Asignatura: Análisis de Algoritmos
Código: 17827
Centro: Escuela Politécnica Superior
Titulación: Grado en Ingeniería Informática
Nivel: Grado
Tipo: Formación obligatoria
Nº de créditos: 6

GUÍA DOCENTE DE ANÁLISIS DE ALGORITMOS

La presente guía docente corresponde a la asignatura Análisis de Algoritmos, aprobada para el curso lectivo 2016-2017 en Junta de Centro y publicada en su versión definitiva en la página web de la Escuela Politécnica Superior. Esta guía docente aprobada y publicada antes del periodo de matrícula tiene el carácter de contrato con el estudiante.

ASIGNATURA

ANÁLISIS DE ALGORITMOS (AA)

1.1. Código

17827 del Grado en Ingeniería Informática

1.2. Materia

Análisis de Algoritmos

1.3. Tipo

Formación obligatoria

1.4. Nivel

Grado

1.5. Curso

2º

1.6. Semestre

1º

1.7. Número de créditos

6 créditos ECTS

1.8. Requisitos previos

Para un buen aprovechamiento del curso, es recomendable haber aprobado las materias Programación I y Programación II. De no ser así, el curso puede seguirse pero requiriendo tal vez un esfuerzo extra, sobre todo en su parte práctica, lo que puede incidir en el rendimiento del estudiante en otras asignaturas en las que esté matriculado.

También se supone unos conocimientos matemáticos básicos al nivel de los cursos Algebra y Cálculo I.

En relación a las clases prácticas, se recomienda que el estudiante sea capaz de expresar en pseudocódigo algoritmos básicos que incluyan bucles anidados así como las estructuras de control if, while y for. A su vez, sus conocimientos del lenguaje C deben incluir

- Conocer la estructura general de un programa en C.
- Comprender los tipos de datos primitivos de C y utilizar los tipos estructurados del lenguaje (tablas y estructuras)
- Utilizar correctamente las estructuras de control
- Escribir prototipos y cuerpos de funciones y conocer los mecanismos de llamada por valor y por dirección.
- Tener una cierta familiaridad con los algoritmos recursivos.
- Conocer y aplicar las estructuras de control y selección de flujo en C.
- Conocer y hacer uso de punteros y de la gestión dinámica de memoria.
- Estar familiarizado con el uso de algún entorno gráficos de programación, como Visual C++, Eclipse, NetBeans o similares.

Asimismo, es recomendable que el estudiante disponga de un dominio del idioma inglés que le permita leer la bibliografía de consulta.

1.9. Requisitos mínimos de asistencia a las sesiones presenciales

Se considerarán dos itinerarios, Evaluación Continua y No Evaluación Continua, tanto para teoría como para prácticas.

El itinerario de Evaluación Continua en Teoría requerirá una asistencia continuada a las clases teóricas **así como presentarse a los dos parciales que se convoquen (salvo por razones de fuerza mayor); el no cumplir este último requisito supondrá el abandono del itinerario**.

El itinerario de Evaluación Continua en Prácticas requerirá una asistencia mínima del 85% de clases prácticas así como la entrega puntual de las prácticas asignadas.

En cualquier caso, **la asistencia a clases de teoría o problemas se considera esencial para la superación de la asignatura**, ya que dicha asistencia supone

- La toma de contacto explicada con el material de la asignatura, con la que el estudiante obtiene una primera y en general suficiente comprensión de dicho material. Dicha comprensión es **muy difícil y costosa de obtener por otras vías**.
- Una parte muy importante, **casi un 30%, del tiempo que el estudiante debe dedicar a la asignatura** (unas 45 horas sobre 150).

Por todo ello es sumamente recomendable asistir a dichas clases, indicando la experiencia que **no hacerlo convierte en extremadamente difícil la superación de la asignatura**.

Las recomendaciones previas son también aplicables a la asistencia clases prácticas.

1.10. Datos del equipo docente

Nota: se debe añadir @uam.es a todas las direcciones de correo electrónico.

