



**Departamento: de SISTEMAS**

**Carrera: Ingeniería en Sistemas de Información**

**Área: Programación – Tecnologías Básicas**

**Asignatura: SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES**

**Nivel: 2**

**Tipo: ANUAL**

**Titular: ESP. ING. JORGE BUABUD**

**Adjuntos: ING. FERNANDO RUBÉN ARAUJO**

ING. JOSÉ UBALDO BONAPARTE

**Auxiliares Graduados: LIC. SERGIO POLITI, ING. MARTA RONVEAUX,  
ING. MARISEL BEDRAN, ING. LILIANA DEL PRADO**

## **PLANIFICACIÓN DE LA ASIGNATURA**

### **FUNDAMENTACIÓN DE LA MATERIA DENTRO DEL PLAN DE ESTUDIOS.**

Este espacio curricular aborda los temas centrales de la Teoría de la Computación o Informática Teórica, que constituyen los pilares fundamentales donde se asientan las Ciencias de la Computación. La temática principal trata sobre la Teoría de Autómatas y Lenguajes Formales, donde se conjugan la Lingüística y el Procesamiento Electrónico de Datos. En cualquier carrera relacionada con la computación o informática, no puede faltar el tratamiento de estos temas que dan origen a conceptos esenciales dentro del área, como lo son la Lógica Algorítmica y los Sistemas de Programación, en relación al software; y la Máquina de Turing y el Modelo de Von Neumann, dentro del hardware.

Esta asignatura es la base para encarar la solución de diversos problemas informáticos: diseño de compiladores; intérpretes de comandos; traductores de lenguajes naturales; criptografía; análisis de patrones; abstracciones de datos; validación de datos y seguridad informática; especificaciones de argumentos en comandos de sistemas operativos o de búsqueda de información en bases de datos; redes neuronales; verificación de software; diseño y verificación de circuitos digitales; modelización de: sistemas de control automático, sistemas robotizados, domótica, etc.; sistemas de Lindenmayer: fractales, análisis de secuencias biológicas, composiciones musicales, estructuras arquitectónicas, etc.

**PROPÓSITOS U OBJETIVOS DE LA MATERIA.**

- ✓ Representar lenguajes formales en sus niveles léxico, sintáctico y semántico; utilizando los conocimientos mínimos de lingüística matemática y los paradigmas declarativos.
- ✓ Comprender la complejidad algorítmica mediante el análisis de los lenguajes recursivos, lenguajes recursivamente enumerables y las Máquinas de Turing.
- ✓ Comprender el proceso de traducción de lenguajes de programación y en particular el proceso de compilación, utilizando herramientas de metacompilación.
- ✓ Diseñar aceptores de lenguajes formales en sus niveles léxico (Autómatas Finitos) y sintáctico (Autómatas de Pila), utilizando las gramáticas formales (regulares y libres de contexto).
- ✓ Implementar los algoritmos propios de la teoría de autómatas, afianzando y consolidando los conocimientos previos adquiridos en el área de programación.
- ✓ Resolver problemas de aplicación de la informática teórica, en prácticas concretas en el laboratorio de la cátedra.

**CONTENIDOS.**

- UNIDAD 1: Introducción a la Teoría de Lenguajes Formales y Autómatas Finitos.
  - Fundamento: El conocimiento de los conceptos básicos de la teoría de la computación, permitirá la comprensión del nivel léxico de los lenguajes formales y en particular los de programación, posibilitando la solución de diversos problemas inherentes a la profesión, cuyo modelo matemático es el autómata finito.
  - Objetivos: Representar lenguajes formales en su nivel léxico, utilizando los conocimientos mínimos de lingüística matemática y los paradigmas declarativos. Diseñar aceptores de lenguajes formales en su nivel léxico (AF) utilizando gramáticas regulares. Implementar los algoritmos propios de la teoría de autómatas, afianzando y consolidando los conocimientos previos adquiridos en el área de programación. Resolver problemas de aplicación de la informática teórica, en prácticas concretas en el laboratorio de la cátedra.
- UNIDAD 2: Lenguajes-Gramáticas Independientes del Contexto y Autómatas de Pila.
  - Fundamento: El conocimiento de los LIC, GIC y AP, permitirá comprender el nivel sintáctico de los lenguajes formales y en particular los de programación, posibilitando la solución de diversos problemas inherentes a la profesión, cuyo modelo matemático es el autómata de pila.
  - Objetivos: Representar lenguajes formales en su nivel sintáctico, comprendiendo el proceso de traducción de lenguajes de programación y en particular el proceso de compilación, utilizando herramientas de



metacompilación. Diseñar aceptores de lenguajes formales en su nivel sintáctico (AP) utilizando gramáticas independientes del contexto. Implementar los algoritmos propios de la teoría de autómatas, afianzando y consolidando los conocimientos previos adquiridos en el área de programación. Resolver problemas de aplicación de la informática teórica, en prácticas concretas en el laboratorio de la cátedra.

