

## Programa de Syllabus

**Unidad Académica Responsable:** DIICC

**CARRERA** a las que se imparte: Ingeniería Civil Informática

### I. - IDENTIFICACION

<b>Nombre:</b> Taller de programación Competitiva 1-6		
<b>Código:</b> 503601 - 503606	<b>Créditos:</b> 1	<b>Créditos SCT:</b> 2
<b>Prerrequisitos:</b> 503601		
<b>Modalidad:</b> Presencial	<b>Calidad:</b> Electivo	<b>Duración:</b> Semestral
<b>Semestre en el plan de estudios:</b>	Ing. Informática - 5to -> 11vo semestre	
<b>Trabajo Académico:</b> 5		
<b>Horas Teóricas:</b> 0 <b>Horas Prácticas:</b> 2 <b>Horas de otras actividades:</b> 4		

Docente Responsable	Waldo Gálvez
Docente Colaborador	Benjamín Letelier
Comisión Evaluación	
Duración (semanas)	
Fecha:	Aprobado por:

### II. - DESCRIPCION

Este taller surge de la necesidad de capacitar a los estudiantes en la comprensión y resolución de problemas no triviales bajo restricciones de tiempo empleando técnicas de programación. Esto les permitirá mejorar sus habilidades de abstracción y programación dirigido tanto a su participación en competencias de programación internacionales como al ejercicio profesional en una industria especializada. La modalidad del taller consiste en el trabajo en laboratorio, donde los estudiantes analizan problemas para luego diseñar e implementar algoritmos que los solucionen. Los Talleres de Programación Competitiva forman parte de una secuencia de talleres semestrales en los cuales los estudiantes podrán perfeccionar sus habilidades de resolución de problemas, variando en cada taller la complejidad de los problemas a los cuales se enfrentan los alumnos (la cual estará en concordancia a su nivel de aprendizaje). Estos talleres aportan a las siguientes competencias del perfil de egreso:

- Aplicar principios de matemáticas, ciencias de la ingeniería y ciencias de la computación, a problemas de ingeniería informática.
- Trabajar en equipos para solucionar problemas de manera colaborativa e incentivando el trabajo multidisciplinario.

### III.- RESULTADOS DE APRENDIZAJE ESPERADOS

Al finalizar la asignatura, los alumnos habrán logrado:

1. Ejercitar la capacidad de entender problemas y proponer soluciones algorítmicas.

2. Ejercitar y profundizar en el aprendizaje de estrategias de diseño de algoritmos
3. Coordinar el trabajo en equipo bajo restricciones de tiempo en la resolución de problemas.
4. Ejercitar sus habilidades de codificación en diversos lenguajes de programación, distinguiendo el más adecuado para cada tipo de problema.

#### **IV.- CONTENIDOS**

1. Juez virtual e introducción al mundo de la programación competitiva
  - a. Órdenes de complejidad
  - b. ACC, WAs, TLEs
  - c. Input/Output
2. Resolución de problemas ad-hoc
3. Contenedores más importantes de la librería estándar de C++ y sus usos
  - a. Vector
  - b. Set
  - c. Map
  - d. Queue
  - e. Stack
  - f. Priority Queues
4. Aplicaciones de ordenamiento y búsqueda
  - a. Búsqueda Binaria
5. Problemas de string
  - a. Prefijos
  - b. Sufijos
  - c. Pattern Matching
  - d. Suffix Arrays + LCP Array
  - e. Tries
6. Uso de estructuras de datos no incluidas en la STL
  - a. Union-Find
  - b. Segment Tree
  - c. Binary Indexed Tree
  - d. Sqrt Decomposition
7. Problemas de grafos y árboles
  - a. Recorridos
  - b. Recorridos con costo mínimo
  - c. MST
  - d. LCA
  - e. Binary Lifting
  - f. Rerooting
  - g. Max Flow, Min Cut
  - h. Algoritmo Húngaro
8. Revisión de estrategias de diseño
  - a. Memoización
  - b. Programación dinámica
9. Problemas matemáticos
  - a. Números primos
  - b. GCD
  - c. LCM
  - d. Matemática modular
  - e. Problemas de combinatoria

Dada la naturaleza del curso, es posible que no todos los contenidos propuestos sean cubiertos, y estos podrían cambiar en base al criterio de los estudiantes y del docente.