Dr. José R. Dorronsoro (Coordinador)
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-358 Edificio B – 3ª Planta
Teléfono: +34 91 497 2329
Correo electrónico: jose.dorronsoro@uam.es
Página web: http://www.ii.uam.es/~jdorronsoro
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Pablo Varona Martínez
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-330 Edificio B – 3ª Planta
Teléfono: +34 91 497 2263
Correo electrónico: pablo.varona@uam.es
Página web: http://www.ii.uam.es/~pvarona
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

Dr. Carlos Aguirre Maeso
Departamento de Ingeniería Informática
Escuela Politécnica Superior
Despacho - Módulo: B-322 Edificio B – 3ª Planta
Teléfono: +34 91 497 2280
Correo electrónico: carlos.aguirre@uam.es
Página web: http://www.ii.uam.es/~aguirre
Horario de atención al alumnado: Petición de cita previa por correo electrónico.

1.11. Objetivos del curso

Las **competencias comunes a la rama de Informática del módulo de Programación y Estructura de Datos** que el estudiante adquiere con la asignatura Análisis de Algoritmos son:

- B4. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- C6. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- C7. Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.

El estudiante también adquiere con la asignatura Análisis de Algoritmos la siguiente **competencia específica:**

- CC3. Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Las competencias generales y transversales del Grado de Ingeniería Informática que se adquieren en esta asignatura son:

- Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión correspondiente al grado en Ingeniería Informática.

Dentro de los **resultados del aprendizaje,** el estudiante adquiere en la asignatura Análisis de Algoritmos con carácter general los siguientes:

- Comprensión de los principales algoritmos así como de las técnicas necesarias para la estimación de su complejidad.
- Comprensión de las técnicas de programación utilizando lenguajes de alto nivel.

De manera más detallada, los mismos pueden desarrollarse como sigue:

- Eficacia de algoritmos: Concepto, elementos de medida, casos de medida. Análisis de bucles y selecciones. Análisis general de algoritmos recursivos.
- Herramientas matemáticas básicas del análisis de eficacia de algoritmos:

Comparación asintótica del crecimiento de funciones. Aproximación de sumas por integrales. Estimación de soluciones de desigualdades recurrentes.

- Conocimiento detallado de algoritmos básicos de ordenación (Inserción, Quicksort, Mergesort, Heapsort): Pseudocódigos, evolución detallada sobre ejemplos, evolución sobre árboles de decisión. Análisis de eficacia en casos peor, medio y mejor. Cotas inferiores de rendimiento para familias de algoritmos.
- Conocimiento detallado del funcionamiento de algoritmos de búsqueda sobre claves: Pseudocódigos, evolución detallada sobre ejemplos. Tipo de datos diccionario: implementación sobre árboles binarios de búsqueda y árboles AVL, rendimiento de primitivas.
- Construcción de funciones hash y resolución de colisiones por encadenamiento y direccionamiento abierto.
- Alcance de un nivel medio-alto de programación en C: trabajo con punteros y memoria dinámica, determinación y control de errores, organización de código, archivos de cabecera.

Los **objetivos generales** son que el estudiante al acabar el curso logre:

- Tener un buen conocimiento general de los principales algoritmos de ordenación y búsqueda
- Ser capaz de determinar el efecto que las estructuras de un algoritmo tienen en su rendimiento.
- Identificar para un algoritmo dado los elementos determinantes de su rendimiento en el caso peor.
- Ser capaz de escribir y resolver las ecuaciones que recogen el rendimiento de algoritmos simples, incluyendo ecuaciones recursivas simples.
- Determinar para un problema dado el coste y ventajas de posibles enfoques algorítmicos para el mismo.
- Comunicar de manera clara, estructurada y concisa los resultados de su trabajo.
- Participar activamente en los análisis y discusiones de grupo que se establezcan al hilo del desarrollo de las actividades propuestas, cooperar con otros compañeros en el desarrollo de trabajos conjuntos y comunicar con propiedad y corrección sus reflexiones sobre los resultados de su trabajo.

Objetivos específicos

Al terminar cada unidad el estudiante debe ser capaz de:

Unidad 1. Introducción al análisis de eficacia de algoritmos:

- Enumerar las medidas básicas de eficacia.
- Efectuar un primer análisis de la eficacia de algoritmos simples.
- Manejar las notaciones o, O y similares de crecimiento asintótico de funciones.
- Estimar el crecimiento de funciones expresadas como sumas.
- Definir y aplicar el rendimiento en los casos peor, mejor y medio.