- UNIDAD 3: Lenguajes, Gramáticas y Modelos Generales.
  - Fundamento: La generalización de los modelos matemáticos que permiten representar los procesos computacionales, facilita la comprensión de los mismos y el análisis de su complejidad; en particular los procesos de traducción de lenguajes de programación.
  - Objetivos: Representar lenguajes formales en sus nivel semántico; utilizando los conocimientos de lógica de predicados y los paradigmas declarativos. Comprender la complejidad algorítmica mediante el análisis de los lenguajes recursivos, lenguajes recursivamente enumerables y las Máquinas de Turing. Comprender el proceso de traducción de lenguajes de programación y en particular el proceso de compilación utilizando un compilador desde la línea de comando del sistema operativo.

Contenido de Unidades Ver ( [Programa Analítico Detallado](#) )

- **Metodología de Enseñanza.**

Utilizamos estrategias centradas en el estudiante y mediadas por las TIC (Tecnologías de la Información y la Comunicación), que a la vez se apoyan en las teorías de cognición situada (constructivista socio-cultural, aprendizaje colaborativo y basado en la solución de problemas reales). Se trata de generar entornos de instrucción basados fundamentalmente en la actividad de los estudiantes, donde pueden concebir, desarrollar y poner en práctica diferentes proyectos que les permitan resolver problemas y les facilite al mismo tiempo, ciertos aprendizajes, ya que se va de lo concreto a lo abstracto. Dentro de las tecnologías educativas utilizadas, destacamos como innovación el uso de robótica para la solución de problemas de aplicación basados en los contenidos de la asignatura. En general la presencia de tecnologías en el aula de clase, busca proveer ambientes de aprendizaje interdisciplinarios donde los estudiantes adquieran habilidades para estructurar investigaciones y resolver problemas concretos, que les permita formarse como profesionales con capacidad para desarrollar nuevas destrezas, nuevos conceptos y dar respuesta eficiente a los entornos cambiantes del mundo actual. Se trata de aportar al desarrollo de la creatividad, el pensamiento crítico y el trabajo colaborativo de los estudiantes. Para la exposición de los temas, el profesor utiliza como apoyo material en formato tipo PPS y presentaciones dinámicas tipo PREZI, además se incentiva la participación de los alumnos, desarrolla ejemplos prácticos, orienta al alumno en la lectura del material bibliográfico y en las actividades de laboratorio y atiende consultas. El



estudiante cuenta con una guía de teoría y material multimedia que le permite ver los temas a desarrollar en forma previa a las clases, confeccionar fichas con síntesis y esquemas de los temas teóricos para utilizarlos en las clases prácticas. Tiene que resolver los ejercicios propuestos de los TP, realizar las actividades de laboratorio y del aula virtual. Luego debe profundizar dichos temas, mediante la lectura de textos y otros recursos que dispone en el aula virtual de la asignatura. El docente realiza un seguimiento de estas actividades mediante cuestionarios que el alumno debe contestar en el aula virtual. Resolución de problemas con la participación de los alumnos en clases prácticas. Realización de actividades en el laboratorio, implementación de algoritmos vistos en clases teóricas y prácticas. Se fomenta el trabajo en grupos y la realización de actividades colaborativas en el aula virtual. El alumno cuenta con una guía de trabajos prácticos, con ejercicios resueltos que se desarrollan en el aula, ejercicios propuestos y ejercicios adicionales. El auxiliar graduado explica ejercicios resueltos del trabajo práctico (TP), guía al alumno en la resolución de los ejercicios propuestos, orienta las actividades de laboratorio y atiende consultas. Confección de proyectos de articulación horizontal con los espacios curriculares Análisis de Sistemas, Sistemas Operativos y Paradigmas de Programación. Articulación vertical con las asignaturas previas y posteriores del área programación, mediante proyectos integradores que los estudiantes deben abordar en forma grupal.

