

EVALUACIÓN T3
ANÁLISIS DE ALGORITMOS Y ESTRATEGIAS DE PROGRAMACIÓN
INGENIERÍA DE SISTEMAS COMPUTACIONALES
SEMESTRE 2025-2

CÓDIGO DEL ESTUDIANTE: n00400647

NÚMERO DE CLASE: _____

APELLIDOS Y NOMBRES DEL ESTUDIANTE : Soldevilla Rios Piero Alexander FECHA: 14/11/2025

DOCENTE: Jorge Bojórquez

Duración del examen: 60 min

INDICACIONES IMPORTANTES QUE DEBE LEER EL Y FIRMAR EL ESTUDIANTE

1. Identificarse con la presentación de su ID Card (colocarlo encima de su carpeta para la verificación)
2. Escribir con letra legible y cuide su redacción en la resolución del examen.
3. Una vez iniciada la evaluación, queda prohibido salir del aula.
4. Cumplir con los artículos del Reglamento de Disciplina del estudiante:

Art. 12º Constituyen faltas muy graves:

A, inciso i: Intentar o realizar un plagio parcial o total, al rendir una evaluación, una práctica o durante la elaboración o presentación de un trabajo o asignación académica, usando cualquier medio, modalidad, objeto o equipos para tal fin, en forma directa o a través de terceros.

A, inciso iii: Presentar un mismo trabajo, en todo o parte, en más de una asignatura sin el conocimiento y consentimiento expreso de los docentes involucrados.

A, inciso vii: *La suplantación de identidad (suplantar y/o ser suplantado) al momento de rendir una evaluación o actividad académica, de cualquier tipo, en beneficio propio o ajeno.*

Art. 21º Las faltas leves son sancionadas, preferentemente, con amonestación escrita; salvo decisión distinta de la Autoridad competente.

Art. 22º Las faltas graves son sancionadas, preferentemente, con separación temporal; salvo decisión distinta de la Autoridad competente.

Art. 20º Las faltas muy graves referidas a las faltas contra la probidad académica, listadas en el literal a) del Art. 12 del presente, serán sancionadas con la desaprobación con calificación de cero en el curso. En caso de reincidencia, la sanción a aplicar será la separación definitiva. Las faltas muy graves en general, serán sancionadas, preferentemente, con la separación definitiva; salvo decisión distinta de la Autoridad competente.

5. Cumplir con los artículos del Reglamento de Estudios:

Art. 29º Revisión de calificación, plazos y procedimientos

- a) Una vez cargada la nota en el sistema, el estudiante puede solicitar la recalificación del examen o rectificaciones de notas al docente del curso, debiendo presentar y sustentar el caso con la evidencia que corresponda en un plazo máximo de 48 horas de cargada la nota en el sistema, para evaluaciones parciales, continuas y evaluaciones finales; y de 24 horas de cargada la nota, para el examen sustitutorio. Una vez cerrado el periodo académico, ya no hay lugar a reclamos o rectificaciones de notas.
 - b) La solicitud de recalificación solo es admitida en los siguientes casos: • Error en registro de nota u olvido por parte del docente. • Si hay error de suma en el puntaje del examen. • Hay evidencia de que el docente dejó de leer accidentalmente parte de la respuesta del examen.
 - c) La revisión del examen puede implicar una corrección con una nota menor, igual o mayor a la nota original.
 - d) El docente tiene la obligación de resolver la recalificación o rectificación de nota de forma escrita, atendiendo o no la solicitud del estudiante, en un plazo máximo de 48 horas.
 - e) Si el estudiante no está de acuerdo con la recalificación o rectificación puede elevar el caso al responsable de carrera, quien responde en un plazo máximo de 48 horas, y cuyo fallo es inimpugnable. No aplica para exámenes sustitutorios.
 - f) Por ningún motivo proceden las solicitudes de rectificación que estén fuera de los plazos establecidos por la Universidad.
6. Está prohibido el intercambio de materiales de clase como copias, lapiceros, cuadernos, etc.
 7. Escriba con lapicero. Se rechazan las solicitudes de revisión de notas si el examen es respondido con lápiz,
 8. Apague y guarde su teléfono celular.
 9. No use hojas adicionales.

