

---

# **FUNDAMENTOS DE COMPUTABILIDAD**

## **GUÍA DOCENTE**

---

## OBJETIVOS:

---

El área de Fundamentos de Computabilidad se encuadra en el campo de la Informática Teórica, que trata de dar respuesta a cuestiones tales como qué clases de problemas pueden ser resueltos algorítmicamente (Teoría de la Computabilidad) y en el caso de los que pueden serlo, cuál es la dificultad inherente a su resolución (Teoría de la Complejidad). De estas dos subáreas trabajaremos únicamente la primera.

Los **objetivos generales** y más conceptuales son los relacionados con la comprensión de la disciplina:

- Conocer la motivación y los contenidos básicos de la Teoría de la Computabilidad.
- Profundizar en el hábito de formalización en el campo informático.
- Manejar de manera precisa los términos propios y usuales del área de Informática Teórica
- Entender la formalización del concepto de problema computable.
- Utilizar correctamente el modelo de programación en casos concretos
- Ser consciente de la existencia de límites absolutos más allá de los cuales los métodos algorítmicos no son aplicables.
- Formarse criterios para identificar intuitivamente problemas no computables.

## COMPETENCIAS:

---

Las **competencias** relacionados con el "saber hacer:

- Manejar el lenguaje "rudimentario" que servirá como herramienta para desarrollar los conceptos de la teoría de la computabilidad.
- Convencerse de que la capacidad de este lenguaje no es menor que la de los lenguajes de programación convencionales.
- Relacionar el comportamiento de los programas con las funciones
- Utilizar el modelo elegido (programas while) para demostrar la computabilidad de problemas concretos.
- Distinguir entre los problemas computables los decidibles y los parcialmente decidibles.
- Dominar las técnicas elementales para determinar la no computabilidad de los problemas: diagonalización y reducción (con uso y sin uso del Teorema s-m-n).

## **TEMARIO:**

---

### **PARTE I**

Definición del formalismo de programación que nos permitirá formalizar los conceptos de computabilidad e incomputabilidad.

- Preliminares
- Los programas-while
- Macroprogramas

### **PARTE II**

Presentación de las técnicas básicas que nos permiten distinguir problemas incomputables.

- Función universal
- La no-computabilidad
- Decidible, semidecidible o indecidible

### **PARTE III**

Profundización en el concepto de computabilidad con otra técnica: Reducibilidad

## DESCRIPCIÓN:

---

La Teoría de la Computabilidad es una disciplina antigua, hasta el punto de preceder la propia Informática al establecer las principales limitaciones de los computadores antes incluso de que estos fueran construidos. Esta antigüedad, siendo como es cosa de admirar, tiene también sus desventajas. Una de las últimas es que los modelos y las notaciones que en principio se usaron para probar resultados de incomputabilidad siguen siendo utilizados prácticamente sin alterar desde entonces. El modelo más clásico de computador abstracto sigue siendo la Máquina de Turing, y el proceso de computación es visto en todos los textos clásicos como una manipulación de números.

Para el o la estudiante universitaria de Ingeniería Informática esto suele suponer un problema. Por un lado la Máquina de Turing resulta un dispositivo ajeno en el que la programación resulta poco menos que un suplicio. Por si fuera poco, esta lejanía entre el modelo teórico y los computadores reales (con los que está empezando a familiarizarse) provocan un escepticismo con respecto a los resultados: "Aunque hayamos probado que una Máquina de Turing no es capaz de resolver este problema, ¿cómo podemos tener la seguridad de que ningún otro computador puede hacerlo? ¿Qué tiene que ver una cosa con la otra?".

Los **Programas-while** permiten resolver los mismos problemas que las máquinas de Turing, pero en cambio son mucho más **sencillos** de utilizar, sobre todo para personas que tienen una experiencia previa en la informática real, pues toman la forma de lenguaje imperativo clásico. Este curso además utiliza los Programas-while aprovechando sus ventajas y reformulándolos de manera que la computación quede definida en términos de manipulación de símbolos arbitrarios, algo que está mucho más en concordancia con la realidad informática.