El proceso instructivo se llevará a cabo a través de encuentros presenciales con distintas modalidades de clases, complementadas con actividades no-presenciales mediante el uso de un aula virtual. En todas las instancias presenciales se utilizará como apoyo las herramientas tecnológicas disponibles en las aulas y gabinetes de informática (proyector, computador portátil del docente, pizarra y marcador). En el caso de las actividades no-presenciales se utilizará el Campus Virtual de la UTN-FRT: <http://fvt.cvg.utn.edu.ar>

- ✓ CLASES TEORICAS: Estarán a cargo de los Profesores de la cátedra. En las mismas se desarrollarán los contenidos más relevantes del programa, con un enfoque sistémico, basándose en problemas de aplicación y con la mayor participación posible de los estudiantes; tratando siempre de estimular la reflexión, la crítica y la creatividad de los estudiantes.
- ✓ CLASES PRACTICAS: Estarán a cargo de los Auxiliares Docentes de la cátedra. En éstas se explicarán ejercicios tipo para cada temática abordada en las clases teórico-prácticas y en concordancia con los lineamientos dados por el profesor. Se aclararán las dudas respecto a los enunciados de los problemas propuestos y se orientará a los estudiantes en la resolución de los mismos, incentivando el trabajo colaborativo en el aula.
- ✓ CLASES DE AULA-TALLER: A cargo de los Profesores y auxiliares de la cátedra. En estas clases se trabajará con aplicaciones de los contenidos de la asignatura utilizando las herramientas de software adecuadas, con la



realización de actividades por parte de los estudiantes. Las mismas se desarrollarán en un Gabinete de Informática dotado de PCs, conexión a Internet y proyector.

- ✓ CLASES DE LABORATORIO: A cargo de los Profesores con el apoyo de los auxiliares graduados. El objeto de estas clases es el diseño de programas para implementar las soluciones de problemas de aplicación concretos. Se brindará a los estudiantes los elementos necesarios (bibliotecas de funciones, software de base: compiladores y metacompiladores, elementos de hardware: maquetas, plataformas de microcontroladores y componentes electrónicos) para que puedan llevar a cabo los programas solicitados. Las mismas se desarrollarán en un Gabinete de Informática dotado de PCs, conexión a Internet y proyector; más el Laboratorio de Robótica Educativa, donde contarán con los elementos necesarios para la prueba de las aplicaciones desarrolladas.
- ✓ CLASES DE CONSULTA: Cada Profesor y Auxiliar Docente de la cátedra, tendrá a cargo una clase de consulta semanal. El objetivo de las mismas es el de atender las dudas e inquietudes de los estudiantes en cuanto a los temas vistos en las clases teóricas o prácticas respectivamente.
- ✓ AULA VIRTUAL: Contendrá toda la información de la cátedra (estructura, programa, metodología, cronograma, etc.) y una serie de recursos que el estudiante podrá disponer como material de estudio, consulta y profundización de todos los temas del programa. Además tendrá una serie de actividades que podrá realizar para mejorar su rendimiento académico y que le facilitarán la comunicación tanto con los docentes como con sus propios pares. También dispondrán de actividades interactivas y colaborativas, que favorecerán el proceso de aprendizaje; dentro de las cuales se destacan: cuestionarios, glosarios, wikis, subidas de TP, Act. de Lab., foro de consulta, foro de socialización de estudiantes, foro de novedades, etc.

• **Metodología de Evaluación. (en concordancia con el Reglamento de Estudios Ord. 1549/2016)**

Para completar el cursado, el estudiante deberá cumplir los siguientes requisitos:

- 1) Asistencia mínima a clases de un 75%, salvo excepción prevista en punto 7.1.1.2. del Reglamento de Estudios.
- 2) Aprobación de 3 instancias de evaluación, una por cada unidad temática, o sus respectivas recuperaciones, con un mínimo de 6 puntos en una escala del 1 al 10, en cada una. Excepcionalmente la cátedra considerará la posibilidad de una



recuperación integral, para aquellos estudiantes que no aprueben la recuperación de cualquiera de las instancias de evaluación.

La calificación de cada instancia de evaluación o su recuperación seguirá el siguiente criterio: Sobre un total de contenidos del 100%, tendrá un 25% de Teoría, un 25% de Práctica, un 25% de Laboratorio, con la condición de obtener como mínimo en cada una de estas partes un 60% satisfactorio, el 25% restante es conceptual, para lo que se tendrá en cuenta la participación y desempeño del estudiante a lo largo del cursado.

- 3) Cumplimentar con las actividades de formación práctica, que consistirán en la defensa grupal de un proyecto de aplicación y una actividad de laboratorio, por cada unidad temática.

