



Universidad Tecnológica Centroamericana
UNITEC

Laboratorio de Programación I

Laboratorio #6

Diego Madrid

21 de noviembre del 2025



Desarrollo de la práctica

Para el desarrollo satisfactorio del laboratorio, siga al pie de la letra cada instrucción que a continuación se le presenta.

Cree un nuevo proyecto Java con el nombre **Lab#P1_PrimerNombrePrimerApellido** (**Lab3P1_DiegoMadrid**). Recuerde hacer su código robusto, tabulado y estar autodocumentado, la falta de alguna de éstas, incluyendo el nombre, será reflejado en su nota. **También debe crear un repositorio en GitHub con el mismo formato de nombre.**

Objetivos del laboratorio

- ✚ Repasar los temas estudiados en el parcial.
- ✚ Manejo de arreglos.

Consideraciones

- ✚ Su programa debe tener un menú correspondiente a cada ejercicio, entonces serían 3 opciones (1,2,3). La opción 0 deberá ser para salir del programa y, por lo tanto, la ejecución terminará. El menú deberá ser elaborado con el ciclo while.
- ✚ Cada opción en el menú deberá llevar a métodos, es decir, las opciones del menú no tendrán código extensivo con enfoque en llamar el método del ejercicio respectivo.
- ✚ **Para este laboratorio deberá realizar 4 commits para mostrar sus avances.**
- ✚ Si en cualquier momento del laboratorio usted tiene una duda, hágasela saber a los instructores.
- ✚ Uso de variables con nombres significativos.
- ✚ **La copia de código o uso de IA será penalizada con la máxima penalización en la nota del laboratorio y será remitido al Comité de Ética.**
- ✚ Seguir instrucciones de cada ejercicio. Haga las validaciones necesarias.
- ✚ **Si usted en su estudio personal descubren métodos que no se vieron en clases, pregunte a su instructor si los puede utilizar. De lo contrario, se verá reflejado en su nota.**
- ✚ **NO SE ACEPTARÁN LABORATORIOS ENVIADOS AL CORREO DEL INSTRUCTOR, a menos que se haya hablado con el mismo previamente.**

Usuario(s) de GitHub a enviar el laboratorio

- ✚ Tatiana-Garcia
- ✚ SaiasMadrid



Ejercicio Práctico 1 – Ahorcado en Java (3.75 Puntos).

La compañía de juegos CrustedFoods™ ha visto su admirable desempeño en sus laboratorios anteriores, y ha decidido contratarlo para liderar el desarrollo de su nuevo juego en Java, Ahorcado.

El usuario elegirá una opción de dificultad al seleccionar el ejercicio, **fácil (cadenas de 7 caracteres, 8 vidas)** o **difícil (cadenas de 12 caracteres, 10 vidas)**. Al elegir la dificultad, se elegirá una cadena de manera aleatoria de un arreglo de palabras predeterminadas, **String[] palabrasFacil** o **String[] palabrasDificil** respectivamente.

El usuario solamente verá en consola el número de vidas restates y una cantidad de '_' igual al número de caracteres que tiene la cadena. Solo se podrá ingresar un carácter a la vez, si el carácter está en la cadena elegida, se reemplazará su espacio de '_' por la letra que va en ese lugar, en caso de haber más de una instancia de la misma letra en la palabra, se reemplazarán todos los espacios por la letra correcta. (Pista, el espacio de '_' es un arreglo de chars).

- Si el usuario ingresa un carácter que no pertenece a la palabra, se le restará una vida.
- El ejercicio terminará cuando el usuario haya completado la palabra exitosamente o se haya quedado con 0 vidas restantes.

