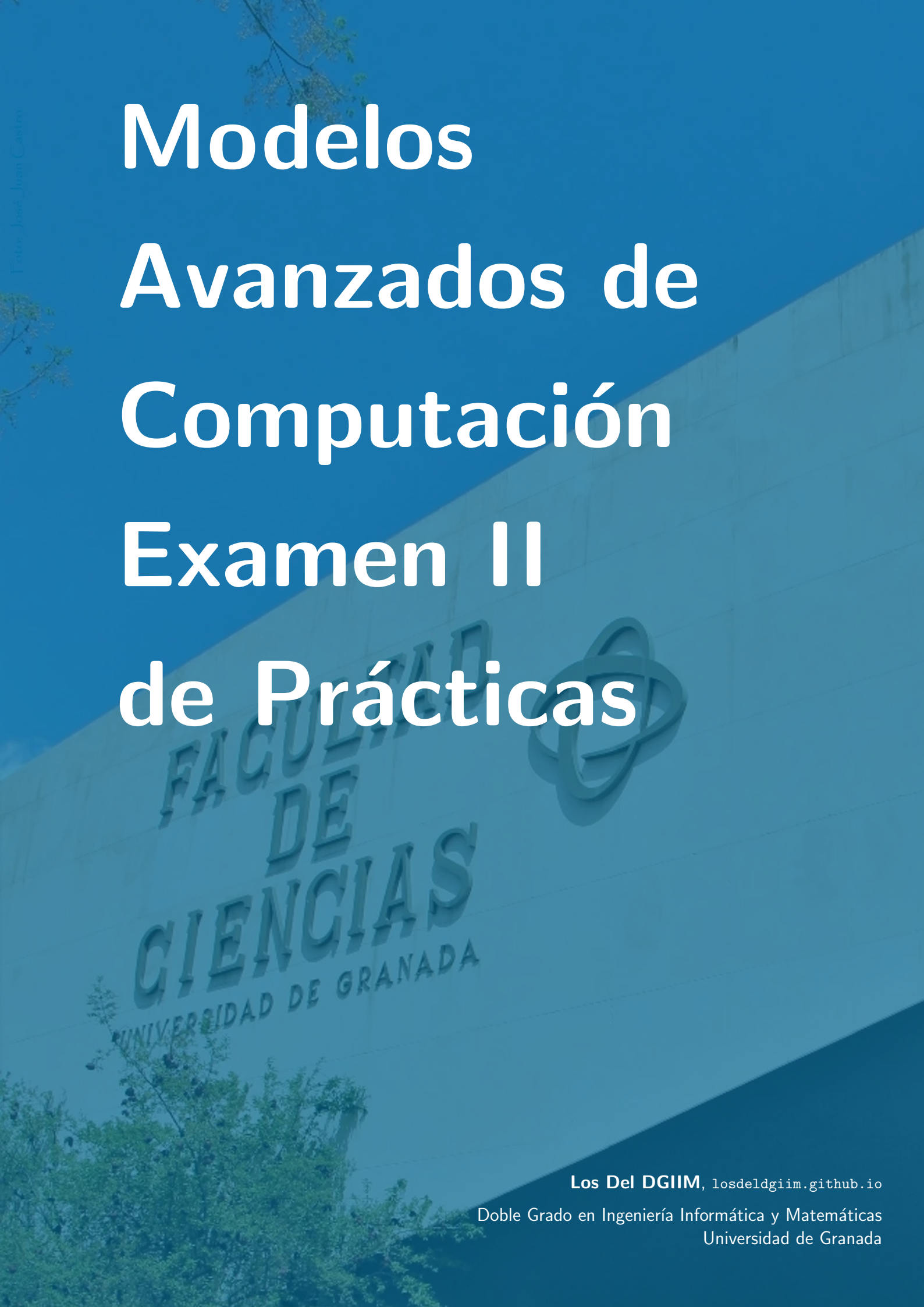
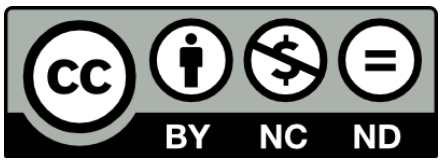
Modelos Avanzados de Computación Examen II de Prácticas



Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Modelos Avanzados de Computación Examen II de Prácticas

Los Del DGIIM, `losdeldgiim.github.io`

José Juan Urrutia Milán

Granada, 2026

Asignatura Modelos Avanzados de Computación.

Curso Académico 2025/26.

Grado Grado en Ingeniería Informática.

Grupo Grupo de Prácticas A1.

Profesor Serafín Moral Callejón.

Descripción Tercer Examen de Prácticas.

Fecha 27 de Mayo.

Duración 1 hora.

Ejercicio 1. Entre los dos siguientes problemas uno es espacio-logarítmico y el otro es tiempo-polinómico. Clasificar los problemas y dar un algoritmo para demostrar sus clases:

1. Dado un conjunto de números, determinar la mediana del conjunto.

   En caso de que la cantidad de números no sea impar, rechazar.

2. Dado un conjunto de números, determinar si existe una asignación en parejas de modo que la suma de todas las parejas sea la misma.

Solución.

Ejercicio 1. Entre los dos siguientes problemas uno es espacio-logarítmico y el otro es tiempo-polinómico. Clasificar los problemas y dar un algoritmo para demostrar sus clases:

1. Dado un conjunto de números, determinar la mediana del conjunto.

   En caso de que la cantidad de números no sea impar, rechazar.

2. Dado un conjunto de números, determinar si existe una asignación en parejas de modo que la suma de todas las parejas sea la misma.