Firma del estudiante

DNI: _____

Observaciones o incidencias registradas por parte del docente:

HE RECIBIDO, LEÍDO Y REVISADO EL PUNTAJE OBTENIDO Y:

<input type="checkbox"/>	ES CONFORME
<input type="checkbox"/>	NO ES CONFORME

Firma del estudiante
DNI: _____

EVALUACIÓN T3
ANÁLISIS DE ALGORITMOS Y ESTRATEGIAS DE PROGRAMACIÓN
INGENIERÍA DE SISTEMAS COMPUTACIONALES
SEMESTRE 2025-2

CÓDIGO DEL ESTUDIANTE: n00400647

NÚMERO DE CLASE: _____

APELLIDOS Y NOMBRES DEL ESTUDIANTE : Soldevilla Rios Piero Alexander FECHA: 14/11/2025

DOCENTE: Jorge Bojórquez

Duración del examen: 60 min

1. Problema 1 (20 pts)

Programar una aplicación en Python la cual debe subir a un repositorio de github, para salir de un laberinto el cual es recorrido por una persona que tiene 18 unidades de energía y cada celda que camina desde el inicio al final va a gastar lo que dice en la celda. Si hay un 1 gasta una unidad de energía, si hay un -2 recupera dos unidades de energía:

- La casilla de salida está la celda verde ni gasta ni repone energía
- La casilla de llegada está en la celda roja ni gasta ni repone energía.
- Se puede avanzar un cuadro en el siguiente orden: izquierda, abajo, arriba y derecha
- Se puede avanzar por todas las celdas (celdas blancas) excepto por las de valor 99 (celdas plomas).
- Aquellas celdas con valores positivos gastan tanta energía como el valor que indican. Las que tienen un valor negativo reponen tanta energía como el valor que indican.
- Debe buscar un camino que le permita llegar a la celda roja desde la celda verde tal que no gasta más de las 18 unidades de energía que tiene la persona.
- Use el algoritmo de backtracking para que partiendo de un laberinto-matriz de 9 x 9 pueda mostrar que camino debería tomar para salir. Ver figura más abajo
- Muestre el laberinto original
- Si logra la salida o si no lo logra debe mostrar un mensaje que avise del resultado, así como la matriz que indica como salir
- El laberinto-matriz a usar será este:

1	1	1	1	99	1	1	1	1	1
1	99	99	1	99	1	99	1	99	
1	1	99	1	1	1	99	1	99	
99	1	99	1	99	99	99	1	99	
1	1	99	-1	1	1	1	3	99	
-2	99	99	1	99	99	99	1	1	
1	99	1	-1	1	1	1	1	99	
1	99	99	99	99	2	99	1	99	
F		1	3	1	1	1	99	1	1

TRAS TERMINAR EL EXAMEN debe adjuntar como respuesta estos archivos SIN COMPRIMIR:

- Link del repositorio problemas 1 resuelto en github.
- En un archivo Word llamado “**resultado.docx**” Tome capturas de pantalla de la ejecución de los problemas 1.