## V.- METODOLOGIA

Los contenidos del curso se revisarán por medio de clases teóricas y listados de ejercicios de programación competitiva. La modalidad será principalmente una clase teórica con un listado asíncrono, proporcionando a los estudiantes ejercicios y material didáctico extra para que trabajen de manera individual. Cada estudiante seleccionará el nivel de taller del que quiere ser parte (Básica [B], Intermedio [I] y Avanzado [A]). Cada taller verá los mismos contenidos, pero tendrá que resolver una cantidad mínima de ejercicios (diferente para cada taller) indicada en cada listado (este mínimo variará en cada listado).

## VI.- EVALUACION

En los listados evaluados, cada ejercicio será evaluado de acuerdo con la siguiente rúbrica:

- **0 puntos:** Ejercicio no subido, no aceptado por juez virtual o tomado de otra parte sin comentar.
- **1 punto:** Ejercicio aceptado con código propio o completamente comentado

Los listados se realizarán de manera individual y cada listado tendrá una duración de entre [1..4] semanas. La calificación final de cada listado será calculada mediante la siguiente fórmula (teniendo en cuenta que  $PTot\_Li$  varía entre cada nivel). Si después de un listado, alguien termina con  $PObt\_Li > PTot\_Li$ , entonces la diferencia se puede añadir a cualquier listado que se desee (resolver menos ejercicios en otro listado).

$$N_{Li} = \frac{PObt_{Li} * 6}{PTot_{Li}} + 1$$

La nota final del curso consistirá en el promedio de todos los listados, calculados con la fórmula anterior. En caso de un promedio menor a 4 se realizará un listado extra de menor duración que reemplazará la nota del peor listado.

## VII.- BIBLIOGRAFIA Y MATERIAL DE APOYO

- Steven S. Skiena: The Algorithm Design Manual, Springer-Verlag, 2008, Second Edition. ISBN-10 038794860, ISBN-13 97-0387948607.
- Steven Halim and Felix Halim: Competitive Programming 3 – The new lower bound for programming contests, Handbook for ACM ICPC and IOI Contestants, 2013, third edition
- Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein: Introduction to Algorithms. MIT Press y McGraw-Hill Book Company, 2009, 3rd edition.

## VIII.- PLANIFICACIÓN

Semana	Fecha	Actividad
1	14/03/25	Coordinación y horario
2	19/03/25	Introducción a la Programación Competitiva en C++ y Jueces Online
3	26/03/25	Estructuras de Datos de la STL
4	02/04/25	Matemáticas I
5	09/04/25	Búsqueda Binaria I
6	16/04/25	Búsqueda Binaria II
7	23/04/25	Grafos I
8	30/04/25	Grafos II
9	07/05/25	Programación Dinámica I
10	14/05/25	Programación Dinámica II
11	21/05/25	<b>FERIADO</b>
12	28/05/25	Strings
13	04/06/25	Range Queries I
14	11/06/25	Range Queries II
15	18/06/25	Matemáticas II
16	25/06/25	Matemáticas III
17	02/07/25	<b>VACACIONES DE INVIERNO</b>
18	09/07/25	Grafos III

## IX. – OTROS

### Contactos y horas de consultas:

- i) Waldo Gálvez: wgalvez@inf.udec.cl (viernes, entre 14:00 y 15:30, oficina 310 de la FI).
- ii) ~~Vicente~~ Benjamín Letelier: vletelier@udec.cl (jueves, entre 14:00 y 15:30, a través de Ms. Teams y Sala Postgrado DCC).

### Horario y sala del curso:

Miércoles de 13:45-15:00, sala IS 3-1 (3er piso edificio de Ing. Sistemas)

- <https://vjudge.net/>
- <https://uva.onlinejudge.org/>
- <http://uhunt.felix-halim.net/> sobre los problemas de la UVA, permite seleccionar por dificultad y crear competencias
- <http://www.spoj.com/> //tags por tema
- <http://acm.timus.ru/> //bueno para comenzar (permite ordenar por dificultad)
- <http://codeforces.com/> //tags por tema + casos en los que se falla
- <https://www.codechef.com/>
- <https://icpcarchive.ecs.baylor.edu/> //competencias ACM ICPC pasadas
- <https://piazza.com/class/hsm4gwdljxr2r2> (Curso en Piazza)
- <https://github.com/dseco/acmudec>
- <https://piazza.com/class/hsm4gwdljxr2r2?cid=52>
- <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html> (Referencia general de C)
- <http://www.cplusplus.com/reference/> (Referencia general de C++)
- <https://cp-algorithms.com> (Guía de muchos algoritmos útiles para programación competitiva)
- <https://usaco.guide> (Guía de muchos algoritmos útiles para programación competitiva)