Podemos intuir que el primer problema es más fácil que el segundo, por lo que trataremos de crear un algoritmo espacio-logarítmico para el primero y un algoritmo tiempo-polinómico para el segundo. A pesar de ser el segundo problema de una clase más difícil, resulta más fácil la tarea de crear dicho algoritmo.

Problema 2. Debemos crear un algoritmo tiempo-polinómico para esta tarea. El algoritmo es el siguiente:

Algoritmo 1

Entrada: Una secuencia de números $n_1, n_2, \ldots, n_m$ separados por &.

Contar la cantidad m de números en una segunda cinta.
Si m es impar **entonces**
  Terminar y decir "NO".
Fin Si
Ordenar los números de menor a mayor.
Almacenar en una tercera cinta la suma del primero y del último, k.
Poner "2" en una cuarta cinta, nos referimos a este contador por i.
Mientras que $i \leqslant {}^{m}/_{2}$ **entonces**
  $t =$ Sumar el elemento i más el $m - i + 1-$ésimo.
  Si $t \neq k$ **entonces**
    Terminar y decir "NO".
  Fin Si
  Incrementar i en una unidad.
Fin Mientras
Terminar y decir "SÍ".

Vemos a continuación:

- El Algoritmo 1 es tiempo-polinómico. Supuesto que el número de caracteres de entrada $0, 1, \&$ es n:
  1. Contar la cantidad m de números es $O(n)$.
  2. Ordenar los números de menor a mayor puede hacerse en[1] $O(n^2)$.
  3. El procedimiento del "mientras" recorre $m - 2$ casillas, por lo que es $O(n)$, y su cuerpo es constante.

[1] Sabemos que puede hacerse en $O(n \log n)$, pero esta suposición facilita el razonamiento.

En definitiva, el algoritmos es $O(n^2)$, por lo que es tiempo-polinómico.

- El Algoritmo 1 es correcto (es decir, resuelve el problema), puesto que si existe una tal asignación, el número máximo M debe pertenecer a alguna pareja, por lo que debe ser emparejado con otro número n_k. Si este número no es el más pequeño de todos ($m < n_k$), al emparejar el más pequeño con otro número n_t (tenemos $n_t \leqslant M$) tendremos que:

$$n_t + m < n_k + M$$

Por lo que este emparejamiento no sería válido. Así, si un emparejamiento de este tipo existe, el número máximo debe ser emparejado con el mínimo. Si ahora consideramos el mismo problema pero sin considerar los números máximo y mínimo, por el mismo razonamiento llegamos a que el segundo más grande debe ser emparejado con el segundo más pequeño.

De esta forma, se puede demostrar por inducción que, en caso de existir una tal asignación, debe ser la que hemos hecho.

Problema 1. Buscamos un algoritmo espacio-logarítmico:

Algoritmo 2

Entrada: Una secuencia de números en binario $n_1, n_2, \ldots, n_m$ separados por &.

Contar la cantidad de números m en una segunda cinta con un contador binario.
Si m es par **entonces**.
  Terminar y decir "NO".
Fin Si
Copiar el primer número n_1 en una tercera cinta, separado por & de su índice.
Inicializar la cuarta cinta con un contador binario c inicializado a 0.
Repetir m veces:
  Usar c para contar la cantidad de números menores o iguales que el de la tercera cinta.
  Usar una quinta cinta para guardar el número n_k que consigue un menor valor de c siendo $c > \frac{m-1}{2}$.
  Si $c = \frac{m-1}{2}$ **entonces**
    Terminar y devolver el número de la tercera cinta.
  Fin Si
  Si en la tercera cinta tenemos "$n_i\&i$", cambiarlo por $n_{i+1}\&i+1$.
Fin Repetir
Terminar y devolver el número de la quinta cinta.

Vemos:

- El algoritmo es espacio-logarítmico. Para verlo, veamos qué espacio usamos, suponiendo que la secuencia de entrada contiene n caracteres:
  - La secuencia de números de entrada nunca se modifica, y la secuencia de salida la escribimos de izquierda a derecha sin volver hacia atrás, por lo que estas secuencias no las contamos.

- En la segunda cinta se cuenta el número m de números en binario. Contar en binario hasta n ocupa un espacio $O(\log n)$, por lo que como $m < n$ tendremos que este contador es $O(\log n)$.
- En la tercera guardamos números de la secuencia de entrada separados por su índice en binario, por lo que también es $O(\log n)$.
- En la cuarta cinta guardamos un contador binario que siempre será menor o igual que m, por lo que es $O(\log n)$.

En definitiva, el algoritmo es $O(\log n)$ en espacio.

- El Algoritmo 2 es correcto, puesto que si m es impar siempre tendremos que $\exists k \in \{1, \ldots, m\}$ con n_k siendo la mediana, por lo que:
  - Bien $c = \frac{m-1}{2}$ para n_k, de donde n_k es la mediana.
  - Bien c para n_k es estrictamente mayor que $\frac{m-1}{2}$ pero es el mínimo valor de c para cualquier otro valor $n_t \neq n_k$. Esta situación ocurre cuando el valor de la mediana se repite, como por ejemplo en:

    $$3\ 1\ 4\ 4\ 4\ 5\ 7$$

    Tenemos que $m = 7 \implies \frac{m-1}{2} = 3$. Vemos que $c = 2$ para n_1, que $c = 1$ para n_2, $c = 5$ para $n_3 = n_4 = n_5$, $c = 6$ para n_6 y $c = 7$ para n_7.
    En este caso la mediana se alcanza en 4, cuyo valor c es mayor que $\frac{m-1}{2}$ pero es el mínimo con esta propiedad.