

APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Algoritmos y Estructuras de Datos	FECHA 29/01/20	DNI	GIN 02	

Hoja 1/9

Duración: tres horas (60 minutos -> test)

Antes de comenzar:

- **No dé la vuelta al examen hasta que se le indique.**
- **Antes de comenzar, no olvide rellenar su nombre, apellidos y DNI en todas las hojas** y, en su caso, en todos los folios que se le entreguen.
- **Escribir únicamente con bolígrafo de tinta negra o azul**, excepto en los casos donde se desee crear diagramas de flujo o pseudocódigo.
- **Todos los dispositivos electrónicos deben estar guardados en la mochila y apagados o sin sonido.** En caso contrario debe indicarse al profesor los motivos por los que debe tenerse el teléfono con sonido antes de comenzar el examen.
- **Se permite el uso de calculadoras, aunque su utilización no es necesaria para la realización del examen.**
- **El uso de cualquier dispositivo electrónico o el plagio o copia conllevará el suspenso del examen y de la evaluación ordinaria, acudiendo directamente a la evaluación extraordinaria según el artículo 9 del reglamento sobre pruebas de evaluación y su revisión.**
- Artículo 9.º. De la utilización de métodos ilícitos para la superación de las pruebas de evaluación. "Cualquier evidencia de plagio, de copia del examen de un compañero, o cualquier intento de obtener de forma fraudulenta las respuestas a las preguntas de una prueba de evaluación, será penalizada y podrá suponer la apertura de expediente y la aplicación de las correspondientes sanciones, pudiendo llegar a sustanciarse en la expulsión de la Universidad del alumno o alumnos implicados, conforme a lo establecido en los arts. 29 y siguientes del Reglamento del Alumnado de esta Universidad. "
- **Rellenar los datos en todas las páginas del examen.**
- **No se podrá abandonar la sala hasta que no haya finalizado el examen completo** (incluyendo esta parte test y la parte teórico-práctica).

Resultados de aprendizaje que se evalúan en este examen:

- Entender los conocimientos básicos de algorítmica y complejidad computacional.
- Ser capaz de realizar análisis de complejidad de algoritmos.
- Ser capaz de programar en un lenguaje de programación utilizando estructuras de datos comunes.
- Ser capaz de implementar algoritmos básicos.
- Ser capaz de implementar algoritmos utilizando técnicas de desarrollo de algoritmos.

APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Algoritmos y Estructuras de Datos	FECHA 29/01/20	DNI	GIN 02	

Hoja 2/9

Ejercicio 1 (0.5 puntos) Tiempo estimado: 10 minutos. Responda a las cuestiones

La calificación máxima de esta parte tipo test es de 0.5 puntos. Tenga en cuenta que:

- Para cada pregunta, sólo existe una respuesta correcta.
- Cada respuesta correcta sumará 0.10 puntos.
- Cada respuesta incorrecta restará 0.10 puntos.
- Las preguntas no contestadas ni suman ni restan puntuación.

- En relación al estado de un objeto, para preservar el principio de encapsulación:
 - Los atributos de nuestra clase deben permanecer públicos, para permitir un acceso total a la información que almacenamos en los objetos. Sin embargo, aquellos métodos que realicen operaciones internas (y no deban ser utilizados), deben establecerse como privados.
 - Debemos establecer la visibilidad más restrictiva (por ejemplo, privada) en los atributos de una clase. Así, cualquier software que utilice nuestro objeto, sólo accederá al estado de los objetos mediante los métodos que le hayamos permitido utilizar.
- Los métodos de una clase no pueden devolver objetos:
 - Verdadero
 - Falso
- ¿Cuál de las siguientes características de la programación orientada a objetos está relacionada con la reutilización de código?
 - Abstracción
 - Herencia
- El tiempo de acceso a un dato en una lista es:
 - $O(n)$
 - $O(1)$
- Un algoritmo de ordenación que implemente el método de inserción, tiene una complejidad en el peor caso de:
 - $O(n^2)$
 - $O(n^3)$



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Algoritmos y Estructuras de Datos	FECHA 29/01/20	DNI	GIN 02	

Hoja 3/9

Ejercicio 2 (2 puntos) Tiempo estimado: 25 minutos. Responda a las cuestiones

Respuestas correctas suman 0.5 puntos. Se debe responder a 4 preguntas de las 5 disponibles. Las 4 preguntas a contestar son elegidas libremente por el estudiante.

1) ¿Cuáles son las limitaciones del análisis O para medir la complejidad computacional?



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA	Algoritmos y Estructuras de Datos	FECHA	29/01/20	DNI
		GIN 02		

Hoja 4/9

- 2) Explica las diferencias entre una tabla Hash y un árbol. ¿Cuándo conviene utilizar cada una de estas estructuras? ¿Cuál es el orden de complejidad de una operación de (1) inserción, de (2) acceso aleatorio, de (3) búsqueda de un dato, y de (4) crecimiento de la estructura?



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA	Algoritmos y Estructuras de Datos	FECHA	29/01/20	DNI
			GIN 02	

Hoja 5/9

- 3) Explica las diferencias entre una pila y una cola. ¿Cuándo conviene utilizar cada una de estas estructuras? ¿Cuál es el orden de complejidad de una operación de (1) inserción, de (2) acceso aleatorio, y de (3) búsqueda de un dato?



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA	Algoritmos y Estructuras de Datos	FECHA	29/01/20	DNI
			GIN 02	

Hoja 6/9

4) Explica el método de ordenación denominado inserción y describe las iteraciones sobre el siguiente conjunto de datos:

50 20 84 13 22 16 89 85



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA	Algoritmos y Estructuras de Datos	FECHA	29/01/20	DNI
GIN 02				

Hoja 7/9

5) Dadas dos funciones y su número de operaciones:

$$A = 1024n$$

$$B = 16n^3$$

Calcular a partir de qué tamaño de entrada A es más eficiente que B.

APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA Algoritmos y Estructuras de Datos	FECHA 29/01/20	DNI	GIN 02	

Hoja 8/9

Ejercicio 3 (0.75 + 0.75 puntos) Tiempo estimado: 20 minutos. Dado el siguiente algoritmo recursivo:

main.py



saved

```

1  def recursive(a,b):
2      if(b<0):
3          return 1/recursive(a,-b)
4      if a == 0 or b == 0:
5          return 1
6      else:
7          return a * recursive(a,b-1)
8
9  def main():
10     print(recursive(2,4))
11
12  if __name__ == "__main__":
13     main()

```

Preguntas:

- ¿Qué imprime el código?
- Explica brevemente qué cálculo está haciendo y qué tipo de recursividad está empleando.

Respuestas:



APELLIDOS	NOMBRE		GRUPO	CALIFICACIÓN
ASIGNATURA	Algoritmos y Estructuras de Datos	FECHA 29/01/20	DNI	GIN 02

Hoja 9/9