Unidad 2. Algoritmos básicos de ordenación

- Evaluar el rendimiento del algoritmo de inserción y su eficacia.
- Definir algoritmo local y enunciar las cotas inferiores para sus casos peor y medio.
- Estimar el rendimiento de un algoritmo local para algunas tablas dadas.

Unidad 3. Algoritmos avanzados de ordenación

- Evaluar el rendimiento de un algoritmo de tipo Divide y Vencerás y obtener la estructura general de las desigualdades recurrentes a partir de un pseudocódigo dado.
- Enunciar el pseudocódigo y rendimiento en los casos peor y medio del algoritmo mergesort.
- Reproducir el funcionamiento del algoritmo mergesort sobre una tabla dada.
- Enunciar el pseudocódigo y rendimiento en los casos peor y medio del algoritmo quicksort.
- Reproducir el funcionamiento del algoritmo quicksort sobre una tabla dada.
- Enunciar la defición de heap y max-heap.
- Representar un heap en forma de árbol binario y en forma de tabla.
- Enunciar el pseudocódigo y rendimiento en el caso peor del algoritmo heapsort.
- Reproducir el funcionamiento del algoritmo heapsort sobre una tabla dada.

- Reproducir los argumentos empleados en el cálculo de esos casos.

Unidad 4. Árboles de decisión para algoritmos de ordenación.

- Definir árbol de decisión.
- Construir árboles de decisión sobre algoritmos concretos y tablas de tamaño 3 y 4.
- Estimar las cotas inferiores para algoritmos de ordenación por comparación de claves en los casos peor y medio.
- Reproducir los argumentos empleados en la obtención de dichas cotas.

Unidad 5. Algoritmos y EdDs básicos de búsqueda

- Enunciar el concepto del TAD Diccionario
- Usar árboles binarios como EdD para el TAD Diccionario.
- Calcular los valores de los casos peor y medio para la búsqueda en árboles binarios.
- Enunciar la definición de AVL
- Construir un AVL a partir de una tabla determinada.
- Estimar el caso peor de un AVL.
- Reproducir los argumentos empleados en la obtención de dichas propiedades.

Unidad 6. Tablas hash

- Enunciar la definición de función hash y los principales métodos de la resolución de colisiones.
- Construir una tabla hash a partir de una tabla determinada.
- Construir tablas hash mediante encadenamiento.
- Definir tablas hash mediante direccionamiento abierto y enunciar los principales métodos de sondeo.
- Calcular el caso medio con éxito de una tabla hash con direccionamiento abierto a partir del caso medio sin éxito

1.12. Contenidos del programa

El programa detallado del curso es:

1. Análisis de eficacia de algoritmos:
    a. Tipos de análisis. Medidas de eficacia.
    b. Herramientas matemáticas: notaciones o, O, etc.
    c. Crecimiento de sumas e integrales.
    d. Casos peor, mejor y medio.
2. Algoritmos básicos de ordenación
    a. Algoritmos de selección y burbuja.
    b. Algoritmo de inserción: pseudocódigo, eficacia.
    c. Algoritmos locales: definición, cotas inferiores para sus casos peor y medio.
3. Algoritmos avanzados de ordenación.
    a. Métodos Divide y Vencerás.
    b. Desigualdades recurrentes. Estimación del crecimiento.
    c. Algoritmo mergesort: pseudocódigo, caso peor.
    d. Algoritmo quicksort: pseudocódigo, caso peor, caso medio.
    e. Algoritmo heapsort: pseudocódigo, caso peor.
4. Árboles de decisión para algoritmos de ordenación.
    a. Árboles de decisión: concepto y construcción
    b. Cotas inferiores para algoritmos de ordenación por comparación de claves.
5. Algoritmos básicos de búsqueda
    a. Búsqueda lineal y binaria.
    b. TAD Diccionario. Árboles binarios como EdD para diccionarios.
    c. Caso peor y medio para la búsqueda en árboles binarios.

d. Árboles AVL: construcción y profundidad máxima.