RÚBRICA

PREG.	CRITERIOS DE EVALUACIÓN	INDICADORES DE RENDIMIENTO				
1	Github (4 ptos)	Llega a solución correcta. Demuestra dominio del tema	La respuesta es casi correcta. Demuestra buen conocimiento del tema	La respuesta es medianamente correcta. Demuestra conocimiento medio del tema	La respuesta es incorrecta en 80%. demuestra poco conocimiento del tema	No llega a solución correcta, No demuestra conocimiento del tema
		4 puntos	3 puntos	2 puntos	1 puntos	0 puntos
	Crea matriz de laberinto y de imprimir laberinto (4 ptos)	Llega a solución correcta. Demuestra dominio del tema	La respuesta es casi correcta. Demuestra buen conocimiento del tema	La respuesta es medianamente correcta. Demuestra conocimiento medio del tema	La respuesta es incorrecta en 80%. demuestra poco conocimiento del tema	No llega a solución correcta, No demuestra conocimiento del tema
		4 puntos	3 puntos	2 puntos	1 puntos	0 puntos
	Hace funciones para backtracking (4 ptos)	Llega a solución correcta. Demuestra dominio del tema	La respuesta es casi correcta. Demuestra buen conocimiento del tema	La respuesta es medianamente correcta. Demuestra conocimiento medio del tema	La respuesta es incorrecta en 80%. demuestra poco conocimiento del tema	No llega a solución correcta, No demuestra conocimiento del tema
		4 puntos	3 puntos	2 puntos	1 puntos	0 puntos
	Mostrar Ejecución (4 ptos)	Llega a solución correcta. Demuestra dominio del tema	La respuesta es casi correcta. Demuestra buen conocimiento del tema	La respuesta es medianamente correcta. Demuestra conocimiento medio del tema	La respuesta es incorrecta en 80%. demuestra poco conocimiento del tema	No llega a solución correcta, No demuestra conocimiento del tema
		4 puntos	3 puntos	2 puntos	1 puntos	0 puntos
	Cursos Cisco: Python 1 y Python 2 (4 ptos)	Llega a solución correcta. Demuestra dominio del tema	La respuesta es casi correcta. Demuestra buen conocimiento del tema	La respuesta es medianamente correcta. Demuestra conocimiento medio del tema	La respuesta es incorrecta en 80%. demuestra poco conocimiento del tema	No llega a solución correcta, No demuestra conocimiento del tema
		4 puntos	3 puntos	2 puntos	1 puntos	0 puntos

Ejecución el Examen:

```
# -----
# LABERINTO CON BACKTRACKING - 9x9
# -----
```

```
laberinto = [
    [1, 1, 1, 1, 99, 1, 1, 1, "I"],
    [1, 99, 99, 1, 99, 1, 99, 1, 99],
    [1, 1, 99, 1, 1, 1, 99, 1, 99],
    [99, 1, 99, 1, 99, 99, 99, 1, 99],
    [1, 1, 99, -1, 1, 1, 1, 3, 1],
```

```

[-2, 99, 99, 1, 99, 99, 99, 1, 1],
[1, 99, 1, 1, -1, 1, 1, 1, 99],
[1, 99, 99, 99, 99, 2, 1, 99, 99],
["F", 1, 3, 1, 1, 1, 99, 1, 1]
]
```

ENERGIA_INICIAL = 18

movimientos = [(0, -1), (1, 0), (-1, 0), (0, 1)]

camino = [[0] * 9 for _ in range(9)]

Buscar la posición inicial "I"

inicio_x = inicio_y = 0

for i in range(9):

 for j in range(9):

 if laberinto[i][j] == "I":

 inicio_x, inicio_y = i, j

FUNCION BACKTRACKING

def resolver(x, y, energia):

 if laberinto[x][y] == "F":

 camino[x][y] = 1

 return True

 camino[x][y] = 1

 if laberinto[x][y] not in ("I", "F"):

 energia += laberinto[x][y]

 if energia < 0 or energia > ENERGIA_INICIAL:

 camino[x][y] = 0

 return False

 for dx, dy in movimientos:

 nx, ny = x + dx, y + dy

 if 0 <= nx < 9 and 0 <= ny < 9:

 if laberinto[nx][ny] != 99 and camino[nx][ny] == 0:

 if resolver(nx, ny, energia):

 return True

 camino[x][y] = 0

 return False

EJECUCION

print("\n--- LABERINTO ORIGINAL ---")

```

for fila in laberinto:
    print(fila)

exito = resolver(inicio_x, inicio_y, ENERGIA_INICIAL)

print("\n-----")
if exito:
    print("SALIDA ENCONTRADA - Ruta marcada con 1:")
else:
    print("NO EXISTE UNA RUTA VALIDA CON")