Aprobación de la Asignatura:

- 1) Aprobación Directa: se regirá por lo establecido en el punto 7.2.1 del Reglamento de Estudio. Además la cátedra establece las siguientes pautas:
  - a) Aprobar las 3 instancias de evaluación del cursado o sus recuperaciones, con un mínimo de 6 puntos cada una y un promedio mínimo de 8 puntos entre las tres.
  - b) Se permitirá recuperar cualquiera de las 3 instancias de evaluación para poder alcanzar el promedio de 8 o para mejorar el mismo. No incluye la posible recuperación integral en caso de ser otorgada.
  - c) Si en una recuperación la calificación es inferior a la obtenida en la instancia de evaluación correspondiente, se tomará la mayor.
- 2) Aprobación No Directa o con Examen Final: se regirá por lo establecido en los puntos 7.2.2. y 8.2.3 del Reglamento de Estudio. Además la cátedra establece las siguientes pautas:
  - a) El examen final es oral.
  - b) El estudiante puede comenzar desarrollando un tema a elección o defender un proyecto de aplicación previamente autorizado y supervisado por los docentes de la cátedra.
  - c) El tribunal puede solicitar el desarrollo de cualquier tema del programa analítico.
  - d) La calificación estará sujeta al criterio del tribunal examinador, el que podrá considerar el concepto que se tenga del estudiante durante el cursado para mejorar la misma.

- **Recursos didácticos a utilizar como apoyo a la enseñanza.**

Recursos físicos: pizarra, marcadores, borrador, computadores, proyector, impresora, acceso a Internet, placa controladora Arduino, maquetas de proyectos de aplicación. Recursos lógicos: software de presentaciones (OpenOffice, Prezi,



Adobe PDF) , compiladores de lenguaje de alto nivel (Dev-C++, NetBeans-Java), generadores de analizadores léxicos y sintácticos (Jflex y CUP), software educativo sobre teoría de autómatas (JFlap), lenguaje Arduino para programar placa controladora.

- **Articulación horizontal y vertical con otras materias.**

Articulación horizontal:

ASIGNATURA	TEMA	ACTIVIDAD
Análisis de Sistemas	UML: Diagrama de Estados.	Actividad de Laboratorio 1: Aplicación del modelo MEF en sistemas de control automático y robótica.
Sistemas Operativos	Expresiones Regulares en UNIX.	Proyecto de Aplicación para el TP1: Procesamiento de archivos de texto utilizando comandos del sistema operativo UNIX con expresiones regulares y el concepto de tuberías o pipes.
Paradigmas de Programación	Paradigma Declarativo: Funcional y Lógico.	Actividad de Laboratorio 1: Operaciones con lenguajes utilizando paradigma declarativo funcional con HASKELL.  Actividad de Laboratorio 3: Sistema Canónico de Donovan: implementación utilizando paradigma declarativo lógico con PROLOG.
Inglés I	Autómatas de Pila con Salida o Push-Down Transducers, cómo tópico en lecto-comprensión, redacción y expresión oral en inglés	Proyecto de Aplicación para el TP2: Interprete de expresiones algebraicas utilizando Transductores Push-Down.



Como experiencia piloto se llevará a cabo una articulación con Inglés I. La misma consistirá en la colaboración de la cátedra de Inglés I con los estudiantes, en la actividad descrita en el cuadro precedente. El estudiante debe desarrollar un programa como proyecto de aplicación del TP2, sobre la temática Transductores Push Down. Para lo cual deberá:

1. Consultar bibliografía en inglés, provista por nuestra cátedra, más otro material de Internet buscado por el estudiante y/o sugerido por la cátedra de Inglés I.
2. Redactar un informe del trabajo en inglés.
3. Realizar una presentación multimedia en inglés.
4. Exponer oralmente el proyecto en inglés.

El trabajo de los docentes de la asignatura Inglés I será la de preparar a los estudiantes para adquirir las cuatro habilidades, a saber, auditiva, oral, lectora y de escritura. Con estas herramientas podrán realizar el proyecto solicitado por la Cátedra de Sintaxis con la tutoría y apoyo de los docentes de Inglés I, así podrán realizar un informe acorde al nivel de inglés y una presentación en concordancia con la solicitada por los Docentes de Sintaxis.

Articulación vertical:

ASIGNATURA	TEMA	ACTIVIDAD
Matemática Discreta.	Teoría de Conjuntos, Combinatoria, Lógica de Predicados, Grafos y Árboles.	TP1: Operatoria con lenguajes formales y autómatas finitos; equivalencia y minimización de AF. TP2: Árbol de derivación y árbol total de un lenguaje formal. TP3: Sistema Canónico de Donovan.
Algoritmos y Estructuras de Datos.	Estructura de datos, paradigma imperativo procedural, lenguaje de programación procedural, estructuras de control y biblioteca de funciones para manejo de strings.	Proyecto de Aplicación para el TP1: Analizador Léxico de expresiones algebraicas programando los autómatas finitos.