6. Tablas hash

   a. Construcción de funciones hash y resolución de colisiones.

   b. Tablas hash sobre encadenamiento.

   c. Tablas hash con direccionamiento abierto: sondeos lineales, cuadráticos y aleatorios.

1.13. Referencias de consulta

No hay un manual que se ajuste en su totalidad a los contenidos del curso. En cualquier caso, las referencias inferiores constituyen un buen complemento para el seguimiento del curso.

Referencias básicas:

- Weiss, Data structures and algorithm analysis in C, Benjamin Cummings.

Referencias complementarias:

- Cormen, Leiserson, Rivest, Introduction to algorithms, The MIT Press--Mc Graw Hill.
- Baase, Computer algorithms, Addison-Wesley.

Referencia para programación

- Kernighan, Ritchie, The C programming language, Prentice hall.

2. Métodos docentes

La asignatura se organiza en clases teóricas, clases de problemas, prácticas de laboratorio, actividades programadas en grupo y pruebas individualizadas.

En las **clases teóricas** se presentarán los conceptos de manera clara y suficientemente concisa utilizando para ello preferentemente el método de lección magistral pero requiriendo la participación del estudiante y fomentando su iniciativa. El estudiante deberá trabajar de manera autónoma el contenido de cada clase para adquirir una comprensión suficiente que le permita seguir las clases sucesivas y abordar los problemas propuestos.

En las **clases de problemas** se presentarán y resolverán ejercicios y problemas tipo, representativos de lo exigible en las pruebas evaluatorias. El estudiante debe complementar este trabajo realizado en clase con la resolución por su cuenta de ejercicios complementarios que se propondrán a lo largo del curso.

En la medida de lo posible en cuanto a medios y calendario, las clases de problemas se complementarán con **sesiones de trabajo en grupos reducidos**, donde se propondrá la resolución de ciertos problemas significativos, dándose unas primeras pautas sobre los mismos y siguiendo el profesor el avance efectuado por los distintos grupos.

Esta actividad dependerá de la disponibilidad de espacios adecuados para este tipo de trabajo.

En las **prácticas de laboratorio** se efectuarán **tres** prácticas sobre algunos de los algoritmos abordados en clase. Se realizarán utilizando el lenguaje de programación C e incorporarán el uso de otras herramientas, como funciones de medida de tiempo de ejecución o visualizadores de curvas de datos. Se fomentará el aprendizaje cooperativo y el trabajo sostenido en el tiempo, inculcándose además el sentido ético que debe primar en el estudio universitario y la vida profesional, persiguiéndose comportamientos fraudulentos como la copia de prácticas.

Material electrónico de trabajo: los documentos electrónicos de trabajo (apuntes, transparencias, ejercicios/problemas del curso, ejemplos de exámenes, enlaces recomendados, vídeos, etc.) se publican en la sección de AA en plataforma Moodle (https://moodle.uam.es).

3. Tiempo de trabajo del estudiante

Recomendaciones generales

- Una aproximación razonable al esfuerzo medio a esperar del estudiante que supere la asignatura se detalla en el **cronograma** de la misma.
- La **asistencia activa** a clase se considera **imprescindible** para la mayoría de los estudiantes. Del cronograma se desprende que dicha asistencia representa unas 38 horas, aproximadamente la mitad del esfuerzo que el estudiante debe hacer para superar los exámenes. Esto es, de no asistir a clase, el estudiante deberá como mínimo efectuar dicho esfuerzo por su cuenta, y ello sin tener en cuenta la dificultad de entender, sin ninguna explicación, materiales de cierta complejidad y dificultad.
- Las clases se impartirán según la fórmula de **lección magistral participativa**, con soporte de trasparencias y explicaciones en pizarra.
- Se recomienda que el estudiante acuda a las mismas provisto de copias impresas de dichas trasparencias así como de los problemas asignados.