Los contenidos podemos organizarlos en tres bloques. Se ha elegido una parte sencilla y motivadora (Preliminares), para revisar algunos conocimientos previos y adaptarlos a las nuevas necesidades. Empezaremos estableciendo los **formalismos** necesarios para el desarrollo de otros temas partiendo de los conceptos sobre funciones. El siguiente paso es explicar en detalle qué son los programas while y cómo se utilizan. Dado que constituyen un lenguaje de programación muy simple, justificaremos por qué no hemos incorporado otras instrucciones o tipos de datos, viendo que ello no era necesario porque dichas instrucciones o tipos de datos son simulables con los pocos mimbres que incorpora el lenguaje. De esa forma concluye el primer bloque.

Los otros dos bloques presentan los resultados fundamentales de Computabilidad y dos de las técnicas de esta área: la de **diagonalización** y la **reducción**, que son básicas en la disciplina y necesitan de un gran desarrollo práctico.

## **ACTIVIDADES:**

---

Las actividades tendrán por objetivo contribuir al logro de los objetivos ya recogidos. Por ello proponemos actividades suficientes para todos los objetivos y que estas se adapten al tipo de objetivo (de comprensión, de aplicación práctica de habilidades, etc.).

Es de utilidad la separación de los problemas en dos tipos: los introductorios, más sencillos que permiten centrarse en el tema que se explica y los avanzados que contemplan los problemas con un grado de dificultad mayor. En cada hoja de problemas se especifican los conocimientos necesarios para abordar su resolución, y que se habrán obtenido con la lectura del material dispuesto para cada tema. Además se detallan objetivos específicos por grupos de problemas.

Para cada hoja de ejercicios se han seleccionado unos cuantos para los que se ofrece una solución detallada. Puede utilizarse el resumen asociado a cada hoja de ejercicios como material de apoyo, ya que incluye un resumen de los conceptos y técnicas a manejar en cada caso.

<b>Preliminares</b>	
Contenidos	Objetivos específicos
Conjuntos	<p>Manejar con soltura las operaciones básicas de conjuntos.</p> <p>Seleccionar conjuntos usuales de trabajo.</p>
Palabras y lenguajes	Manejar con soltura las operaciones básicas de palabras y lenguajes
<p>Funciones parciales.</p> <p>Propiedades.</p>	<p>Utilizar el fenómeno de la parcialidad al trabajar con funciones</p> <p>Establecer claramente el dominio y rango de cualquier función</p> <p>Distinguir las nociones de convergencia y divergencia</p> <p>Diferenciar funciones totales y no totales</p> <p>Describir con lenguaje preciso propiedades de conjuntos y funciones</p> <p>Adaptar las nociones de inyectividad, suprayectividad, biyectividad, composición e inversa al caso de funciones no totales.</p>
Material de estudio	<p>Preliminares</p> <p>R. Preliminares</p>
Actividades	I.1. Preliminares

<b>Los programas-while</b>	
Contenidos	Objetivos específicos
Descripción de los programas-while.	<p>Construir programas utilizando el lenguaje de los programas-while.</p> <p>Relacionar los conceptos de función y programa a través de las nociones de cómputo asociado a un programa y de función calculada por un programa.</p> <p>Demostrar la while-computabilidad de algunas funciones y la while-decibilidad de algunos predicados utilizando los programas-while.</p>
Las macros como abreviaturas de programas.  Funciones de código.	<p>Distinguir claramente entre macroprogramas y programas-while.</p> <p>Obtener programas-while equivalentes a partir de macroprogramas.</p> <p>Definir macros y establecer su expansión</p> <p>Demostrar la while-computabilidad de algunas funciones y la while-decibilidad de algunos predicados utilizando los macroprogramas.</p> <p>Diferenciar entre dificultad de resolución y posibilidad de resolución.</p>
Computabilidad sobre tipos de datos arbitrarios.	Utilizar la implementación de algunos tipos de datos.
Los programas-while como datos.	<p>Usar el tipo de datos de los programas en la construcción de funciones y predicados.</p> <p>Utilizar el concepto de índice para referenciar las funciones while-computables, sus dominios y sus rangos.</p>
Material de estudio	<p>Los programas-while</p> <p>Otros recursos a modo de resumen: R. Los Programas-while; R. Macroprograma; R. Implementaciones; R. El tipo de datos programas-while</p>
Actividades	<p>I.2. Los Programas-while</p> <p>I.3. Macroprograma</p> <p>I.4. Implementaciones</p> <p>II.5. El tipo de datos programas-while</p>