Gestión de Datos.	Lenguaje de consulta estructurado (SQL)	Actividad de Laboratorio 2: Análisis Léxico y Sintáctico de un lenguaje de consulta estructurado para bases de datos relacionales simplificado (SQL): Implementación de Scanners y Parsers con las herramientas de metaprogramación (JFLEX-CUP).
-------------------	---	--

Con el objetivo de promover y fomentar las actividades de programación, la cátedra organizará eventos dentro del ámbito del Departamento de Sistemas, con características similares a los torneos de programación competitiva, en las cuales podrán participar estudiantes de todos los niveles de la carrera; así como jornadas de integración en programación orientadas al trabajo colaborativo con la participación de otras instituciones educativas.

- **Cronograma estimado de clases.**

Ver ( [Cronograma\\_de\\_Clases\\_y\\_Temas](#) )

- **Bibliografía.**

Se detalla la misma en Programa Analítico que se adjunta

- **Reuniones de cátedra programadas.**

Se llevarán a cabo al menos 7 reuniones de cátedra. Dos antes del inicio de clases, para planificación y coordinación de actividades. Una a mediados del primer cuatrimestre y otra a mediados del segundo cuatrimestre para seguimiento del desarrollo de la asignatura. Una al final del primer cuatrimestre y otra al final del segundo cuatrimestre para organizar actividades de evaluación. Una al final del ciclo lectivo para evaluar todo el proceso instruccional y cierre del mismo.



- **Seminarios de cátedra.**

Se llevarán a cabo dos seminarios internos de cátedra. Uno durante el primer cuatrimestre para abordar temáticas relacionadas con aplicaciones de las MEF en los sistemas de control automático y robótica. El otro durante el segundo cuatrimestre para tratar cuestiones concernientes a las herramientas de metaprogramación.

- **Cuadro de horas estimadas para intensidad en la formación práctica (de acuerdo a la Res.Min. 786/09)**

TEORICO-PRACTICAS Y AULA-TALLER	PROYECTOS Y DISEÑO (Actividades de Laboratorio)	RESOLUCIÓN DE PROBLEMAS DE INGENIERÍA (Proyectos de Aplicación para los Trabajos Prácticos)	TOTAL
96	16	16	128

**Programa Analítico Detallado****Contenido Analítico****Unidad 1: Introducción a la Teoría de Lenguajes Formales y Autómatas Finitos.**

Lingüística Matemática: Alfabetos, palabras y lenguajes. Operaciones con cadenas y con lenguajes. Niveles de un lenguaje. Gramáticas para estructuras de frases. Jerarquía de Chomsky. Lenguajes Regulares (LR) y Autómatas Finitos (AF): Gramáticas Regulares (GR), Expresiones Regulares (ER). Máquinas de Estados Finitos (MEF): Modelos de Mealy y de Moore. Autómata Finito Determinista (AFD). Equivalencia y minimización. Autómatas Finitos No Deterministas (AFND). Operaciones con AF. Conversión AFND/AFD. Obtención del AF a partir de una ER. Propiedades de los LR. Analizador Léxico (Scanner).

**Unidad 2: Lenguajes-Gramáticas Independientes del Contexto y Autómatas de Pila.**

Gramáticas Independientes del Contexto (GIC). Diagramas de Sintaxis y Formato BNF. Árboles de derivación. Ambigüedad. Árbol Total del Lenguaje. Simplificaciones de una GIC. Formas Normales. LIC y Autómatas de Pila (AP). Criterios de aceptación por estado final y por pila vacía. Diseño de un AP a partir de una GIC. Obtención de la GIC a partir de un AP. Propiedades de los LIC. Transductores Push Down. Analizador Sintáctico (Parser) Descendente (LL) y Ascendente (LR).

**Unidad 3: Lenguajes, Gramáticas y Modelos Generales.**

Gramáticas y Modelos: Los aceptores de lenguajes formales: Máquina de Turing (MT) y Autómata Linealmente Limitado (ALL). MT transformadora de secuencias, multicinta y multicelda. Lenguajes Recursivos y recursivamente enumerables. Resolubilidad y complejidad computacional. Formalismos para el análisis semántico: Sistemas Canónicos de Donovan y Esquemas de Traducción. Proceso de Compilación.