En principio la distribución **aproximada** de las diversas actividades del curso será la siguiente:

Resumen		**Horas**	**Porcentaje**	**Tipo**
	Clases teóricas	22	15	Teo./Práct
	Clases de problemas	15	10	Teo./Práct
	Clases prácticas	24	16	Laboratorio
	Tutorías	4	3	Tutoría
	Pruebas escritas	6	4	
No presencial	Estudio semanal material teórico	20	13	Estudio
	Resolución semanal de ejercicios y problemas	11	7	Laboratorio
	Trabajo práctico complementario semanal	12	8	Laboratorio
	Estudio de material teórico para el examen final	14	9	Estudio
	Ejercicios y problemas para el examen final	22	15	Estudio
	Totales	150	100	

4. Métodos de evaluación y porcentaje en la calificación final

4.1. Consideraciones generales

La asignatura consta de una parte práctica y una teórica. Ambas partes se puntúan sobre 10 puntos. La nota final de la asignatura se obtiene a partir de las notas obtenidas en las partes de teoría y de prácticas como sigue

Calificación=0.4*Prácticas + 0.6*Teoría

Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos, tanto en la parte de teoría como en la práctica. En caso contrario, la nota final que figurará en actas será

Calificación= 0.4*Mín(5, Prácticas) + 0.6*Mín(5,Teoría)

Las notas de teoría y de prácticas se guardan para la convocatoria extraordinaria del mismo curso académico.

4.2. Calificación de teoría

Evaluación en el itinerario de evaluación continua:
La parte de teoría del curso se evaluará en el itinerario de Evaluación Continua mediante tres exámenes parciales, correspondientes cada uno de ellos aproximadamente a una tercera parte del curso. Tendrán una duración de 1 hora con dos preguntas sobre 10 puntos de varios apartados. Dichos exámenes tendrán lugar de acuerdo al siguiente calendario aproximado

- Parcial 1: en la semana del 17 de octubre de 2016.
- Parcial 2: en la semana del 28 de noviembre de 2016.
- Parcial 3: primeros de enero de 2017, coincidiendo con la fecha del examen final.

Los parciales 1 y 2 serán liberatorios para aquellos estudiantes que los superen obteniendo además una nota de 4 o superior en cada pregunta.
Caso de no superar alguno de estos dos parciales, el estudiante se examinará de nuevo del mismo (o, en su caso, de ambos) en el examen final, además de, naturalmente, la tercera parte del curso.
La nota final de teoría se calculará como la media de las tres notas de cada parte del curso. Aquellos estudiantes que hayan liberado los dos parciales deberán tener una nota de 3,5 o superior en el examen de la tercera parte para superar la asignatura.

Evaluación fuera del itinerario de evaluación continua:
Para aquellos estudiantes que no hayan seguido el itinerario de Evaluación Continua, la nota final será exclusivamente la del examen final de 180'. En este caso, dicho

examen **tendrá una primera parte en la que será necesario obtener una nota mínima de 7 para superarlo.**

Observación importante:
La nota de teoría sólo se guardará para la convocatoria extraordinaria en el caso de superarse la materia en su totalidad. **No se guardarán notas de parciales individuales.**

4.3. Calificación de prácticas

En la **convocatoria ordinaria**, habrá una **prueba final de prácticas** con dos modalidades, según el estudiante haya seguido o no el itinerario de evaluación continua.

Evaluación en el itinerario de evaluación continua:
En este itinerario la prueba será de una hora y supondrá efectuar una modificación en el laboratorio de alguna de las tres prácticas realizadas y la respuesta a algunas cuestiones sobre la misma.

Evaluación fuera del itinerario de evaluación continua:
En este caso el estudiante deberá haber completado y entregado previamente a la fecha del examen las prácticas propuestas y la prueba, de dos horas, supondrá efectuar en el laboratorio una modificación de cierto alcance de alguna de las prácticas propuestas así como responder a algunas cuestiones.