String[] palabrasFacil = {"hangman", "scanner", "validar", "ceviche", "strings", "3arrays", "algebra", "program", "calculo", "chatgpt", "pokemon", "strokes", "11yonce"};

String[] palabrasDificil = {"abc123bac321", "crabominable", "siguatepeque", "cadenascon12", "hollowknight", "thebattlecat", "ventiladores", "programacion", "eloteconatol", "alvaritodiaz"};

Ejemplo de Ejecución:



```
-----[ MENU ]-----
0) Salir
1) Ahorcado en Java
2) Producto Punto de Dos Vectores
3) Turing
Ingrese su seleccion:
1
Ingrese la dificultad del juego (1. Facil, 2. Dificil): 1
Vidas: 8
Progreso: _____
Ingrese un caracter: a
a no es parte de la cadena

Vidas: 7
Progreso: _____
Ingrese un caracter: e
e es parte de la cadena

Vidas: 7
Progreso: _e____e
Ingrese un caracter: i
i es parte de la cadena

Vidas: 7
Progreso: _e_i__e
Ingrese un caracter: c
c es parte de la cadena
```

```
Vidas: 7
Progreso: ce_ic_e
Ingrese un caracter: h
h es parte de la cadena

Vidas: 7
Progreso: ce_iche
Ingrese un caracter: v
v es parte de la cadena

Felicidades, ganaste esta ronda en modo facil, la cadena era: ceviche
```

```
Vidas: 1
Progreso: programac_on
Ingrese un caracter: d
d no es parte de la cadena

Perdiste, la palabra era: programacion
```



Ejercicio Práctico 2 – Producto Punto de dos Vectores (2 Puntos).

Un día usted se levantó y decidió adelantar temas de su carrera de ingeniería, y pensó que Algebra Lineal es un buen lugar para empezar. El tema en cuestión es el producto punto de un vector, definido de la siguiente manera:

$$\vec{A} = (A_X, A_Y, A_Z)$$

$$\vec{B} = (B_X, B_Y, B_Z)$$

$$\vec{A} \cdot \vec{B} = (A_X B_X + A_Y B_Y + A_Z B_Z)$$

El ejercicio leerá dos arreglos de enteros, el usuario va a definir el tamaño de los arreglos, los cuales no pueden ser menores a 1 (NOTA: AMBOS ARREGLOS DEBEN TENER EL MISMO TAMAÑO) y retorna el resultado del producto punto entre esos dos arreglos. Los números de los arreglos serán generados aleatoriamente con un límite de 99.

Ejemplo de Ejecución:

```
-----[ MENU ]-----
0) Salir
1) Ahorcado en Java
2) Producto Punto de Dos Vectores
3) Turing
Ingrese su seleccion:
2
Ingrese el tamaño de los arreglos: 4
El producto punto de [ 14 72 8 69 ] y [ 61 58 9 69 ] es: 9863
```

```
-----[ MENU ]-----
0) Salir
1) Ahorcado en Java
2) Producto Punto de Dos Vectores
3) Turing
Ingrese su seleccion:
2
Ingrese el tamaño de los arreglos: 3
El producto punto de [ 0 65 1 ] y [ 54 34 0 ] es: 2210
```



Ejercicio Práctico 3 – Turing (2 Puntos).

Elabore una opción en la que el programa le pida al usuario que ingrese un entero, el cual será el tamaño de un arreglo que se debe llenar con números aleatorios del 0 al 9. Deberá desarrollar un método que reciba un string, el cual representa instrucciones, y un arreglo. Este método retornará un string que fue generado con estas instrucciones.

En el método usted debe interpretar la cadena de instrucciones. Al iniciar, el método contiene una variable de tipo entero llamada apuntador, que se debe inicializar en 0, es decir, apunta al primer elemento del arreglo.

Las instrucciones pueden ser las siguientes:

- R: Mueva el puntero una casilla a la derecha.
- L: Mueva el puntero una casilla a la izquierda.
- X: Agregue el elemento al que está apuntando el apuntador a la cadena final (la que retorna el método).

Debe asegurarse de validar que el apuntador no se pase de los límites del arreglo.

Ejemplo de Ejecución:

```
-----[ MENU ]-----
0) Salir
1) Ahorcado en Java
2) Producto Punto de Dos Vectores
3) Turing
Ingrese su seleccion:
3
Ingrese el tamaño de su arreglo
5
Arreglo generado: [7][2][9][8][2]
(El puntero empieza en indice 0)
R: Mueva el puntero una casilla a la derecha
L: Mueva el puntero una casilla a la izquierda
X: Agregar el elemento apuntado a la cadena final
Ingrese la cadena de instrucciones a enviar (ej. RRRXLX): XXRRRRXXLLX
Cadena generada: 77229
```