<b>Técnicas básicas de Computabilidad</b>	
Contenidos	Objetivos específicos
<p>Función Universal</p> <p>Los predicados T y E</p> <p>Intercalado de procesos</p>	<p>Manejar con precisión las definiciones de <math>\Phi</math>, T y E</p> <p>Diseñar programas que simulan la ejecución de otros programas</p> <p>Aplicar la técnica de intercalado en demostraciones de computabilidad</p>
<p>Computabilidad no efectiva</p> <p>Computabilidad versus incomputabilidad</p> <p>Tesis de Church-Turing</p>	<p>Analizar el concepto de computabilidad</p> <p>Conocer problemas computables sin algoritmo concreto</p> <p>Reflexionar sobre el alcance de la noción de computabilidad</p>
<p>La no computabilidad</p> <p>El problema de parada.</p>	<p>Establecer los pasos de las demostraciones de incomputabilidad.</p> <p>Dar ejemplos de problemas no computables.</p> <p>Entender las limitaciones de los métodos algorítmicos.</p>
<p>Técnica de diagonalización.</p> <p>Variaciones de la técnica de diagonalización</p>	<p>Manipular la clase de todas las funciones computables.</p> <p>Construir funciones de diagonalización.</p> <p>Armar contradicciones para refutar supuestos de computabilidad.</p> <p>Utilizar diversas versiones de la técnica de diagonalización para demostrar la incomputabilidad de algunas funciones.</p> <p>Comparar problemas para intuir su clasificación como computable o no.</p>
<p>Decibilidad y Semidecidibilidad</p>	<p>Clasificar problemas en decidibles, casi decidibles pero no decidibles y no semidecidibles</p>
<p>Material de estudio</p>	<p>Técnicas Básicas de Computabilidad</p> <p>Ejemplos de Diagonalización</p> <p>La función universal</p> <p>Otros recursos a modo de resumen: R. La técnica de intercalado; R. Computabilidad no efectiva; R. Diagonalización; R. Ctos. decidibles y semidecidibles</p>
<p>Actividades</p>	<p>II.6. La función universal</p> <p>II.7. La técnica de intercalado</p> <p>II.8. Computabilidad no efectiva</p> <p>II.9. Diagonalización</p> <p>II.10. Ctos. decidibles y semidecidibles</p>



<b>Otra técnica de Computabilidad: reducción</b>	
Contenidos	Objetivos específicos
Limitaciones de la técnica de diagonalización	Razonar sobre la imposibilidad de construir una función diagonal en algunos casos
Reducción en computabilidad	<p>Establecer los pasos de las demostraciones de incomputabilidad reduciendo un problema cuya incomputabilidad queremos demostrar a otro cuya incomputabilidad conocemos.</p> <p>Conocer otros ejemplos de problemas no computables.</p>
<p>Jeraquía de conjuntos</p> <p>Parametrizaciones</p> <p>Teorema s-m-n</p> <p>Aplicaciones</p>	<p>Jerarquizar problemas de acuerdo a su dificultad Construir funciones de reducción.</p> <p>Armar contradicciones para refutar supuestos de computabilidad.</p> <p>Utilizar la técnica de reducción para demostrar la incomputabilidad de algunas funciones.</p> <p>Comparar problemas para intuir su clasificación como computable o no.</p>
Material de estudio	<p>El método de reducción en Teoría de la Computabilidad</p> <p>Otros recursos a modo de resumen: R. Reducción</p>
Actividades	III.11. Reducción