Tester :: resultado casos de prueba.	OK	-
Tester :: resultado casos de prueba indicados por la cátedra.	OK	- - -
Código fuente :: manejo de memoria dinámica en a_insertar.	OK	- - - -
Código fuente :: manejo de memoria dinámica en a_eliminar.	ERROR	- No se mantiene consistencia en la lista de hijos al eliminar un dado nodo. - -
Código fuente :: manejo de memoria dinámica en a_destruir.	SIN CONTROLAR	- - - -
Código fuente :: manejo de memoria dinámica en a_sub_arbol.	SIN CONTROLAR	- -



ORGANIZACIÓN DE COMPUTADORAS  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Segundo Cuatrimestre de 2019



**Entrega Final**

**Sobre las condiciones de entrega**

Ítem	Resultado	Observaciones
Entrega a término	ERROR	- El asunto del e-mail no se corresponde con el solicitado por el enunciado.
Respetar estructura requerida	ERROR	- No respeta la estructura de carpetas solicitadas. - La documentación no fue presentada en formato PDF. - Se adjuntan los archivos correspondientes al proyecto de la IDE Codeblock. - <b>Importante:</b> el proyecto estará automáticamente desaprobado en la reentrega en caso de no cumplimentar con las restricciones especificadas.

**Sobre la documentación**

TDA	Resultado	Observaciones
TDA Lista	ERROR	- La descripción de <i>crear_lista()</i> es imprecisa: ¿cómo se crea la lista? ¿Vacía? ¿Qué significa declarar una variable y asignarle memoria? - La descripción de <i>l_anterior()</i> no se corresponde con una descripción de alto nivel acerca de cómo la implementación resuelve la operación.
TDA Arbol	OK	-
TaTeTi	ERROR	- Documentación ausente. - No describe y/o introduce la estrategia de resolución para la dinámica del juego. - No describe cada uno de los módulos considerados para la resolución, indicando su funcionalidad y responsabilidades.
Otros	ERROR	- El formateo del documento debe ser mejorado. - Se abusa de la indentación para listar las diferentes firmas de las operaciones, junto con la descripción de las mismas. En pasajes, se observa que la mitad de la hoja está inutilizada por la exagerada indentación. - Las imágenes incorporadas ilustran escenarios relevantes, sin embargo, se deben dar con mayor orden y prolijidad: las flechas deben indicar desde dónde parten y hacia dónde van; las celdas correspondientes a las listas deberían estar de forma consecutiva, el tamaño de los diagramas podría fácilmente ser reducido, etc.



### Sobre los TDAs solicitados

TDA	Ítem	Resultado	Observaciones
TDA Lista	Respeto ED	OK	<ul style="list-style-type: none"><li>- En <code>l_destruir()</code>, es <code>(*l)</code> quien se debe hacer nulo, y no <code>l</code>.</li><li>- La función <code>destruirREC()</code> es incorrecta. Si bien la lógica del planteo parece correcta, el uso del bucle <code>while</code> provoca un llamado recursivo a cada celda de la lista, por cada celda que se está pasando como parámetro.</li><li>- En la función <code>l_anterior()</code> se hace uso de la función <code>l_primera()</code> lo cual es incorrecto. Dentro del TDA Lista, se debe hacer uso de la estructura interna del TDA, puesto que no viola el encapsulamiento. Además, falta verificar si la posición pasada por parámetro pertenece a la lista.</li></ul>
	Funcionalidad	ERROR	
TDA Árbol	Respeto ED	OK	<ul style="list-style-type: none"><li>- Declara variables dentro de bloques de código.</li><li>- En la función <code>a_raiz()</code>, no se verifica por la nulidad del árbol pasado por parámetro.</li><li>- La función <code>a_destruir()</code> sólo destruye la raíz del árbol, dando lugar a <i>memory leaks</i>.</li></ul>
	Funcionalidad	ERROR	
Partida	Nueva partida	OK	<ul style="list-style-type: none"><li>- Al momento de crear la partida, no se implementa la elección de un jugador aleatorio para comenzar el juego.</li><li>- En la función <code>nuevo_movimiento()</code>, no se hace la verificación del estado de la partida (empate, gana_1 o gana_2).</li><li>- En la función <code>finalizar_partida()</code> se hace <code>free()</code> de los nombres de jugadores, lo cual es incorrecto, ya que estos dos campos son arreglos, no punteros.</li></ul>
	Nuevo movimiento	ERROR	
	Finalizar	ERROR	
IA	Sucesores min-max	OK	<ul style="list-style-type: none"><li>- Cuando se destruye la búsqueda adversaria, no se libera la memoria de la estructura.</li><li>- En la función <code>estados_sucesores()</code> se realiza una clonación de más.</li><li>- La función <code>proximo_movimiento()</code> está parcialmente implementada.</li></ul>
	Valor utilidad	OK	
	Estados sucesores	ERROR	
	Clonar estados	OK	
	Próximo movimiento	ERROR	
	Destruir	ERROR	



ORGANIZACIÓN DE COMPUTADORAS  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Segundo Cuatrimestre de 2019



### Sobre la aplicación

TDA	Ítem	Resultado	Observaciones
Usuario vs. Usuario	Funcionamiento	ERROR	- La partida no finaliza cuando hay un estado ganador. -
Usuario vs. IA	Funcionamiento	ERROR	- La partida no inicia (segmentation Fault). -

### Otros

#### **Observaciones generales**

- FJ: Se recomienda tomar contacto de forma **urgente** con el personal docente para debatir las correcciones indicadas, debido a que el estado de avance del proyecto presentado no es el adecuado para estas instancias. En particular, se observan errores **moderados** y **graves** que, de no ser resueltos, imposibilitarán la aprobación del proyecto.

- En gran parte del proyecto, se realiza la definición de variables por todas partes del código, lo que no es una buena técnica de programación. En este sentido, es esperado que todas las declaraciones se realicen al comenzar la definición de una función o procedimiento, para mejorar la legibilidad del código fuente.

- Se recomienda verificar la indentación del código, adoptando un estándar y siguiéndolo a lo largo de todos los archivos del proyecto.