Tabla ASCII

Caracteres ASCII de control			Caracteres ASCII imprimibles			ASCII extendido (Página de código 437)		
00	NULL	(carácter nulo)	32	espacio	64	@	96	`
01	SOH	(inicio encabezado)	33	!	65	A	97	a
02	STX	(inicio texto)	34	"	66	B	98	b
03	ETX	(fin de texto)	35	#	67	C	99	c
04	EOT	(fin transmisión)	36	\$	68	D	100	d
05	ENQ	(consulta)	37	%	69	E	101	e
06	ACK	(reconocimiento)	38	&	70	F	102	f
07	BEL	(timbre)	39	'	71	G	103	g
08	BS	(retroceso)	40	(72	H	104	h
09	HT	(tab horizontal)	41)	73	I	105	i
10	LF	(nueva línea)	42	*	74	J	106	j
11	VT	(tab vertical)	43	+	75	K	107	k
12	FF	(nueva página)	44	,	76	L	108	l
13	CR	(retorno de carro)	45	-	77	M	109	m
14	SO	(desplaza afuera)	46	.	78	N	110	n
15	SI	(desplaza adentro)	47	/	79	O	111	o
16	DLE	(esc.vínculo datos)	48	0	80	P	112	p
17	DC1	(control disp. 1)	49	1	81	Q	113	q
18	DC2	(control disp. 2)	50	2	82	R	114	r
19	DC3	(control disp. 3)	51	3	83	S	115	s
20	DC4	(control disp. 4)	52	4	84	T	116	t
21	NAK	(conf. negativa)	53	5	85	U	117	u
22	SYN	(inactividad sínc)	54	6	86	V	118	v
23	ETB	(fin bloque trans)	55	7	87	W	119	w
24	CAN	(cancelar)	56	8	88	X	120	x
25	EM	(fin del medio)	57	9	89	Y	121	y
26	SUB	(sustitución)	58	:	90	Z	122	z
27	ESC	(escape)	59	;	91	[123	{
28	FS	(sep. archivos)	60	<	92	\	124	
29	GS	(sep. grupos)	61	=	93]	125	}
30	RS	(sep. registros)	62	>	94	^	126	~
31	US	(sep. unidades)	63	?	95	_		
127	DEL	(suprimir)						
128	Ç		160	á		192	ó	
129	ü		161	í		193	¸	
130	é		162	ó		194	¸	
131	à		163	ú		195	¸	
132	ä		164	ñ		196	—	
133	å		165	Ñ		197	¸	
134	â		166	ª		198	¸	
135	ç		167	º		199	À	
136	ê		168	¿		200	¸	
137	ë		169	®		201	¸	
138	è		170	¬		202	¸	
139	ï		171	½		203	¸	
140	ì		172	¼		204	¸	
141	í		173	¸		205	¸	
142	Ä		174	«		206	¸	
143	Å		175	»		207	¸	
144	É		176	¸		208	¸	
145	æ		177	¸		209	¸	
146	Æ		178	¸		210	¸	
147	ø		179	¸		211	¸	
148	ö		180	¸		212	¸	
149	õ		181	À		213	¸	
150	ù		182	Á		214	¸	
151	û		183	Â		215	¸	
152	ü		184	Ã		216	¸	
153	Ö		185	¸		217	¸	
154	Ü		186	¸		218	¸	
155	ø		187	¸		219	¸	
156	ø		188	¸		220	¸	
157	ø		189	¸		221	¸	
158	x		190	¥		222	¸	
159	f		191	¸		223	¸	
						224	¸	
						225	¸	
						226	¸	
						227	¸	
						228	¸	
						229	¸	
						230	¸	
						231	¸	
						232	¸	
						233	¸	
						234	¸	
						235	¸	
						236	¸	
						237	¸	
						238	¸	
						239	¸	
						240	¸	
						241	¸	
						242	¸	
						243	¸	
						244	¸	
						245	¸	
						246	¸	
						247	¸	
						248	¸	
						249	¸	
						250	¸	
						251	¸	
						252	¸	
						253	¸	
						254	¸	
						255	nbsp	