En ambos casos la calificación de la parte práctica de la asignatura será:

40% Nota Examen + 20% Nota Pract1 + 20% Nota Pract2 + 20% Nota Pract3

Para poder aplicar la fórmula anterior el estudiante deberá, en ambos itinerarios, obtener al menos una calificación de 4 puntos en cada una de las prácticas, un promedio de al menos 5 puntos entre las tres entregas y una calificación de al menos 5 puntos en el examen. Si el estudiante no cumple alguna de estas condiciones la calificación de prácticas se calculará mediante la siguiente fórmula:

0.4*min(5, Nota examen) + 0.2*min(5, Nota Prac 1)
+ 0.2*min(5, Nota Prac 2)
+ 0.2*min(5, Nota Prac 3)

En caso de retraso en la entrega de las prácticas, se podrá imponer una penalización en la nota de la práctica, que en ningún caso será superior a un punto por semana de retraso.
La nota de prácticas podrá guardarse de un curso al siguiente **una única vez** y siempre que se haya superado una determinada nota. Dicha nota se indicará a los estudiantes en la primera clase teórica del curso siguiente para que los interesados procedan a solicitar la aplicación de la nota del año anterior.

En la **convocatoria extraordinaria** el estudiante es evaluado fuera del itinerario de evaluación continua, conservando, si procede, la calificación de las prácticas con una calificación superior a 4. El estudiante deberá entregar, si no lo ha hecho ya en la convocatoria ordinaria y previamente a la realización del examen de la convocatoria extraordinaria la totalidad de las prácticas y realizar el examen de prácticas correspondiente a la convocatoria extraordinaria.

ATENCIÓN:
Cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría como en cualquiera de los apartados de las prácticas, penalizará por igual, tanto a los alumnos que copian como a los copiados. La penalización por copia implica la aplicación de la normativa interna La penalización por copia implica la aplicación de las normativas de la UAM y interna de la EPS http://www.ii.uam.es/esp/alumnos/informacion/normativa.php

5. Cronograma

El esfuerzo total medio del estudiante se estima en unas 150 horas (incluyendo pruebas), repartidas entre el inicio del curso y la celebración del examen final.
A continuación se da una aproximación al cronograma de la asignatura, que podría experimentar pequeños cambios según se desarrolle el curso.

Semana	Clase	Contenido
05-sep	1	Tema I: Tiempo abstracto de ejecución (TAE)
	2	Ejemplos de cálculo, análisis de rendimiento
	3	OB, Ejemplos de cálculo
12-sep	4	Comp. Asint. de funciones
	5	Est. Crecimiento I
	6	Est. Creciemiento II. Complejidad de algoritmos
19-sep	7	Complejidad de algoritmos
	8	Complejidad de la búsqueda lineal
	9	Problemas (1)
26-sep	10	Problemas (2)
	11	Problemas en grupo I
	12	Tema II: Mét. básicos de ordenación. Insersort
03-oct	13	Métodos locales de ordenación. Cotas inferiores caso peor
	14	Cotas inferiores caso medio.
	15	Problemas (3)
10-oct	16	Problemas (4)
	17	Problemas (5)
	18	Métodos DyV, MergeSort
17-oct	19	**Parcial I**
	20	Análisis de MergeSort. Introd a QuickSort
	21	QuickSort, caso medio
24-oct	22	Problemas (6)
	23	HeapSort
	24	Árboles de decisión. Ejemplos
31-oct	25	Cotas inf para el caso peor de ord. por cdc
	26	Cotas inf para el caso de ord. por cdc
	27	Problemas (7)
07-nov	28	Tema III: Revisión de búsq. lineal, búsq. binaria
	29	TAD Diccionario: primitivas
	30	TAD Diccionario: implementación
14-nov	31	Rend. de primitivas de diccionario sobre ABdB

	32	Árboles AVL: construcción: rotaciones, ejemplos
	33	Prof. Máx. de AVLs, rend. sobre AVLs
21-nov	34	Problemas (8)
	35	Problemas (9)
	36	Problemas (10)
28-nov	37	**Parcial II**
	38	Hashing: introducción
	39	Hashing: funciones hash
05-dic	40	Hashing por encadenamiento
	41	Hashing por direccionamiento abierto
	42	Caso medio de hashing por DA uniforme
12-dic	43	Caso medio de hashing por DA lineal y cuadrático
	44	Problemas (11)
	45	Problemas (12)
19-dic	46	Problemas (13)
	47	Problemas (14)
	48	Problemas (15)