**BIBLIOGRAFÍA:**

Título:	INTRODUCCION A LA TEORIA DE AUTOMATAS LENGUAJES Y COMPUTACION ISBN 9788478290888		
Autor/es:	HOPCROFT JOHN E. , MOTWANI RAJEEV , ULLMAN JEFFREY D.		
Editorial:	PEARSON EDUCACION		
Edición:	2008	Ejemplares disponibles:	16

Título:	COMPILADORES: PRINCÍPIOS TÉCNICAS Y HERRAMIENTAS. ISBN 9789702611332		
Autor/es:	AHO ALFRED V., LAM MONICA S. , SETHI RAVI, ULLMAN JEFFREY D.		
Editorial:	PEARSON ADDISON-WESLEY		
Edición:	2008	Ejemplares disponibles:	8

Título:	COMPILADORES: TEORÍA E IMPLEMENTACIÓN ISBN 9786077854685		
Autor/es:	JACINTO RUIZ CATALAN		
Editorial:	ALFAOMEGA		
Edición:	2010	Ejemplares disponibles:	1

Título:	SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES (Guía de Teoría Digital)		
Autor/es:	BUABUD JORGE		
Editorial:	UTN-FRT		
Edición:	2009		

Sitio Web:	CAMPUS VIRTUAL DE LA UTN-FRT sobre plataforma MOODLE		
URL:	<a href="http://frt.cvg.utn.edu.ar">http://frt.cvg.utn.edu.ar</a>		
Curso:	Aula Virtual de Sintaxis y Semántica de los Lenguajes		

Título:	Teoría de Autómatas y Lenguajes Formales		
Autor/es:	JURADO MÁLAGA, ELENA.		
Editorial:	E-Book de Colección manuales UEX – 55 (E.E.E.S.)		
Edición:	2008		

Título:	Introduction to Computer Theory		
Autor/es:	COHEN, DANIEL I. A.		
Editorial:	E-Book de John Wiley & Son		
Edición:	1998		



**Cronograma de Clases y Temas**

<b><u>Área:</u></b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b><u>Asignatura:</u></b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
---------------------	--	---------------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
1	1	Nº 1	Teórico	Objetivos, contenidos, metodología, bibliografía, cronograma (Información disponible en Aula Virtual). Alfabetos, palabras y lenguajes. Niveles de un lenguaje: léxico, sintáctico, semántico y pragmático.
1	1	Nº 2	Práctico	TP1: Operaciones con cadenas y lenguajes.
2	1	Nº 3	Teórico	Actividad de Laboratorio 1: Operaciones con lenguajes utilizando paradigma declarativo-funcional con HASKELL.
2	1	Nº 4	Práctico	TP1: Especificación de formatos de datos con operaciones de lenguajes y validación de datos en la programación estructurada-modular.
3	1	Nº 5	Teórico	Gramáticas para estructuras de frases. Jerarquía de Chomsky. Ejemplos de Gramáticas para Lenguajes de Programación Hipotéticos. Herramienta de simulación JFLAP.
3	1	Nº 6	Práctico	TP1: Gramáticas.
4	1	Nº 7	Teórico	Gramáticas Regulares (GR): Formato estándar. Pasaje de GR Izq. a Der. GR con JFLAP.
4	1	Nº 8	Práctico	TP1: GR



## Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
5	1	Nº 9	Teórico	Expresiones Regulares (ER): Ecuaciones características y derivada de E.R. Propiedades de Lenguajes Regulares. ER con JFLAP.
5	1	Nº 10	Práctico	TP1: ER
6	1	Nº 11	Teórico	Máquinas de Estado Finito (MEF): Presentación del problema de aplicación en automatización. Estudio de las propuestas de solución del problema de aplicación en automatización. Formalización de la MEF: Modelos de Mealy y de Moore. MEF con JFLAP.
6	1	Nº 12	Práctico	TP1: Proyecto de Aplicación: Procesamiento de archivos de texto en UNIX con ER.
7	1	Nº 13	Teórico	Autómata Finito Determinista (AFD) AFD con JFLAP.
7	1	Nº 14	Práctico	TP1: AFD
8	1	Nº 15	Teórico	Equivalencia y minimización de AF. Simulación con JFLAP.
8	1	Nº 16	Práctico	TP1: Equivalencia y minimización de AF.



Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
9	1	Nº 17	Teórico	Actividad de Laboratorio 1: Aplicación del modelo MEF (Máquina de Estados Finitos) en sistemas de control automático y robótica. Plataforma Arduino.
9	1	Nº 18	Práctico	TP1: Proyecto de Aplicación: Analizador Léxico de expresiones algebraicas programando los autómatas finitos.
10	1	Nº 19	Teórico	Autómatas Finitos No Deterministas (AFND). Operaciones con AF AF- $\lambda$ con JFLAP.
10	1	Nº 20	Práctico	TP1: AF no deterministas.
11	1	Nº 21	Teórico	Obtención del AF- $\lambda$ a partir de una E.R. Propiedades de los LR. Simulación con JFLAP.
11	1	Nº 22	Práctico	TP1: ER a AF- $\lambda$
12	1	Nº 23	Teórico	Conversión AF- $\lambda$ a AFD. Simulación con JFLAP.
12	1	Nº 24	Práctico	TP1: AF- $\lambda$ a AFD.



Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
13	1	Nº 25	Teórico	Revisión Unidad 1.
13	1	Nº 26	Práctico	Revisión TP1.
14	1	Nº 27	Evaluativo	1era. Instancia de Evaluación: Parte Teórica.
14	1	Nº 28	Evaluativo	1era. Instancia de Evaluación: Parte Práctica.
14	1	Nº 29	Evaluativo	1era. Instancia de Evaluación: Parte Laboratorio.
				Receso Invernal
15	2	Nº 30	Teórico	Gramáticas Independientes del Contexto (GIC). Diagramas de Sintaxis y BNF. Árboles de derivación. Ambigüedad. Árbol Total del Lenguaje. GIC y Árboles con JFLAP.
15	2	Nº 31	Práctico	TP2: GIC, Diagramas de Sintaxis, BNF y Árboles.





Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
16	2	Nº 32	Teórico	Simplificaciones de una GIC: gramática bien formada. Formas Normales: formato normal de Chomsky (FNC). Simulación con JFLAP.
16	2	Nº 33	Práctico	TP2: Forma Normal de Chomsky.
17	2	Nº 34	Teórico	Simplificaciones de una GIC: eliminación recursividad por izquierda y factorización por izquierda. Formas Normales: formato normal de Greibach
17	2	Nº 35	Práctico	TP2: Formal Normal de Greibach.
18	2	Nº 36	Teórico	L.I.C. y Autómatas de Pila (AP) AP con JFLAP
18	2	Nº 37	Práctico	TP2: LIC a AP
19	2	Nº 38	Teórico	Actividad de Laboratorio 2: Análisis Léxico y Sintáctico de un lenguaje de consulta estructurado para bases de datos relacionales simplificado (SQL): Implementación de Scanners y Parsers con las herramientas de metaprogramación (JFLEX-CUP).
19	2	Nº 39	Práctico	TP2: Proyecto de Aplicación: Interprete de expresiones algebraicas utilizando Transductores Push Down.



## Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
20	2	Nº 40	Teórico	Obtención de la GIC a partir de un AP. Propiedades de los LIC. Simulación con JFLAP.
20	2	Nº 41	Práctico	TP2: GIC a AP
21	2	Nº 42	Teórico	Revisión Unidad 2
21	2	Nº 43	Práctico	Revisión TP2.
22	2	Nº 44	Evaluativo	2da. Instancia de Evaluación: Parte Teórica
22	2	Nº 45	Evaluativo	2da. Instancia de Evaluación: Parte Práctica
22	2	Nº 46	Evaluativo	2da. Instancia de Evaluación: Parte Laboratorio
23	3	Nº 47	Teórico	Gramáticas y Modelos: Los aceptores de lenguajes formales.



Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
23	3	Nº 48	Práctico	TP3: GI y GDC.
24	3	Nº 49	Teórico	Máquina de Turing (MT) y Autómata Linealmente Limitado (ALL). Simulación con JFLAP.
24	3	Nº 50	Práctico	TP3: Proyecto de Aplicación: Analizador de tipo de gramática y generación de una palabra.
25	3	Nº 51	Teórico	MT transformadora de secuencias. MT multicinta y multicelda. Lenguajes Recursivos y recursivamente enumerables. Conceptos de resolubilidad y complejidad computacional. Simulación con JFLAP.
25	3	Nº 52	Práctico	TP3: MT y ALL
26	3	Nº 53	Teórico	Actividad de Laboratorio 3: Formalismos para el análisis semántico: Sistemas Canónicos de Donovan (programación con paradigma declarativo-lógico en PROLOG) y Esquemas de Traducción (implementación con metacompilador CUP).
26	3	Nº 54	Práctico	TP3: MT transformadora de secuencias.
27	3	Nº 55	Teórico	Actividad de Laboratorio 3: Proceso de traducción de lenguaje de programación con compilador GCC desde consola de comandos.