[www.
elCodigoAscii
.com.ar](http://www.elCodigoAscii.com.ar)

los más consultados

\	barra invertida (alt + 92)
@	arroba (alt + 64)
ñ	eñe minúscula (alt + 164)
'	comilla simple, apóstrofe (alt + 39)
#	signo numeral (alt + 35)
!	signo de admiración (alt + 33)
_	guión bajo, subrayado (alt + 95)
*	asterisco (alt + 42)
~	equivalencia, tilde (alt + 126)
-	guión medio (alt + 45)

de uso frecuente (idioma español)	vocales con acento (acento agudo español)	vocales con diéresis	símbolos matemáticos	símbolos comerciales	comillas, llaves paréntesis
ñ alt + 164	á alt + 160	ä alt + 132	½ alt + 171	\$ alt + 36	" alt + 34
Ñ alt + 165	é alt + 130	ë alt + 137	¼ alt + 172	£ alt + 156	' alt + 39
@ alt + 64	í alt + 161	ï alt + 139	¾ alt + 243	¥ alt + 190	(alt + 40
¿ alt + 168	ó alt + 162	ö alt + 148	¹ alt + 251	¢ alt + 189) alt + 41
? alt + 63	ú alt + 163	ü alt + 129	² alt + 252	¤ alt + 207	[alt + 91
¡ alt + 173	Á alt + 181	Ä alt + 142	³ alt + 253	© alt + 169] alt + 93
! alt + 33	É alt + 144	È alt + 211	f alt + 159	© alt + 184	{ alt + 123
: alt + 58	Í alt + 214	Ï alt + 216	± alt + 241	ª alt + 166	} alt + 125
/ alt + 47	Ó alt + 224	Ö alt + 153	× alt + 158	º alt + 167	« alt + 174
\ alt + 92	Ú alt + 233	Ü alt + 154	÷ alt + 246	° alt + 248	» alt + 175

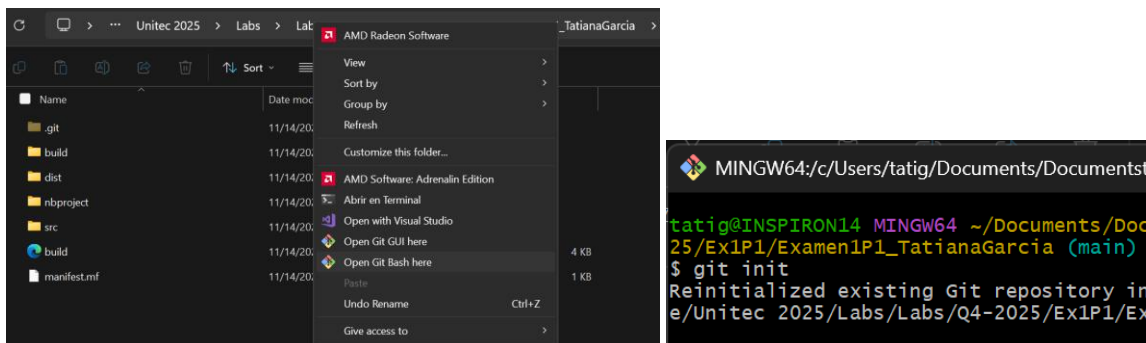
Ponderación

Elemento	Puntaje
Ejercicio 1	3.5 Puntos
Ejercicio 2	3 Puntos
Seguir instrucciones	1 Punto
Total	7.5 Puntos

Comentarios adicionales

El uso de variables significativas será considerado en cada ejercicio. Si desea utilizar algo que usted aprendió en su estudio individual, **consulte con un instructor** antes de usarlo. Recuerde invitar a los instructores como colaboradores, crear su repositorio en privado y subir su laboratorio a Canvas (link del repositorio) dentro del tiempo establecido. **El no seguir las instrucciones tal y como se especifican, se verá reflejado en su nota.**

ICM Activación



Recuerden desactivar el Internet antes de activar el ICM luego:

```
tatig@INSPIRON14 MINGW64 ~/Documents/DocumentstoSave/Unitec 2025/Labs/Labs/Q4-20
25/Ex1P1/Examen1P1_TatianaGarcia (main)
$ ICM
```

ICM Desactivación y Entrega

- Control + C para desactivar el ICM.
- Luego van a buscar el ICM.tgz en su computador. Por lo general está en la dirección C:\Users\nombre\ICM.tgz o en Home/Archivos Recientes.
- Copiar el archivo.tgz y pegarlo en su proyecto de donde hicieron git bash anteriormente y hacer commit.