```

EVIDENCIAS:

```

# -----
#   LABERINTO CON BACKTRACKING - 9x9
# -----


✓ laberinto = [
    [1, 1, 1, 1, 99, 1, 1, 1, "I"],
    [1, 99, 99, 1, 99, 1, 99, 1, 99],
    [1, 1, 99, 1, 1, 1, 99, 1, 99],
    [99, 1, 99, 1, 99, 99, 99, 1, 99],
    [1, 1, 99, -1, 1, 1, 1, 3, 1],
    [-2, 99, 99, 1, 99, 99, 99, 1, 1],
    [1, 99, 1, 1, -1, 1, 1, 1, 99],
    [1, 99, 99, 99, 99, 2, 1, 99, 99],
    ["F", 1, 3, 1, 1, 1, 99, 1, 1]
]

ENERGIA_INICIAL = 18

movimientos = [(0, -1), (1, 0), (-1, 0), (0, 1)]

camino = [[0] * 9 for _ in range(9)]


# Buscar la posición inicial "I"
inicio_x = inicio_y = 0
✓ for i in range(9):
    ✓ for j in range(9):
        ✓ if laberinto[i][j] == "I":
            inicio_x, inicio_y = i, j

```

```

# -----
#   FUNCION BACKTRACKING
# -----
def resolver(x, y, energia):
    if laberinto[x][y] == "F":
        camino[x][y] = 1
        return True

    camino[x][y] = 1

    if laberinto[x][y] not in ("I", "F"):
        energia += laberinto[x][y]

    if energia < 0 or energia > ENERGIA_INICIAL:
        camino[x][y] = 0
        return False

    for dx, dy in movimientos:
        nx, ny = x + dx, y + dy
        if 0 <= nx < 9 and 0 <= ny < 9:
            if laberinto[nx][ny] != 99 and camino[nx][ny] == 0:
                if resolver(nx, ny, energia):
                    return True

    camino[x][y] = 0
    return False

# -----
#   EJECUCION
# -----
print("\n--- LABERINTO ORIGINAL ---")
for fila in laberinto:
    print(fila)

exito = resolver(inicio_x, inicio_y, ENERGIA_INICIAL)

print("\n-----")
if exito:
    print("SALIDA ENCONTRADA - Ruta marcada con 1:")
else:
    print("NO EXISTE UNA RUTA VALIDA CON")

```

```
--- LABERINTO ORIGINAL ---
```

```
[1, 1, 1, 1, 99, 1, 1, 1, 'I']
[1, 99, 99, 1, 99, 1, 99, 1, 99]
[1, 1, 99, 1, 1, 1, 99, 1, 99]
[99, 1, 99, 1, 99, 99, 99, 1, 99]
[1, 1, 99, -1, 1, 1, 1, 3, 1]
[-2, 99, 99, 1, 99, 99, 99, 1, 1]
[1, 99, 1, 1, -1, 1, 1, 1, 99]
[1, 99, 99, 99, 99, 2, 1, 99, 99]
['F', 1, 3, 1, 1, 1, 99, 1, 1]
```

```
-----  
NO EXISTE UNA RUTA VALIDA CON
```

```
Press any key to continue . . .
```

```
--- LABERINTO ORIGINAL ---
```

```
[1, 1, 1, 1, 99, 1, 1, 1, 'I']
[1, 99, 99, 1, 99, 1, 99, 1, 99]
[1, 1, 99, 1, 1, 1, 99, 1, 99]
[99, 1, 99, 1, 99, 99, 99, 1, 99]
[1, 1, 99, -1, 1, 1, 1, 3, 1]
[-2, 99, 99, 1, 99, 99, 99, 1, 1]
[1, 99, 1, 1, -1, 1, 1, 1, 99]
[1, 99, 99, 99, 99, 2, 1, 99, 99]
['F', 1, 3, 1, 1, 1, 99, 1, 1]
```

```
-----  
NO EXISTE UNA RUTA VALIDA CON
```

```
El programa 'python.exe' terminó con código 0 (0x0).
```