Cronograma\_de\_Clases\_y\_Temas

<b>Área:</b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b>Asignatura:</b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
--------------	--	--------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
27	3	Nº 56	Práctico	TP3: Proyecto de Aplicación: Suma y multiplicación binaria con MT.
28	3	Nº 57	Teórico	Revisión Unidad 3.
28	3	Nº 58	Práctico	Revisión TP3.
29	3	Nº 59	Evaluativo	3era. Instancia de Evaluación: Parte Teórica
29	3	Nº 60	Evaluativo	3era. Instancia de Evaluación: Parte Práctica
29	3	Nº 61	Evaluativo	3era. Instancia de Evaluación: Parte Laboratorio
30	1	Nº 62	Evaluativo	Recuperación 1era. Instancia de Evaluación: Parte Teórica
30	1	Nº 63	Evaluativo	Recuperación 1era. Instancia de Evaluación: Parte Práctica



## Cronograma\_de\_Clases\_y\_Temas

<b><u>Área:</u></b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b><u>Asignatura:</u></b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
---------------------	--	---------------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
30	1	Nº 64	Evaluativo	Recuperación 1era. Instancia de Evaluación: Parte Laboratorio
				Vacaciones
31	2	Nº 65	Evaluativo	Recuperación 2da. Instancia de Evaluación: Parte Teórica
31	2	Nº 66	Evaluativo	Recuperación 2da. Instancia de Evaluación: Parte Práctica
31	2	Nº 67	Evaluativo	Recuperación 2da. Instancia de Evaluación: Parte Laboratorio
32	3	Nº 68	Evaluativo	Recuperación 3era. Instancia de Evaluación: Parte Teórica
32	3	Nº 69	Evaluativo	Recuperación 3era. Instancia de Evaluación: Parte Práctica
32	3	Nº 70	Evaluativo	Recuperación 3era. Instancia de Evaluación: Parte Laboratorio



Cronograma\_de\_Clases\_y\_Temas

<b><u>Área:</u></b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS.</b>	<b><u>Asignatura:</u></b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
---------------------	--	---------------------------	--

Nº de Semana Académica	Unidad Temática	Clase Nº	Carácter	Contenidos Temáticos
33	1,2,3	Nº 71	Evaluativo	Recuperación Integral: Parte Teórica
33	1,2,3	Nº 72	Evaluativo	Recuperación Integral: Parte Práctica
33	1,2,3	Nº 73	Evaluativo	Recuperación Integral: Parte Laboratorio
				Fin ciclo lectivo



**Fechas Estimativas de Parciales y Recuperatorios (En caso de corresponder será consensuado con Dpto. Ciencias Básicas)**

<b><u>Área:</u></b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS</b>	<b><u>Asignatura:</u></b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
---------------------	---	---------------------------	--

<b>Evaluación</b>	<b>Julio 2017</b>	<b>Octubre 2017</b>	<b>Noviembre 2017</b>	<b>Diciembre 2017</b>	<b>Febrero 2018</b>	<b>Marzo 2018</b>
<b>1º Instancia de Evaluación</b>	<b>3, 5 y 7</b>					
<b>Recuperación 1º Instancia de Evaluación</b>						
<b>2º Instancia de Evaluación</b>		<b>9, 11 y 13</b>				
<b>Recuperación 2º Instancia de Evaluación</b>					<b>19, 21, 23</b>	
<b>3º Instancia de Evaluación</b>			<b>27, 29</b>	<b>y 1</b>		
<b>Recuperación 3º Instancia de Evaluación</b>					<b>26, 28</b>	<b>y 2</b>
<b>Recuperación Integral</b>						<b>5, 7 y 9</b>



**Máquina Virtual** (En caso de corresponder) – Información para el Laboratorio, la misma debe ser lo más precisa posible

<b><u>Área:</u></b>	<b>PROGRAMACIÓN – TECNOLOGÍAS BÁSICAS</b>	<b><u>Asignatura:</u></b>	<b>SINTAXIS Y SEMÁNTICA DE LOS LENGUAJES</b>
---------------------	---	---------------------------	--

Software	Detalle	Observaciones
Sistema Operativo	Windows 7 o superior	
Buscador	Google Chrome, Mozilla Firefox o Internet Explorer.	
MS Office	Word, Excel, Power Point.	
Lenguajes de programación	Haskell, Prolog, C++, Java, Arduino.	
Herramientas de metaprogramación	JFLEX y CUP	
Software Simulador	JFLAP	
Compilador	GCC (GNU Compiler Collection)	
Herramienta de portabilidad	Cygwin (Comandos Unix para Windows)	